53-1003258-02
17 March 2015

Brocade NetIron Routing

Configuration Guide

Supporting Multi-Service IronWare R05.8.00a

BROCADE

© 2015, Brocade Communications Systems, Inc. All Rights Reserved.

ADX, Brocade, Brocade Assurance, the B-wing symbol, DCX, Fabric OS, HyperEdge, ICX, MLX, MyBrocade, OpenScript, The Effortless Network, VCS, VDX, Vplane, and Vyatta are registered trademarks, and Fabric Vision and vADX are trademarks of Brocade Communications Systems, Inc., in the United States and/or in other countries. Other brands, products, or service names mentioned may be trademarks of others.

Notice: This document is for informational purposes only and does not set forth any warranty, expressed or implied, concerning any equipment, equipment feature, or service offered or to be offered by Brocade. Brocade reserves the right to make changes to this document at any time, without notice, and assumes no responsibility for its use. This informational document describes features that may not be currently available. Contact a Brocade sales office for information on feature and product availability. Export of technical data contained in this document may require an export license from the United States government.

The authors and Brocade Communications Systems, Inc. assume no liability or responsibility to any person or entity with respect to the accuracy of this document or any loss, cost, liability, or damages arising from the information contained herein or the computer programs that accompany it.

The product described by this document may contain open source software covered by the GNU General Public License or other open source license agreements. To find out which open source software is included in Brocade products, view the licensing terms applicable to the open source software, and obtain a copy of the programming source code, please visit http://www.brocade.com/support/oscd.

Contents

Preface **18**
Document conventions 18
Text formatting conventions 18
Command syntax conventions 18
Notes, cautions, and warnings 19
Brocade resources 20
Contacting Brocade Technical Support 20
Document feedback 21

About This Document **22**
Audience 22
Supported hardware and software 22
Supported software 23
Notice to the reader 23
Related publications 24
How command information is presented in this guide 24

Configuring BGP4 (IPv4) **25**
BGP4 feature support 25
BGP4 overview 28
Relationship between the BGP4 route table and the IP route table 28
How BGP4 selects a path for a route (BGP best path selection algorithm) 29
BGP4 message types 31
Grouping of RIB-out peers 32
Implementation of BGP4 33
BGP4 restart 33
BGP4 Peer notification during a management module switchover 34
BGP4 neighbor local AS 35
Basic configuration and activation for BGP4 36
Disabling BGP4 37
BGP4 parameters 38
Parameter changes that take effect immediately 39
Parameter changes that take effect after resetting neighbor sessions 39
Parameter changes that take effect after disabling and re-enabling redistribution 40
Memory considerations 40
Basic configuration tasks required for BGP4 40
Enabling BGP4 on the device 40
Changing the device ID 40
Setting the local AS number 41
Adding a loopback interface 42
Adding BGP4 neighbors 43
Adding a BGP4 peer group 53
Optional BGP4 configuration tasks 55
Changing the Keep Alive Time and Hold Time 55
Changing the BGP4 next-hop update timer 56

Enabling fast external fallover....56
Changing the maximum number of paths for BGP4 Multipath load sharing....57
Customizing BGP4 Multipath load sharing....58
Specifying a list of networks to advertise....59
Changing the default local preference....60
Using the IP default route as a valid next-hop for a BGP4 route....60
Changing the default MED (Metric) used for route redistribution....61
Enabling next-hop recursion....61
Changing administrative distances....63
Requiring the first AS to be the neighbor AS....64
Disabling or re-enabling comparison of the AS-Path length....66
Enabling or disabling comparison of device IDs....66
Configuring the device to always compare Multi-Exit Discriminators....66
Treating missing MEDs as the worst MEDs....67
Configuring route reflection parameters....67
Configuring confederations....70
Aggregating routes advertised to BGP4 neighbors....73
Configuring BGP4 restart....74
Configuring BGP4 Restart for the global routing instance....74
Configuring BGP4 Restart for a VRF....74
Configuring timers for BGP4 Restart (optional)....74
BGP4 null0 routing....75
Configuring BGP4 null0 routing....76
Modifying redistribution parameters....78
Redistributing connected routes....79
Redistributing RIP routes....79
Redistributing OSPF external routes....80
Redistributing static routes....80
Redistributing IBGP routes....81
Filtering....81
AS-path filtering....81
BGP4 filtering communities....84
Defining and applying IP prefix lists....86
Defining neighbor distribute lists....87
Defining route maps....87
Using a table map to set the tag value....95
Configuring cooperative BGP4 route filtering....96
Four-byte Autonomous System Numbers (AS4)....98
Enabling AS4 numbers....99
BGP4 AS4 attribute errors....103
Error logs....103
Configuring route flap dampening....104
Globally configuring route flap dampening....105
Using a route map to configure route flap dampening for specific routes....106
Using a route map to configure route flap dampening for a specific neighbor....106
Removing route dampening from a route....107
Displaying and clearing route flap dampening statistics....107
Generating traps for BGP4....109
Configuring BGP4....109
Entering and exiting the address family configuration level....111
BGP route reflector....112
Configuring BGP route reflector....112
Specifying a maximum AS path length....115
Setting a global maximum AS path limit....115

Setting a maximum AS path limit for a peer group or neighbor..... 116
BGP4 max-as error messages.....116
Originating the default route.....117
Changing the default metric used for route cost..... 117
Configuring a static BGP4 network 117
Setting an administrative distance for a static BGP4 network.....118
Limiting advertisement of a static BGP4 network to selected neighbors.....118
Route-map continue clauses for BGP4 routes.....119
Specifying route-map continuation clauses.....119
Dynamic route filter update..... 120
Generalized TTL Security Mechanism support.....123
show metro mp-vlp-queue.....125
clear metro mp-vlp-queue..... 127
Displaying BGP4 information..... 127
Displaying summary BGP4 information..... 127
Displaying the active BGP4 configuration.....130
Displaying summary neighbor information.....130
Displaying BGP4 neighbor information..... 132
Displaying peer group information..... 142
Displaying summary route information.....142
Displaying the BGP4 route table.....143
Displaying BGP4 route-attribute entries.....150
Displaying the routes BGP4 has placed in the IP route table..... 151
Displaying route flap dampening statistics.....151
Displaying the active route map configuration..... 152
Displaying BGP4 graceful restart neighbor information..... 153
Displaying AS4 details..... 153
Displaying route-map continue clauses..... 162
Updating route information and resetting a neighbor session.....164
Using soft reconfiguration..... 164
Dynamically requesting a route refresh from a BGP4 neighbor.....166
Closing or resetting a neighbor session.....169
Clearing and resetting BGP4 routes in the IP route table..... 169
Clearing traffic counters..... 169
Clearing diagnostic buffers.....170

OSPFv2..... 171
OSPFv2 (NI) PROPOSED..... 171
OSPF overview..... 172
OSPF point-to-point links..... 174
Designated routers in multi-access networks.....175
Designated router election in multi-access networks..... 175
OSPF RFC 1583 and 2328 compliance.....177
Reduction of equivalent AS external LSAs..... 177
Algorithm for AS external LSA reduction.....178
Support for OSPF RFC 2328 Appendix E.....179
OSPF graceful restart..... 180
OSPF stub router advertisement.....181
OSPF Shortest Path First throttling.....181
IETF RFC and internet draft support.....182
Dynamic OSPF activation and configuration.....182
OSPF VRF-Lite for customer edge routers.....182
Configuring OSPF.....183
Configuration rules.....183
OSPF parameters..... 183
Enable OSPF on the device.....184

Assign OSPF areas....185
Assign a totally stubby area....186
Assigning an area range (optional)....189
Assigning an area cost (optional parameter)....190
Assigning interfaces to an area....191
Setting all OSPFv2 interfaces to the passive state....191
Modify interface defaults....192
Change the timer for OSPF authentication changes....194
Block flooding of outbound LSAs on specific OSPF interfaces....195
Assign virtual links....196
Modify virtual link parameters....197
Changing the reference bandwidth for the cost on OSPF interfaces....199
Define redistribution filters....200
Modify default metric for redistribution....202
Enable route redistribution....202
Disable or re-enable load sharing....204
Configure external route summarization....205
Configure default route origination....206
Supported match and set conditions....207
OSPF non-stop routing....208
Synchronization of critical OSPF elements....208
Link state database synchronization....208
Neighbor router synchronization....209
Interface synchronization....209
BFD with OSPF NSR....210
Standby module operations....210
Neighbor database....210
LSA database....210
Enabling and disabling NSR....211
Limitations of NSR....211
Adding additional parameters....211
Disabling configuration....212
OSPF distribute list....213
Configuring an OSPF distribution list using ACLs....213
Configuring an OSPF distribution list using route maps....214
Modify SPF timers....215
Modify redistribution metric type....216
Modify administrative distance....216
Configure OSPF group Link State Advertisement (LSA) pacing....217
OSPF ABR type 3 LSA filtering....218
Displaying the configured OSPF area prefix list....220
Modify OSPF traps generated....221
Modify OSPF standard compliance setting....222
Modify exit overflow interval....222
Specify types of OSPF Syslog messages to log....222
Configuring an OSPF network type....223
Configuring OSPF Graceful Restart....224
Configuring OSPF router advertisement....226
Configuring OSPF shortest path first throttling....227
Displaying OSPF information....228
Displaying general OSPF configuration information....229
Displaying CPU utilization and other OSPF tasks....231
Displaying OSPF area information....232
Displaying OSPF neighbor information....233
Displaying OSPF interface information....235
Displaying OSPF interface brief information....237
Displaying OSPF route information....238

Displaying OSPF database information....240
Displaying OSPF external link state information....242
Displaying OSPF database-summary information....243
Displaying OSPF database link state information....244
Displaying OSPF ABR and ASBR information....246
Displaying OSPF trap status....247
Viewing Configured OSPF point-to-point links....247
Displaying OSPF virtual neighbor and link information....249
Clearing OSPF neighbors....251
Displaying OSPF Graceful Restart information....251
Displaying OSPF Router Advertisement information....252
Displaying the OSPF area translator status information....252
Clearing OSPF information....253
Clearing OSPF neighbors....253
Disabling and re-enabling the OSPF process....254
Clearing OSPF routes....254

IS-IS (IPv4)....255
Relationship to IP route table....258
Intermediate systems and end systems....258
Domain and areas....259
Level-1 routing and Level-2 routing....259
Neighbors and adjacencies....260
Designated IS....260
Three-way handshake for point-to-point adjacencies....261
IS-IS CLI levels....262
Global configuration level....262
Address family configuration level....262
Interface level....263
Enabling IS-IS globally....263
Globally configuring IS-IS on a device....264
Setting the overload bit....264
Configuring authentication....265
Changing the IS-IS level globally....269
Disabling or re-enabling display of hostname....269
Changing the Sequence Numbers PDU interval....270
Changing the maximum LSP lifetime....270
Changing the LSP refresh interval....271
Changing the LSP generation interval....271
Changing the LSP interval and retransmit interval....271
Changing the SPF timer....271
Configuring the IS-IS PSPF exponential back-off feature....272
Configuring the IS-IS flooding mechanism....272
Globally disabling or re-enabling hello padding....273
Logging adjacency changes....273
Logging invalid LSP packets received....274
Disabling partial SPF optimizations....274
Disabling incremental SPF optimizations....274
IS-IS incremental shortcut LSP SPF optimization....275
Configuring IPv4 address family route parameters....275
Changing the metric style....276
Changing the maximum number of load sharing paths....276
Enabling advertisement of a default route....276
Changing the administrative distance for IPv4 IS-IS....278
Configuring summary addresses....278
Redistributing routes into IPv4 IS-IS....279
Changing the default redistribution metric....279

Globally change the default redistribution metric 280
Configuration steps..... 280
ISIS Show command..... 281
Redistributing static IPv4 routes into IPv4 IS-IS..... 281
Redistributing directly connected routes into IPv4 IS-IS..... 281
Redistributing RIP routes into IPv4 IS-IS..... 282
Redistributing OSPF routes into IPv4 IS-IS..... 282
Redistributing BGP4+ routes into IPv4 IS-IS..... 282
Redistributing IPv4 IS-IS routes within IPv4 IS-IS..... 282
Configuring IS-IS point-to-point over Ethernet..... 283
Brocade IS-IS Router A configuration..... 284
Brocade IS-IS Router B configuration..... 284
Displaying IS-IS point-to-point configuration..... 284
Configuring IS-IS over a GRE IP tunnel 284
Configuration considerations..... 285
Configuring IS-IS over a GRE IP tunnel..... 285
Displaying IS-IS over GRE IP tunnel..... 287
IS-IS Non-Stop Routing..... 287
Overview..... 288
Limitations..... 288
Enabling and disabling IS-IS NSR..... 288
Displaying the IS-IS NSR status..... 289
Configuring ISIS properties on an interface..... 294
Disabling and enabling IS-IS on an interface..... 294
Disabling or re-enabling formation of adjacencies..... 294
Setting the priority for designated IS election..... 295
Limiting access to adjacencies with a neighbor..... 295
Changing the IS-IS level on an interface..... 295
Disabling and enabling hello padding on an interface..... 296
Changing the hello interval..... 296
Changing the hello multiplier..... 296
DIS hello interval..... 297
Changing the metric added to advertised routes..... 297
Displaying IPv4 IS-IS information..... 298
Displaying ISIS general information..... 298
Displaying the IS-IS configuration in the running-config..... 302
Displaying the name mappings..... 303
Displaying neighbor information..... 303
Displaying IS-IS Syslog messages..... 306
Displaying interface information..... 306
Displaying route information..... 311
Displaying LSP database entries..... 312
Displaying traffic statistics..... 316
Displaying error statistics..... 316
Displaying the IS-IS SPF Log..... 318
Clearing the IS-IS SPF Log..... 321
Triggering the router to run SPF..... 322
Clearing IS-IS information..... 322
Clearing a specified LSP from IS-IS database..... 323

RIP..... 324
RIP feature support..... 324
RIP Overview..... 324
RIP parameters and defaults..... 325
RIP global parameters..... 325
RIP interface parameters..... 326
Configuring RIP parameters..... 327

Enabling RIP 327
Configuring metric parameters 327
Changing the administrative distance 328
Configuring redistribution 328
Configuring route learning and advertising parameters 330
Changing the route loop prevention method 331
Suppressing RIP route advertisement on a VRRP or VRRPE backup interface 331
Configuring RIP route filters using prefix-lists and route maps 332
Setting RIP timers 333
Displaying RIP Information 334
Displaying CPU utilization statistics 336

Policy-Based Routing (IPv4) 337
Configuration considerations 338
Configuring a PBR policy 339
Configure the route map 339
Enabling PBR 343
Configuration examples 344
Basic example 344
Setting the next hop 345
Setting the next hop to a GRE tunnel 346
Setting the output interface to the null interface 346
Selectively applying normal routing to packets 347
Applying IPv6 PBR next hop VLAN flooding 347
LAG formation 348
Policy based routing with the preserve VLAN option 348
Configuring a physical interface to accept all VLAN packets for PBR 348
Configuration considerations 348
Configuring policy based routing with the preserve VLAN option 348
Configuration examples 349
Preserve VLAN IDs and forwarding to single destination 349
Preserve VLAN IDs and replicate to multiple ports within a VLAN 350
Specify an interface as a PBR next-hop 350
Show telemetry command 350
Policy-based routing support for preserve VLAN 351
Configuration considerations 352

Multi-VRF 353
Overview of Multi-VRF 353
Benefits and applications of Multi-VRF 356
Summary 358
Configuring Multi-VRF 358
Configuration 1 359
Configuration 2 363

Inter-VRF Routing 366
Features and benefits 367
Configuration considerations 369
Tie breaker rules 369
Maximum route limitations 369
Configuring Inter-VRF routing 370
Show commands 371
Clearing IP routes 375

Configuring the number of VRFs for IPv4 and IPv6 377
Modified CLI commands 377

Management VRF 379
Management VRF overview 379
Source interface and management VRF compatibility 380
Supported management applications 380
Configuring a global management VRF 383
Configuration notes 384
Displaying the management VRF information 384

Basic IPv6 Connectivity 388
Enabling IPv6 routing 390
Configuring IPv6 on each interface 391
Configuring a global or unique local IPv6 unicast address 391
Configuring a link-local IPv6 address 392
Configuring IPv6 anycast addresses 393
Configuring IPv6 127 bit mask address 394
Benefits of using 127 bit mask: 394
Configuring the management port for an IPv6 automatic address configuration 394
IPv6 host support 395
IPv6 host supported features 395
Restricting SNMP access to an IPv6 node 395
Specifying an IPv6 SNMP trap receiver 396
Restricting Telnet access by specifying an IPv6 ACL 396
Restricting SSH access by specifying an IPv6 ACL 396
Restricting Web management access by specifying an IPv6 ACL. 397
Restricting SNMP access by specifying an IPv6 ACL 397
Restricting Web management access to your device to a specific IPv6 host 398
Specifying an IPv6 Syslog server 398
Viewing IPv6 SNMP server addresses 398
Disabling router advertisement and solicitation messages 399
IPv6 Non stop routing and graceful restart 399
Limitations 399
Supported protocols 400
Restart global timers 400
Configuring NSR and graceful restart on OSPFv3 401
Configuring Non Stop Routing on IS-IS 404
Configuring BGP graceful restart 404
IPv6 Hitless OS upgrade 406
Configuring IPv4 and IPv6 protocol stacks 407
Configuring IPv6 Domain Name Server (DNS) resolver 408
Defining a DNS entry 408
IPv6 Non-Stop Routing support 409
Limitations 409
Configuring IPv6 NSR support 409
ECMP load sharing for IPv6 410
Disabling or re-enabling ECMP load sharing for IPv6 410
Changing the maximum number of load sharing paths for IPv6 410
DHCP relay agent for IPv6 411
DHCPv6 Relay Agent Prefix Delegation Notification 411
Configuring DHCP for IPv6 relay agent 413
Configuring DHCPv6 Relay Agent Prefix Delegation Notification 413

Displaying the DHCPv6 Relay Agent Prefix Delegation Notification information....415
Enabling support for network-based ECMP load sharing for IPv6....418
Displaying ECMP load-sharing information for IPv6....419
Configuring IPv6 ICMP....419
Configuring ICMP rate limiting....420
Enabling ICMP redirect messages....420
Disabling or re-enabling ICMP redirect messages....421
Disabling ICMP error messages for source-routed IPv6 packets....421
Enabling ICMP error messages for an unreachable address....421
Enabling ICMP messages for an unreachable route....422
Enabling ICMP error messages for IPv6 packets with hop-limit 0....422
Enabling ICMP error messages for multicast Too Big packets....422
Enabling ICMP error messages for CES/CER devices....423
Configuring IPv6 neighbor discovery....423
Neighbor solicitation and advertisement messages....424
Router advertisement and solicitation messages....424
Neighbor redirect messages....425
Setting neighbor solicitation parameters for duplicate address detection....425
Setting IPv6 router advertisement parameters....426
Controlling prefixes advertised in IPv6 router advertisement messages....426
Configuring the Domain Name of DNS suffix....427
Configuring the recursive DNS server addresses and lifetime multiplier....428
Setting flags in IPv6 router advertisement messages....428
Configuring reachable time for remote IPv6 nodes....429
IPv6 source routing security enhancements....430
Complete filtering of IPv6 source-routed packets....430
Selective filtering of IPv6 source-routed packets using ACLs....431
Complete and selective filtering combination and order of application....432
Configuration examples for complete and selective filtering of source routed packets....432
Changing the IPv6 MTU....434
How to determine the actual IPv6 MTU value....435
Configuring static neighbor entries....435
Limiting the number of hops an IPv6 packet can traverse....436
QoS for IPv6 traffic....436
Configuring the rate limit for IPv6 subnet traffic....437
Clearing global IPv6 information....437
Clearing the IPv6 cache....437
Clearing IPv6 neighbor information....438
Clearing IPv6 routes from the IPv6 route table....438
Clearing IPv6 traffic statistics....438
Clearing statistics for IPv6 subnet rate limiting....439
Displaying global IPv6 information....439
Displaying IPv6 cache information....439
Displaying IPv6 interface information....440
Displaying interface counters for all ports....442
Displaying IPv6 neighbor information....443
Displaying the IPv6 route table....445
Displaying local IPv6 devices....450
Displaying IPv6 TCP information....451
Displaying IPv6 traffic statistics....454
Displaying statistics for IPv6 subnet rate limiting....458
Displaying IPv6 information for Router Advertisement Options....459

Displaying IPv6 interface information for Router Advertisement Options....460
Commands....461
ipv6 nd ra-dns-server....461
ipv6 nd ra-domain-name....462

IPv6 Addressing....463
IPv6 addressing overview....463
IPv6 address types....464
IPv6 stateless auto-configuration....465

IPv6 Prefix List....467
Configuring an IPv6 prefix list....467
Displaying prefix list information....468
IPv6 ND Prefix Suppress....469
Configuring IPv6 Prefix Suppress....469
IPv6 ND Router Advertisement Control....470

Managing a Device Over IPv6....472
Using the IPv6 copy command....473
Copying a file to an IPv6 TFTP server....473
Copying a file from an IPv6 TFTP server....474
Using the IPv6 ncopy command....475
Copying a primary or secondary boot image from flash memory to an IPv6 TFTP server....475
Copying the running or startup configuration to an IPv6 TFTP server....476
Uploading files from an IPv6 TFTP server....476
Using the IPv6 ping command....477
Using the IPv6 traceroute command....478
Using Telnet....478
Using the IPv6 Telnet command....479
Establishing a Telnet session from an IPv6 host....479
Using Secure Shell....480

Static IPv6 Routes....481
Configuring a static IPv6 route....481
Configuring a IPv6 static multicast route....483
BFD for Static Routes....484
Configuration considerations....484
Configuring BFD for static routes....485
Show commands....486

Configuring BGP4+....488
Supported BGP4+ features....488
BGP4+ overview....489
Address family configuration level....489
Configuring BGP4+....490
Enabling BGP4+....490
Configuring BGP4+ neighbors using global or unique link local IPv6 addresses....491
Adding BGP4+ neighbors using link-local addresses....492

Configuring a BGP4+ peer group....493
Advertising the default BGP4+ route....495
Importing routes into BGP4+ 496
Redistributing prefixes into BGP4+....496
Aggregating routes advertised to BGP4 neighbors....497
Using route maps....497
Enabling next-hop recursion....498
Clearing BGP4+ information....500
Removing route flap dampening....500
Clearing route flap dampening statistics....500
Clearing BGP4+ local route information....501
Clearing BGP4+ neighbor information....501
Clearing and resetting BGP4+ routes in the IPv6 route table....504
Clearing traffic counters for all BGP4+ neighbors....504
Displaying BGP4+ information....504
Displaying the BGP4+ route table....505
Displaying BGP4+ route information....510
Displaying BGP4+ route-attribute entries....512
Displaying the BGP4+ running configuration....514
Displaying dampened BGP4+ paths....514
Displaying filtered-out BGP4+ routes....515
Displaying route flap dampening statistics....519
Displaying BGP4+ neighbor information....520
Displaying BGP4+ peer group configuration information....543
Displaying BGP4+ summary....544
Configuring BGP4+ graceful restart....546
Configuring BGP4+ graceful restart for the global routing instance..546
Configuring timers for BGP4+ graceful restart (optional)....546
Displaying BGP4+ graceful restart neighbor information....547

OSPFv3....548
OSPFv3 feature support....548
OSPFv3 overview....549
Link-state advertisement types for OSPFv3....550
Configuring OSPFv3....550
Enabling OSPFv3....550
Assigning OSPFv3 areas....552
Assigning an area cost for OSPFv3 (optional parameter)....555
Specifying a network type....557
Configuring virtual links....557
Changing the reference bandwidth for the cost on OSPFv3 interfaces....559
Redistributing routes into OSPFv3....561
Filtering OSPFv3 routes....564
Configuring default route origination....566
Modifying Shortest Path First timers....567
Modifying administrative distance....568
Configuring the OSPFv3 LSA pacing interval....569
Modifying exit overflow interval....569
Modifying external link state database limit....570
Setting all OSPFv3 interfaces to the passive state....570
Modifying OSPFv3 interface defaults....570
Disabling or re-enabling event logging....571
IPsec for OSPFv3....571
Configuring IPsec for OSPFv3....572
Configuring OSPFv3 Graceful Restart Helper mode....578
Configuring OSPFv3 Non-stop routing (NSR)....579

Configuring OSPFv3 max-metric router LSA..... 579
Displaying OSPFv3 information..... 581
General OSPFv3 configuration information..... 582
Displaying OSPFv3 area information..... 582
Displaying OSPFv3 database information..... 583
Displaying IPv6 interface information..... 589
Displaying IPv6 OSPFv3 interface information..... 589
Displaying OSPFv3 memory usage..... 593
Displaying OSPFv3 neighbor information..... 594
Displaying routes redistributed into OSPFv3..... 597
Displaying OSPFv3 route information..... 598
Displaying OSPFv3 SPF information..... 600
Displaying OSPFv3 GR Helper mode information 602
Displaying OSPFv3 NSR information..... 602
Displaying OSPFv3 max-metric router LSA information..... 603
Displaying IPv6 OSPF virtual link information..... 603
Displaying OSPFv3 virtual neighbor information..... 604
IPsec examples..... 605
OSPFv3 clear commands 612
Clearing all OSPFv3 data..... 612
Clearing all OSPFv3 packet counters..... 612
Scheduling Shortest Path First (SPF) calculation..... 613
Clearing all redistributed routes from OSPFv3..... 613
Clearing OSPFv3 neighbors..... 613

IS-IS (IPv6)..... 615
IPv6 IS-IS single-topology mode..... 616
IS-IS CLI levels..... 617
Global configuration level..... 618
Address family configuration level..... 618
Interface level 619
Configuring IPv6 IS-IS..... 619
Enabling IPv6 IS-IS globally..... 619
Enabling IS-IS and assigning an IPv6 address to an interface 620
Configuring IPv6 IS-IS single topology..... 621
Globally configuring IS-IS on a device..... 621
Configuring IPv6 specific address family route parameters..... 621
Changing the maximum number of load sharing paths..... 622
Enabling advertisement of a default route..... 622
Changing the administrative distance for IPv6 IS-IS..... 623
Configuring summary prefixes..... 624
Redistributing routes into IPv6 IS-IS..... 624
Changing the default redistribution metric..... 625
Globally change the default redistribution metric..... 625
Configuration steps..... 625
ISIS Show command..... 626
Redistributing static IPv6 routes into IPv6 IS-IS..... 626
Redistributing directly connected routes into IPv6 IS-IS..... 627
Redistributing RIPng routes into IPv6 IS-IS..... 627
Redistributing OSPF version 3 routes into IPv6 IS-IS..... 627
Redistributing BGP4+ routes into IPv6 IS-IS..... 628
Redistributing IPv6 IS-IS routes within IPv6 IS-IS..... 628
Disabling and re-enabling IPv6 protocol-support consistency checks..... 629
Configuring IS-IS properties on an interface..... 629
Changing the metric added to advertised routes..... 629
IPv6 IS-IS Non-Stop Routing..... 630

Overview....630
Configuring IS-IS NSR....631
Displaying IPv6 IS-IS information....631
Displaying IPv6 IS-IS information....631
Displaying the IPv6 IS-IS configuration in the running configuration....633
Displaying IPv6 IS-IS error statistics....633
Displaying LSP database entries....634
Displaying the system ID to name mappings....640
Displaying IPv6 IS-IS interface information....640
Displaying IPv6 IS-IS memory usage....644
Displaying IPv6 IS-IS neighbor information....644
Displaying IPv6 IS-IS redistribution information....646
Displaying the IPv6 IS-IS route information....647
Displaying IPv6 IS-IS traffic statistics....648
IPv6 IS-IS Multi-Topology....649
Configuration considerations for IPv6 IS-IS MT....649
Migrating to IPv6 IS-IS MT....650
Maintaining MT adjacencies....650
New TLV attributes....650
Enabling IPv6 IS-IS MT....650
Configuring the IS-IS IPv6 PSPF exponential back-off feature....651
Changing the SPF timer....652
Changing the metric added value....652
Configuration example to deploy IPv6 IS-IS MT....652
default-link-metric....655
reverse-metric....657
isis reverse-metric....660

RIPng....663
RIPng feature support....663
RIPng Overview....663
Configuring RIPng....664
Enabling RIPng....664
Configuring RIPng timers....665
Configuring route learning and advertising parameters....666
Redistributing routes into RIPng....667
Controlling distribution of routes through RIPng....668
Configuring poison reverse parameters....668
Clearing RIPng routes from IPv6 route table....669
Clearing RIPng for a VRF instance....669
Displaying RIPng information....669
Displaying RIPng configuration....669
Displaying RIPng routing table....671

Policy-Based Routing (IPv6)....673
Configuration considerations....674
Considerations specific to Brocade NetIron CES Series and Brocade NetIron CER Series....675
Configuring an IPv6 PBR policy....675
Configuring the route map....675
Enabling IPv6 PBR....679
LAG formation....679
Configuration examples....679
Basic example....679
Combined example....679

Selectively applying normal routing to packets..... 680
Displaying IPv6 PBR information..... 680
Displaying IPv6 accounting information..... 680
Displaying IPv6 PBR route map information..... 681
Displaying IPv6 ACL and route map informationon the Brocade NetIron CES Series and Brocade NetIron CER Series..... 681
Policy based routing with the preserve VLAN option..... 682
Configuring a physical interface to accept all VLAN packets for PBR..... 683
Configuration considerations.....683
Configuring policy based routing with the preserve VLAN option... 683
Configuration examples..... 683
Preserve VLAN IDs and forwarding to specific egress port 684
Preserve VLAN IDs and forwarding to multiple ports within a VLAN..... 684
Applying IPv6 PBR next hop VLAN flooding.....684
Policy-based routing support for preserve VLAN.....685
Configuration considerations.....685

VRRP and VRRP-E..... 686
Overview of VRRP..... 688
Standard VRRP 688
Enhancements to VRRP..... 690
Overview of VRRP-E.....694
ARP behavior with VRRP-E.....695
Comparison of VRRP and VRRP-E..... 695
VRRP and VRRP-E parameters..... 696
Configuring parameters specific to VRRP.....700
Configuring the VRRP version..... 700
Configuring the Owner for IPv4.....701
Configuring the Owner for IPv6.....701
Configuring a Backup for IPv4..... 702
Configuring a Backup for IPv6..... 702
Configuration rules and feature limitations for VRRP..... 703
Configuring parameters specific to VRRP-E 703
Configuring IPv4 VRRP-E.....703
Configuring IPv6 VRRP-E.....704
Configuration rules and feature limitations for VRRP-E.....704
Configuring additional VRRP and VRRP-E parameters..... 705
Authentication type.....705
Suppressing RIP advertisements on backup routers for the backup up interface..... 707
Hello interval..... 707
Dead interval.....708
Backup hello message state and interval..... 708
Track port.....709
Track priority..... 709
Backup preempt.....709
Master router abdication and reinstatement..... 710
VRRP-extended slow start.....711
VRRP-extended scale timer.....711
Enable and Disable password display..... 712
Multiple virtual IP address support for VRRP-E.....712
Configuring multiple virtual IP addresses for VRRP-E.....713
Displaying multiple virtual IP addresses for VRRP-E information...714
VRRP-E scaling using logical groups..... 715
Configuring VRRP-E scaling.....716

Displaying VRRP-E scaling information....717
Alternate VRRPv2 checksum for VRRPv3 IPv4 sessions....718
Enabling the v2 checksum computation method in a VRRPv3 IPv4 session....718
Displaying alternate v2 checksum settings....719
Displaying VRRP and VRRP-E information for IPv4....719
Displaying summary information....720
Displaying detailed information....721
Displaying statistics....725
Displaying VRRP and VRRP-E information for IPv6....725
Displaying summary information....726
Displaying detailed information....726
Displaying statistics....727
Displaying configuration information for VRRP and VRRP-E....727
Clearing VRRP or VRRP-E statistics....728
Configuration examples....728
VRRP example for IPv4....728
VRRP example for IPv6....729
VRRP-E example for IPv4....731
VRRP-E example for IPv6....732
VRRP-E Extension for Server Virtualization....734
VRRP-E Extension for server virtualization configuration example....735
Packets from the local subnet of the virtual IP address....735
IPv4 VRF support....735
Configuration considerations....735

GPRS Tunneling Protocol....737
GPRS Tunneling Protocol Overview....737
GPRS Tunneling Protocol Filtering and Load-balancing....737
About GTP load-balance configuration....737
Enable masking of the TEID (Tunnel endpoint identifier) field for GTP packets....739
MPLS unknown label handling....739
Enabling internal loopback....740
GTP Profile configuration commands....740

Enabling and Disabling MAC movement syslog on MLX device....745

Routing commands....747
default-link-metric....748
ipv6 nd ra-dns-server....749
ipv6 nd ra-domain-name....750
isis reverse-metric....752
mac-move-det-syslog....755
neighbor next-hop-self (BGP)....757
reverse-metric....758
show metro mp-vlp-queue....761
system-max ip-vrf-route....763
system-max ipv6-vrf-route....764

Index....765

Preface

- Document conventions....18
- Brocade resources....20
- Contacting Brocade Technical Support....20
- Document feedback....21

Document conventions

The document conventions describe text formatting conventions, command syntax conventions, and important notice formats used in Brocade technical documentation.

Text formatting conventions

Text formatting conventions such as boldface, italic, or Courier font may be used in the flow of the text to highlight specific words or phrases.

Format	Description
bold text	Identifies command names
	Identifies keywords and operands
	Identifies the names of user-manipulated GUI elements
	Identifies text to enter at the GUI
italic text	Identifies emphasis
	Identifies variables and modifiers
	Identifies paths and Internet addresses
	Identifies document titles
`Courier font`	Identifies CLI output
	Identifies command syntax examples

Command syntax conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.

Convention	Description
value	In Fibre Channel products, a fixed value provided as input to a command option is printed in plain text, for example, **--show** WWN.
[ ]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{ **x** | **y** | **z** }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options. In Fibre Channel products, square brackets may be used instead for this purpose.
x | **y**	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a “soft” line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Notes, cautions, and warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

NOTE
A Note provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

ATTENTION
An Attention statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

CAUTION

A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

DANGER

A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Brocade resources

Visit the Brocade website to locate related documentation for your product and additional Brocade resources.

You can download additional publications supporting your product at www.brocade.com. Select the Brocade Products tab to locate your product, then click the Brocade product name or image to open the individual product page. The user manuals are available in the resources module at the bottom of the page under the Documentation category.

To get up-to-the-minute information on Brocade products and resources, go to MyBrocade. You can register at no cost to obtain a user ID and password.

Release notes are available on MyBrocade under Product Downloads.

White papers, online demonstrations, and data sheets are available through the Brocade website.

Contacting Brocade Technical Support

As a Brocade customer, you can contact Brocade Technical Support 24x7 online, by telephone, or by e-mail. Brocade OEM customers contact their OEM/Solutions provider.

Brocade customers

For product support information and the latest information on contacting the Technical Assistance Center, go to http://www.brocade.com/services-support/index.html.

If you have purchased Brocade product support directly from Brocade, use one of the following methods to contact the Brocade Technical Assistance Center 24x7.

Online	Telephone	E-mail
Preferred method of contact for non-urgent issues: • My Cases through MyBrocade • Software downloads and licensing tools • Knowledge Base	Required for Sev 1-Critical and Sev 2-High issues: • Continental US: 1-800-752-8061 • Europe, Middle East, Africa, and Asia Pacific: +800-AT FIBREE (+800 28 34 27 33) • For areas unable to access toll free number: +1-408-333-6061 • Toll-free numbers are available in many countries.	support@brocade.com Please include: • Problem summary • Serial number • Installation details • Environment description

Brocade OEM customers

If you have purchased Brocade product support from a Brocade OEM/Solution Provider, contact your OEM/Solution Provider for all of your product support needs.

- OEM/Solution Providers are trained and certified by Brocade to support Brocade® products.
- Brocade provides backline support for issues that cannot be resolved by the OEM/Solution Provider.
- Brocade Supplemental Support augments your existing OEM support contract, providing direct access to Brocade expertise. For more information, contact Brocade or your OEM.
- For questions regarding service levels and response times, contact your OEM/Solution Provider.

Document feedback

To send feedback and report errors in the documentation you can use the feedback form posted with the document or you can e-mail the documentation team.

Quality is our first concern at Brocade and we have made every effort to ensure the accuracy and completeness of this document. However, if you find an error or an omission, or you think that a topic needs further development, we want to hear from you. You can provide feedback in two ways:

- Through the online feedback form in the HTML documents posted on www.brocade.com.
- By sending your feedback to documentation@brocade.com.

Provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.

About This Document

- Audience..... 22
- Supported hardware and software..... 22
- Notice to the reader..... 23
- Related publications..... 24
- How command information is presented in this guide..... 24

Audience

This document is designed for system administrators with a working knowledge of Layer 2 and Layer 3 switching and routing.

If you are using a Brocade device, you should be familiar with the following protocols if applicable to your network - IP, RIP, OSPF, BGP, ISIS, IGMP, PIM, MPLS, and VRRP.

Supported hardware and software

The following hardware platforms are supported by this release of this guide:

TABLE 1 Supported devices

Brocade NetIron XMR Series	Brocade MLX Series	NetIron CES 2000 and NetIron CER 2000 Series
Brocade NetIron XMR 4000	Brocade MLX-4	Brocade NetIron CES 2024C
Brocade NetIron XMR 8000	Brocade MLX-8	Brocade NetIron CES 2024F
Brocade NetIron XMR 16000	Brocade MLX-16	Brocade NetIron CES 2048C
Brocade NetIron XMR 32000	Brocade MLX-32	Brocade NetIron CES 2048CX
	Brocade MLXe-4	Brocade NetIron CES 2048F
	Brocade MLXe-8	Brocade NetIron CES 2048FX
	Brocade MLXe-16	Brocade NetIron CER 2024C
	Brocade MLXe-32	Brocade NetIron CER-RT 2024C
		Brocade NetIron CER 2024F
		Brocade NetIron CER-RT 2024F
		Brocade NetIron CER 2048C
		Brocade NetIron CER-RT 2048C
		Brocade NetIron CER 2048CX
		Brocade NetIron CER-RT 2048CX
		Brocade NetIron CER 2048F
		Brocade NetIron CER-RT 2048F
		Brocade NetIron CER 2048FX
		Brocade NetIron CER-RT 2048FX

Supported software

For the complete list of supported features and the summary of enhancements and configuration notes for this release, refer to the *Multi-Service IronWare R05.7.00 Release Notes* .

Notice to the reader

This document may contain references to the trademarks of the following corporations. These trademarks are the properties of their respective companies and corporations.

These references are made for informational purposes only.

Corporation	Referenced Trademarks and Products
Microsoft Corporation	Internet Explorer
Mozilla Corporation	Mozilla Firefox
Sun Microsystems	Java Runtime Environment

Related publications

For the latest edition of these documents, which contain the most up-to-date information, see Documentation athttp://www.brocade.com/ethernetproducts

- *Multi-Service IronWare Administration Guide*
- *Multi-Service IronWare Security Configuration Guide*
- *Multi-Service IronWare Switching Configuration Guide*
- *Multi-Service IronWare Routing Configuration Guide*
- *Multi-Service IronWare Traffic Management Configuration Guide*
- *Multi-Service IronWare Multicast Configuration Guide*
- *Multi-Service IronWare Multiprotocol Label Switch (MPLS) Configuration Guide*
- *Multi-Service IronWare Software Defined Networking (SDN) Guide*
- *Brocade MLX Series and NetIron Family YANG Guide*
- *Brocade MLX Series and NetIron XMR Series Diagnostic Reference*
- *Unified IP MIB Reference*
- *Multi-Service IronWare Software Upgrade Procedures for Brocade MLX Series and NetIron Family devices*
- *Brocade MLXe Series Installation Guide*
- *Brocade MLX Series and Brocade NetIron XMR Installation Guide*
- *Brocade NetIron CES 2000 Series and Brocade NetIron CER 2000 Series Hardware Installation Guide*

How command information is presented in this guide

For all new content, command syntax and parameters are documented in a separate command reference section at the end of the publication.

In an effort to provide consistent command line interface (CLI) documentation for all products, Brocade is in the process of preparing standalone Command References for the IP platforms. This process involves separating command syntax and parameter descriptions from configuration tasks. Until this process is completed, command information is presented in two ways:

- For all new content included in this guide, the CLI is documented in separate command pages. The new command pages follow a standard format to present syntax, parameters, usage guidelines, examples, and command history. Command pages are compiled in alphabetical order in a separate command reference chapter at the end of the publication.
- Legacy content continues to include command syntax and parameter descriptions in the chapters where the features are documented.

If you do not find command syntax information embedded in a configuration task, refer to the command reference section at the end of this publication for information on CLI syntax and usage.

Configuring BGP4 (IPv4)

- BGP4 feature support 25
- BGP4 overview 28
- Implementation of BGP4 33
- BGP4 restart 33
- Basic configuration and activation for BGP4 36
- BGP4 parameters 38
- Memory considerations 40
- Basic configuration tasks required for BGP4 40
- Optional BGP4 configuration tasks 55
- Configuring BGP4 restart 74
- Modifying redistribution parameters 78
- Filtering 81
- Four-byte Autonomous System Numbers (AS4) 98
- BGP4 AS4 attribute errors 103
- Configuring route flap dampening 104
- Generating traps for BGP4 109
- Configuring BGP4 109
- Entering and exiting the address family configuration level 111
- BGP route reflector 112
- Specifying a maximum AS path length 115
- BGP4 max-as error messages 116
- Originating the default route 117
- Changing the default metric used for route cost 117
- Configuring a static BGP4 network 117
- Generalized TTL Security Mechanism support 123
- show metro mp-vlp-queue 125
- clear metro mp-vlp-queue 127
- Displaying BGP4 information 127
- Clearing traffic counters 169
- Clearing diagnostic buffers 170

BGP4 feature support

Documents software features supporting BGP4.

The following features are documented in the *Multi-Service IronWare Routing Configuration Guide* .

Feature	Brocade NetIron XMR Series	Brocade NetIron MLX Series	Brocade NetIron CES 2000 Series BASE package	Brocade NetIron CES 2000 Series ME_PREM package	Brocade NetIron CES 2000 Series L3_PREM package	Brocade NetIron CER 2000 Series Base package	Brocade NetIron CER 2000 Series Advanced Services package
BGP4	Yes	Yes	No	No	Yes	Yes	Yes
BGP4 Restart	Yes	Yes	No	No	Yes	Yes	Yes
BGP4 Restart helper mode	Yes	Yes	No	No	Yes	Yes	Yes
Redistributing IBGP routes	Yes	Yes	No	No	Yes	Yes	Yes
Client-to-Client Routes	Yes	Yes	No	No	Yes	Yes	Yes
Route Flap Dampening	Yes	Yes	No	No	Yes	Yes	Yes
Originating the Default Route	Yes	Yes	No	No	Yes	Yes	Yes
Multipath Load Sharing	Yes	Yes	No	No	Yes	Yes	Yes
Using the IP Default Route as a Valid Next Hop for a BGP4 Route	Yes	Yes	No	No	Yes	Yes	Yes
Next-Hop Recursion	Yes	Yes	No	No	Yes	Yes	Yes
Next-Hop Update Timer	Yes	Yes	No	No	Yes	Yes	Yes
Generalized TTL Security Mechanism Support	Yes	Yes	No	No	Yes	Yes	Yes
Enhanced per-neighbor debug statements and new per-neighbor BGP4 debug filters	Yes	Yes	No	No	No	No	No
BGP4 Peer Notification During a Management Module Switchover	Yes	Yes	No	No	No	No	No
Auto Shutdown of BGP4 neighbors on Initial Configuration	Yes	Yes	No	No	Yes	Yes	Yes
New encryption code for passwords, authentication keys, and community strings	Yes	Yes	No	No	No	No	No
BGP4 MD5 Authentication	Yes	Yes	No	No	Yes	Yes	Yes
Route redistribution to other protocols	Yes	Yes	No	No	Yes	Yes	Yes

Feature	Brocade NetIron XMR Series	Brocade NetIron MLX Series	Brocade NetIron CES 2000 Series BASE package	Brocade NetIron CES 2000 Series ME_PREM package	Brocade NetIron CES 2000 Series L3_PREM package	Brocade NetIron CER 2000 Series Base package	Brocade NetIron CER 2000 Series Advanced Services package
BGP4 Peer Group	Yes	Yes	No	No	Yes	Yes	Yes
BGP4 Route Reflectors	Yes	Yes	No	No	Yes	Yes	Yes
BGP4 Neighbor Local-AS	Yes	Yes	No	No	Yes	Yes	Yes
BGP4 Local-AS for VRF	Yes	Yes	No	No	Yes	No	No
BGP4 Processing Optimization for Administratively Down Peers	Yes	Yes	No	No	Yes	Yes	Yes
BGP4 Outbound Policy Processing Optimization	Yes	Yes	No	No	Yes	Yes	Yes
Generalized TTL Security Mechanism Support	Yes	Yes	No	No	Yes	Yes	Yes
Enhanced per-neighbor debug statements and new per-neighbor BGP4 debug filters	Yes	Yes	No	No	Yes	Yes	Yes
Requiring the First AS to be the Neighbor's AS	Yes	Yes	No	No	Yes	Yes	Yes
Four-byte AS Numbers (AS4)	Yes	Yes	No	No	Yes	Yes	Yes
BGP4 AS4 Confederation Error Checking	Yes	Yes	No	No	No	No	No
RTM Scalability Enhancement	Yes	Yes	No	No	No	No	No
Route Map Continue Clause	Yes	Yes	No	No	Yes	Yes	Yes
Static BGP4 Networks	Yes	Yes	No	No	Yes	Yes	Yes
Limiting Advertisement of a Static BGP4 Network	Yes	Yes	No	No	Yes	Yes	Yes
Use IGP cost instead of BGP MED value	Yes	Yes	No	No	Yes	Yes	Yes
Medium priority MP-VLP queue support for BGP packets in CER	No	No	Yes	Yes	Yes	Yes	Yes

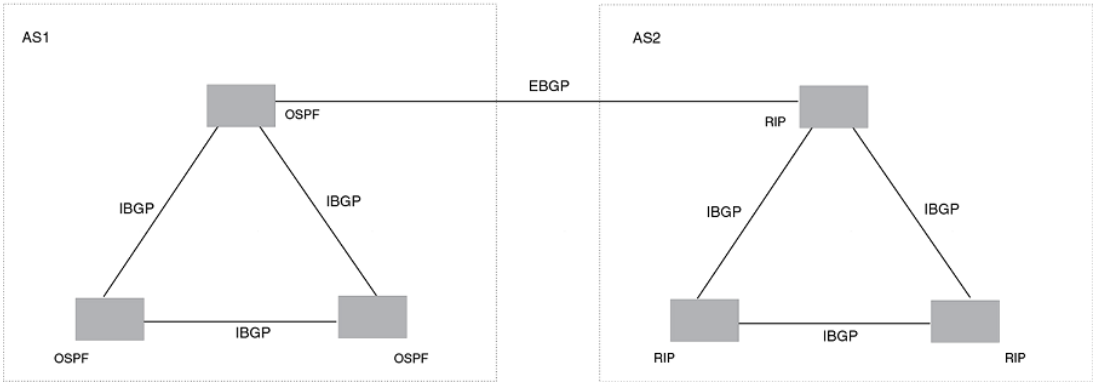
BGP4 overview

BGP4 is the standard Exterior Gateway Protocol (EGP) used on the Internet to route traffic between Autonomous Systems (AS) and to maintain loop-free routing. An AS is a collection of networks that share the same routing and administration characteristics. For example, a corporate Intranet consisting of several networks under common administrative control might be considered an AS. The networks in an AS can but do not need to run the same routing protocol to be in the same AS, nor do they need to be geographically close.

Devices within an AS can use different Interior Gateway Protocols (IGPs) such as RIP, IS-IS, and OSPF to communicate with one another. However, for devices in different autonomous systems to communicate, they need to use an EGP. BGP4 is the standard EGP used by Internet devices and therefore is the EGP implemented on Brocade devices.

This is a simple example of two BGP4 ASs. Each AS contains three BGP4 devices. All of the BGP4 devices within an AS communicate using IBGP. BGP4 devices communicate with other autonomous systems using EBGP. Notice that each of the devices also is running an Interior Gateway Protocol (IGP). The devices in AS1 are running OSPF and the devices in AS2 are running RIP. The device can be configured to redistribute routes among BGP4, IS-IS, RIP, and OSPF. They also can redistribute static routes.

FIGURE 1 Example BGP4 autonomous systems



Relationship between the BGP4 route table and the IP route table

The device BGP4 route table can have multiple routes or paths to the same destination, which are learned from different BGP4 neighbors. A BGP4 neighbor is another device that also is running BGP4. BGP4 neighbors communicate using Transmission Control Protocol (TCP) port 179 for BGP4 communication. When you configure the device for BGP4, one of the configuration tasks you perform is to identify the device's BGP4 neighbors.

Although a device's BGP4 route table can have multiple routes to the same destination, the BGP4 protocol evaluates the routes and chooses only one to send to the IP route table. The route that BGP4 chooses and sends to the IP route table is the preferred route . This route is what the device advertises to other BGP4 neighbors. If the preferred route goes down, BGP4 updates the route information in the IP route table with a new BGP4 preferred route.

NOTE
If IP load sharing is enabled and you enable multiple equal-cost paths for BGP4, BGP4 can select more than one equal-cost path to a destination.

A BGP4 route consists of the following information:

- Network number (prefix) - A value made up of the network mask bits and an IP address; for example, 10.215.129.0/18 indicates a network mask of 18 bits applied to the IP address 10.215.129.0. When a BGP4 device advertises a route to one of its neighbors, it uses this format.
- AS-path - A list of the other autonomous systems through which a route passes. BGP4 devices can use the AS-path to detect and eliminate routing loops. For example, if a route received by a BGP4 device contains the AS that the device is in, the device does not add the route to its own BGP4 table. (The BGP4 RFCs refer to the AS-path as "AS_PATH", and RFC 4893 uses "AS4_PATH" in relation to AS4s.)
- Additional path attributes - A list of additional parameters that describe the route. The route MED and next hop are examples of these additional path attributes.

NOTE
The device re-advertises a learned best BGP4 route to the device's neighbors even when the software does not select that route for installation in the IP route table. This can happen if a route from another protocol, for example, OSPF, is preferred. The best BGP4 route is the route that BGP4 selects based on comparison of the BGP4 route path's attributes.

After a device successfully negotiates a BGP4 session with a neighbor (a BGP4 peer), the device exchanges complete BGP4 route tables with the neighbor. After this initial exchange, the device and all other RFC 1771-compliant BGP4 devices send UPDATE messages to inform neighbors of new, changed, or no longer feasible routes. BGP4 devices do not send regular updates. However, if configured to do so, a BGP4 device does regularly send KEEPALIVE messages to its peers to maintain BGP4 sessions with them if the device does not have any route information to send in an UPDATE message.

How BGP4 selects a path for a route (BGP best path selection algorithm)

When multiple paths for the same route prefix are known to a BGP4 device, the device uses the following algorithm to weigh the paths and determine the optimal path for the route. The optimal path depends on various parameters, which can be modified.

1. Is the next hop accessible though an Interior Gateway Protocol (IGP) route? If not, ignore the route.

   NOTE
   The device does not use the default route to resolve BGP4 next hop.

2. Use the path with the largest weight.
3. If the weights are the same, prefer the path with the largest local preference.
4. Prefer the route that was originated locally (by this BGP4 device).
5. If the local preferences are the same, prefer the path with the shortest AS-path. An AS-SET counts as 1. A confederation path length, if present, is not counted as part of the path length.

NOTE
This step can be skipped if **BGP4-as-path-ignore** is configured.

6. If the AS-path lengths are the same, prefer the path with the lowest origin type. From low to high, route origin types are valued as follows:
   - IGP is lowest.
   - EGP is higher than IGP but lower than INCOMPLETE.
   - INCOMPLETE is highest.
7. If the paths have the same origin type, prefer the path with the lowest MED.

   If the routes were learned from the same neighboring AS, BGP4 compares the MEDs of two otherwise equivalent paths. This behavior is called deterministic MED. Deterministic MED is always enabled and cannot be disabled. You can also enable the device to always compare the MEDs, regardless of the AS information in the paths. To enable this comparison, enter the **always-compare-med** command at the BGP4 configuration level of the CLI. This option is disabled by default.

NOTE
By default, value 0 (most favorable) is used in MED comparison when the MED attribute is not present. The default MED comparison results in the device favoring the route paths that are missing their MEDs. You can use the **med-missing-as-worst** command to make the device regard a BGP4 route with a missing MED attribute as the least favorable path, when comparing the MEDs of the route paths.

NOTE
MED comparison is not performed for internal routes originated within the local AS or confederation unless the **compare-med-empty-aspath** command is configured.

8. Prefer routes in the following order:
   - Routes received through EBGP from a BGP4 neighbor outside of the confederation
   - Routes received through EBGP from a BGP4 device within the confederation OR Routes received through IBGP.
9. If all the comparisons above are equal, prefer the route with the lowest IGP metric to the BGP4 next hop. This is the closest internal path inside the AS to reach the destination.
10. If the internal paths also are the same and BGP4 load sharing is enabled, load share among the paths. Otherwise prefer the route that comes from the BGP4 device with the lowest device ID.

NOTE
Brocade devices support BGP4 load sharing among multiple equal-cost paths. BGP4 load sharing enables the device to balance traffic across the multiple paths instead of choosing just one path based on device ID. For EBGP routes, load sharing applies only when the paths are from neighbors within the same remote AS. EBGP paths from neighbors in different autonomous systems are not compared, unless multipath **multi-as** is enabled.

11. If the **compare-router ID** is enabled, prefer the path that comes from the BGP4 device with the lowest device ID. If a path contains originator ID attributes, then originator ID is substituted for the ROUTER ID in the decision.
12. Prefer the path with the minimum cluster list length.
13. Prefer the route that comes from the lowest BGP4 neighbor address.

BGP4 message types

BGP4 devices communicate with neighbors (other BGP4 devices) using the following types of messages:

- OPEN
- UPDATE
- KEEPALIVE
- NOTIFICATION
- ROUTE REFRESH

OPEN message

After a BGP4 device establishes a TCP connection with a neighboring BGP4 device, the devices exchange OPEN messages. An open message indicates the following:

- BGP4 version - Indicates the version of the protocol that is in use on the device. BGP4 version 4 supports Classless Interdomain Routing (CIDR) and is the version most widely used in the Internet. Version 4 also is the only version supported on devices.
- AS number - An autonomous system number (ASN) identifies the AS to which the BGP4 device belongs. The number can be up to four bytes.
- Hold Time - The number of seconds a BGP4 device will wait for an UPDATE or KEEPALIVE message (described below) from a BGP4 neighbor before assuming that the neighbor is not operational. BGP4 devices exchange UPDATE and KEEPALIVE messages to update route information and maintain communication. If BGP4 neighbors are using different Hold Times, the lowest Hold Time is used by the neighbors. If the Hold Time expires, the BGP4 device closes the TCP connection to the neighbor and clears any information it has learned and cached from the neighbor.

  You can configure the Hold Time to be 0, in which case a BGP4 device will consider neighbors to always be up. For directly-attached neighbors, you can configure the device to immediately close the TCP connection to the neighbor and clear entries learned from an EBGP neighbor if the interface to that neighbor goes down. This capability is provided by the fast external fail over feature, which is disabled by default.
- BGP4 Identifier - The device ID. The BGP4 Identifier (device ID) identifies the BGP4 device to other BGP4 devices. The device use the same device ID for OSPF and BGP4. If you do not set a device ID, the software uses the IP address on the lowest numbered loopback interface configured on the device. If the device does not have a loopback interface, the default device ID is the lowest numbered IP address configured on the device.
- Parameter list - An optional list of additional parameters used in peer negotiation with BGP4 neighbors.

UPDATE message

After BGP4 neighbors establish a BGP4 connection over TCP and exchange their BGP4 routing tables, they do not send periodic routing updates. Instead, a BGP4 neighbor sends an update to a neighbor when it has a new route to advertise or routes have changed or become unfeasible. An UPDATE message can contain the following information:

- Network Layer Reachability Information (NLRI) - The mechanism by which BGP4 supports Classless Interdomain Routing (CIDR). An NLRI entry consists of an IP prefix that indicates a network being advertised by the UPDATE message. The prefix consists of an IP network number and the length of the network portion of the number. For example, an UPDATE message with the NLRI entry

10.215.129.0/18 indicates a route to IP network 10.215.129.0 with network mask 255.255.192.0. The binary equivalent of this mask is 18 consecutive one bits, thus "18" in the NLRI entry.

- Path attributes - Parameters that indicate route-specific information such as Autonomous System path information, route preference, next hop values, and aggregation information. BGP4 uses path attributes to make filtering and routing decisions.
- Unreachable routes - A list of routes that have been in the sending device BGP4 table but are no longer feasible. The UPDATE message lists unreachable routes in the same format as new routes: *IP address* and *CIDR prefix* .

KEEPALIVE message

BGP4 devices do not regularly exchange UPDATE messages to maintain BGP4 sessions. For example, if a device configured to perform BGP4 routing has already sent the latest route information to peers in UPDATE messages, the device does not send more UPDATE messages. Instead, BGP4 devices send KEEPALIVE messages to maintain BGP4 sessions. KEEPALIVE messages are 19 bytes long and consist only of a message header. They do not contain routing data.

BGP4 devices send KEEPALIVE messages at a regular interval, called the Keep Alive Time. The default Keep Alive Time is 60 seconds.

A parameter related to the Keep Alive Time is the Hold Time. The Hold Time for a BGP4 device determines how many seconds the device waits for a KEEPALIVE or UPDATE message from a BGP4 neighbor before deciding that the neighbor is not operational. The Hold Time is negotiated when BGP4 devices exchange OPEN messages, the lower Hold Time is then used by both neighbors. For example, if BGP4 device A sends a Hold Time of 5 seconds and BGP4 device B sends a Hold Time of 4 seconds, both devices use 4 seconds as the Hold Time for their BGP4 session. The default Hold Time is 180 seconds. Generally, the Hold Time is configured to three times the value of the Keep Alive Time.

If the Hold Time is 0, a BGP4 device assumes that a neighbor is alive regardless of how many seconds pass between receipt of UPDATE or KEEPALIVE messages.

NOTIFICATION message

When you close the BGP4 session with a neighbor, the device detects an error in a message received from the neighbor, or an error occurs on the device, the device sends a NOTIFICATION message to the neighbor. No further communication takes place between the BGP4 device that sent the NOTIFICATION and the neighbors that received the NOTIFICATION.

REFRESH message

BGP4 sends a REFRESH message to a neighbor to request that the neighbor resend route updates. This type of message can be useful if an inbound route filtering policy has been changed.

Grouping of RIB-out peers

To improve efficiency in the calculation of outbound route filters, the device groups BGP4 peers together based on their outbound policies. To reduce RIB-out memory usage, the device then groups the peers within an outbound policy group according to their RIB-out routes. All peers sharing a single RIB-out route (up to 32 peers per group) also share a single physical RIB-out entry, resulting in as much as a 30-fold memory usage reduction.

NOTE
RIB-out peer grouping is not shared between different VRFs or address families, and is not supported for VPNV4 or L2VPN peers.

Implementation of BGP4

BGP4 is described in RFC 1771 and the latest BGP4 drafts. The Brocade BGP4 implementation fully complies with RFC 1771. Brocade BGP4 implementation also supports the following RFCs:

- RFC 1745 (OSPF Interactions)
- RFC 1997 (BGP Communities Attributes)
- RFC 2385 (TCP MD5 Signature Option)
- RFC 2439 (Route Flap Dampening)
- RFC 2796 (Route Reflection)
- RFC 2842 (Capability Advertisement)
- RFC 3065 (BGP4 Confederations)
- RFC 2858 (Multiprotocol Extensions)
- RFC 2918 (Route Refresh Capability)
- RFC 3392 (BGP4 Capability Advertisement)
- Draft-ietf-idr-restart-10.txt (restart mechanism for BGP4)

BGP4 restart

BGP4 restart is a high-availability routing feature that minimizes disruption in traffic forwarding, diminishes route flapping, and provides continuous service during a system restart, switchover, failover, or hitless OS upgrade. During such events, routes remain available between devices. BGP4 restart operates between a device and its peers, and must be configured on each participating device.

Under normal operation, when a BGP4 device is restarted, the network is automatically reconfigured. Routes available through the restarting device are deleted when the device goes down, and are then rediscovered and added back to the routing tables when the device is back up and running. In a network with devices that regularly restart, performance can degrade significantly and limit the availability of network resources. BGP4 restart dampens the network response and limits route flapping by allowing routes to remain available between devices during a restart. BGP4 restart operates between a device and peers, and must be configured on each participating device.

A BGP4 restart-enabled device advertises the capability to establish peering relationships with other devices. When a restart begins, neighbor devices mark all of the routes from the restarting device as stale, but continue to use the routes for the length of time specified by the restart timer. After the device is restarted, it begins to receive routing updates from the peers. When it receives the end-of-RIB marker that indicates it has received all of the BGP4 route updates, it recomputes the new routes and replaces the stale routes in the route map with the newly computed routes. If the device does not come back up within the time configured for the purge timer, the stale routes are removed.

NOTE
A second management module must be installed for the device to function as a restart device. If the device functions as a restart helper device only, there is no requirement for a secondary management module.

The implementation of BGP4 Restart supports the following Internet Draft:

- Draft-ietf-idr-restart-10.txt: restart mechanism for BGP4

BGP4 Peer notification during a management module switchover

The BGP4 Peer notification process restores BGP4 adjacency quickly and allows packet forwarding between the newly active management module and the BGP4 peers. The handling of TCP packets with an MD5 digest prevents the silent dropping of TCP packets without triggering a RESET packet.

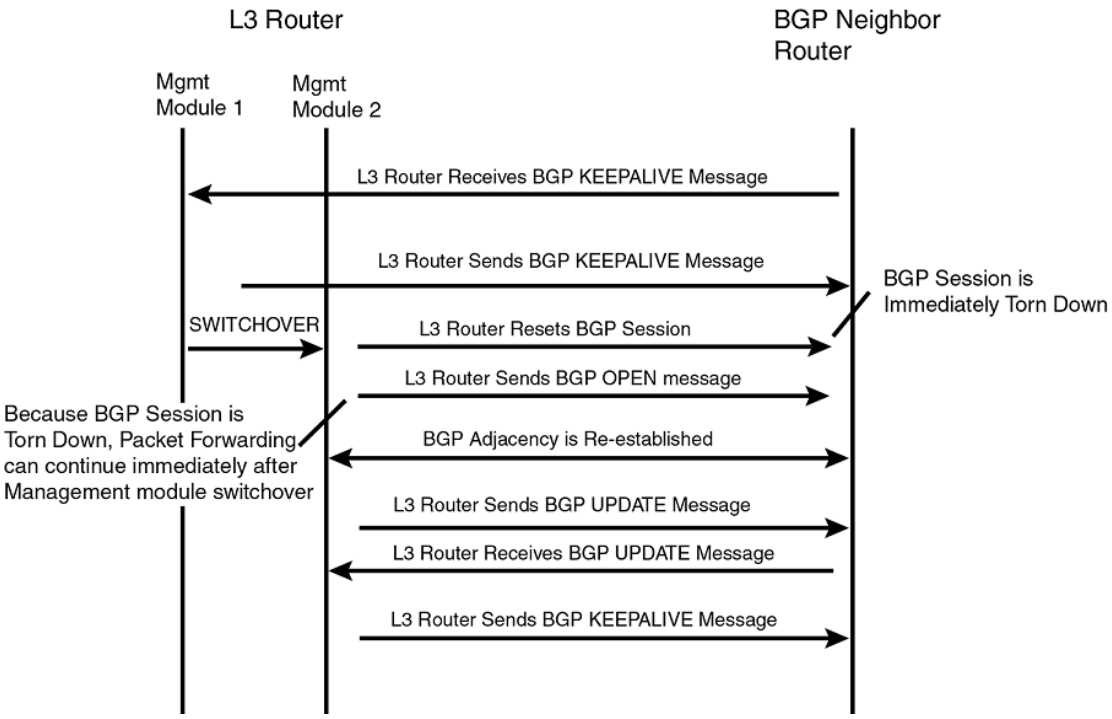
The BGP4 peer notification process operates effectively when implemented for the following processes that involve the intentional switching of the active status from one management module to another:

- System Reload - When a device undergoes the reload process, both management modules and all interface modules are rebooted. All BGP4 sessions are terminated BEFORE the system triggers the hardware reset.
- Switchover Requested by User - Switching over to a standby management module can be triggered by the **switchover** , **reset** , **reload** , and **hitless-reload** commands. When these commands are executed, the active management module resets the BGP4/TCP sessions with BGP4 neighbors before transferring control to the standby management module.

NOTE
Restart-enabled BGP4 sessions are not reset. The BGP4 restart protocol allows a BGP4 session to reconnect gracefully without going through the normal process.

This example describes the procedure used between the management modules in a device and a BGP4 neighbor device.

FIGURE 2 Management module switchover behavior for BGP4 peer notification



If the active management module fails due to a fault, the management module does not have the opportunity to reset BGP4 sessions with neighbors as described for intentional failovers. In this situation the management module will reboot, or the standby management module becomes the new active management module. Since the new active management module does not have the TCP/BGP4 information needed to reset the previous sessions, a remote BGP4 peer session is only reset when it sends a BGP4/TCP keep-alive packet to this device, or when the BGP4 hold-time expires.

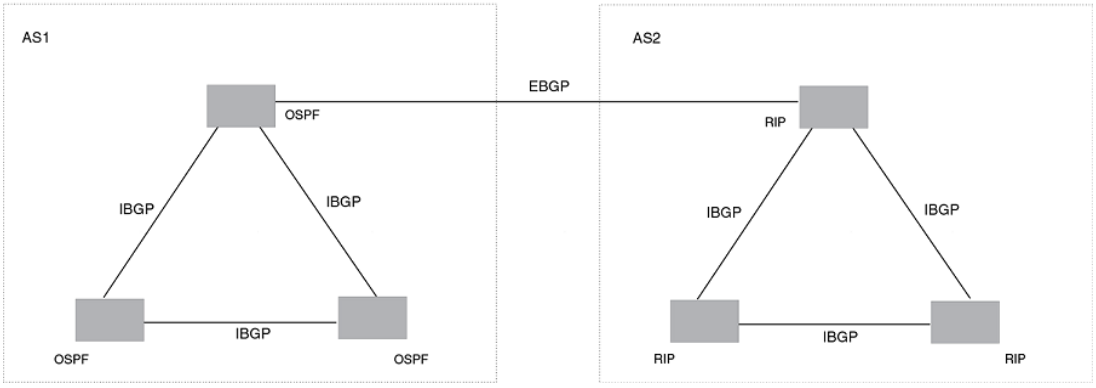
To help reduce the reconnection time after a management module failover or system reload, if an incoming TCP packet contains an MD5 digest, and no matching TCP session is found, the device attempts to find a matching BGP4 peer based on the IP address. If a BGP4 peer configuration can be found, the device looks up the MD5 password configured for the peer, and uses it to send a RESET packet.

BGP4 neighbor local AS

This feature allows you to configure a device so that it adds a peer to an AS that is different from the AS to which it actually belongs. This feature is useful when an ISP is acquired by another ISP. In this situation, customers of the acquired ISP might not want to (or might not be able to) adjust their configuration to connect to the AS of the acquiring provider.

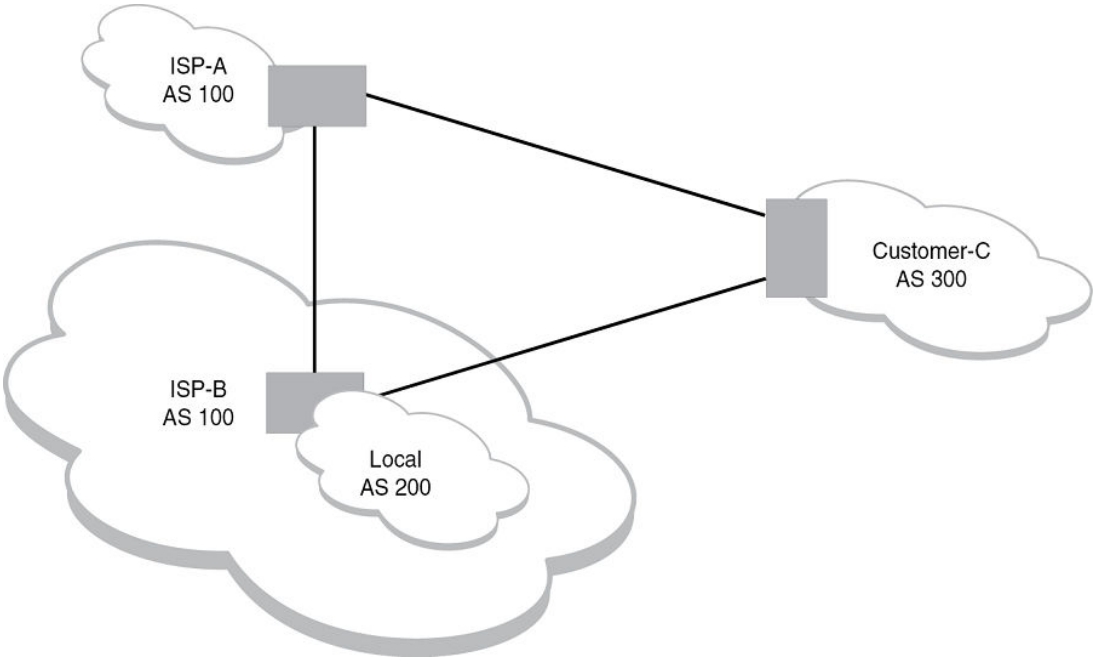
In this example, Customer C is connected to ISP-A which is in AS 100 and ISP-B which is in AS 200.

FIGURE 3 Example of customer connected to two ISPs



In the next example, ISP-A has purchased ISP-B. The AS associated with ISP-B changes to AS 100. If Customer C cannot or does not want to change their configuration or peering relationship with ISP-B, a peer with Local-AS configured with the value 200 can be established on ISP-B.

FIGURE 4 Example of Local AS configured on ISP-B



A Local AS is configured using the BGP4 **neighbor** command. To confirm that a Local AS has been configured, use the **show ip bgp neighbors** command.

Basic configuration and activation for BGP4

BGP4 is disabled by default. Follow the steps below to enable BGP4.

1. Enable the BGP4 protocol.
2. Set the local AS number.

NOTE
You must specify the local AS number for BGP4 to become functional.

3. Add each BGP4 neighbor (peer BGP4 device) and identify the AS the neighbor is in.
4. Save the BGP4 configuration information to the system configuration file.

For example, enter commands such as the following.

```
device> enable
device# configure terminal
device(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
device(config-bgp)# local-as 10
device(config-bgp)# write memory
```

Syntax: router bgp

The **router bgp** command enables the BGP4 protocol.

NOTE
By default, the Brocade device ID is the IP address configured on the lowest numbered loopback interface. If the device does not have a loopback interface, the default device ID is the lowest numbered IP interface address configured on the device. If you change the device ID, all current BGP4 sessions, OSPF adjacencies, and OSPFv3 adjacencies are cleared.

NOTE
When BGP4 is enabled on a Brocade device, you do not need to reset the system. The protocol is activated as soon as you enable it. The device begins a BGP4 session with a BGP4 neighbor when you add the neighbor.

Disabling BGP4

If you disable BGP4, the device removes all the running configuration information for the disabled protocol from the running configuration. To restore the BGP4 configuration, you must reload the software to load the BGP4 configuration from the startup configuration. When you save the startup configuration file after disabling the protocol, all of the BGP4 configuration information for the disabled protocol is removed from the startup configuration file.

The CLI displays a warning message such as the following.

```
device(config)# no router bgp
router bgp mode now disabled and runtime configuration is erased. All bgp config data
will be lost when writing to flash!
```

The Web Management Interface does not display a warning message.

If you are testing a BGP4 configuration and need to disable and re-enable the protocol, you should make a backup copy of the startup configuration file containing the BGP4 configuration information. If you remove the configuration information by saving the configuration after disabling the protocol, you can restore the BGP4 configuration by copying the backup copy of the startup configuration file onto the flash memory.

NOTE
To disable BGP4 without losing the BGP4 configuration information, remove the local AS (for example, by entering the **no local-as** command). When you remove the local AS, BGP4 retains the other configuration information but will not become operational until you reset the local AS.

BGP4 parameters

You can modify or set the following BGP4 parameters:

- Optional - Define the router ID. (The same router ID also is used by OSPF.)
- Required - Specify the local AS number.
- Optional - Add a loopback interface for use with neighbors.
- Required - Identify BGP4 neighbors.
- Optional - Change the Keep Alive Time and Hold Time.
- Optional - Change the update timer for route changes.
- Optional - Enable fast external fallover.
- Optional - Specify a list of individual networks in the local AS to be advertised to remote autonomous systems using BGP4.
- Optional - Change the default local preference for routes.
- Optional - Enable the default route (default-information-originate).
- Optional - Enable use of a default route to resolve a BGP4 next-hop route.
- Optional - Change the default MED (metric).
- Optional - Enable next-hop recursion.
- Optional - Change the default administrative distances for EBGP, IBGP, and locally originated routes.
- Optional - Require the first AS in an Update from an EBGP neighbor to be the neighbor AS.
- Optional - Change MED comparison parameters.
- Optional - Disable comparison of the AS-Path length.
- Optional - Enable comparison of the device ID.
- Optional - Enable auto summary to summarize routes at an IP class boundary (A, B, or C).
- Optional - Aggregate routes in the BGP4 route table into CIDR blocks.
- Optional - Configure the device as a BGP4 route reflector.
- Optional - Configure the device as a member of a BGP4 confederation.
- Optional - Change the default metric for routes that BGP4 redistributes into RIP or OSPF.
- Optional - Change the parameters for RIP, OSPF, or static routes redistributed into BGP4.
- Optional - Change the number of paths for BGP4 load sharing.
- Optional - Change other load-sharing parameters
- Optional - Define BGP4 address filters.
- Optional - Define BGP4 AS-path filters.
- Optional - Define BGP4 community filters.
- Optional - Define IP prefix lists.
- Optional - Define neighbor distribute lists.
- Optional - Define BGP4 route maps for filtering routes redistributed into RIP and OSPF.
- Optional - Define route flap dampening parameters.

NOTE
When using the CLI, you set global level parameters at the BGP CONFIG level of the CLI. You can reach the BGP CONFIG level by entering the **router bgp** command at the global CONFIG level.

Some parameter changes take effect immediately while others do not take full effect until the device sessions with its neighbors are reset. Some parameters do not take effect until the device is rebooted.

Parameter changes that take effect immediately

The following parameter changes take effect immediately:

- Enable or disable BGP4.
- Set or change the local AS.
- Add neighbors.
- Change the update timer for route changes.
- Disable or enable fast external failover.
- Specify individual networks that can be advertised.
- Change the default local preference, default information originate setting, or administrative distance.
- Enable or disable use of a default route to resolve a BGP4 next-hop route.
- Enable or disable MED (metric) comparison.
- Require the first AS in an update from an EBGP neighbor to be the neighbor AS.
- Change MED comparison parameters.
- Disable comparison of the AS-Path length.
- Enable comparison of the device ID.
- Enable next-hop recursion.
- Change the default metric.
- Disable or re-enable route reflection.
- Configure confederation parameters.
- Disable or re-enable load sharing.
- Change the maximum number of load sharing paths.
- Change other load-sharing parameters.
- Define route flap dampening parameters.
- Add, change, or negate redistribution parameters (except changing the default MED as described in Changing the default MED (Metric) used for route redistribution on page 61).
- Add, change, or negate route maps (when used by the **network** command or a redistribution command).
- Aggregate routes.
- Apply maximum AS path limit settings for UPDATE messages.

Parameter changes that take effect after resetting neighbor sessions

The following parameter changes take effect only after the BGP4 sessions on the device are cleared, or reset using the "soft" clear option:

- Change the Hold Time or Keep Alive Time.
- Aggregate routes
- Add, change, or negate filter tables that affect inbound and outbound route policies.
- Apply maximum AS path limit settings to the RIB.

Parameter changes that take effect after disabling and re-enabling redistribution

The following parameter change takes effect only after you disable and then re-enable redistribution:

- Change the default MED (metric).

Memory considerations

BGP4 can handle a very large number of routes and therefore requires a lot of memory. For example, in a typical configuration with a single BGP4 neighbor, receiving a full internet route table, a BGP4 device may need to hold up to millions of route updates. Many configurations, especially those involving more than one neighbor, can require the device to hold even more routes. Brocade devices provide dynamic memory allocation for BGP4 data. BGP4 devices automatically allocate memory when needed to support BGP4 neighbors, routes and route attribute entries. Dynamic memory allocation is performed automatically by the software and does not require a reload.

As a guideline, a device with a 2 GB Management module can accommodate 150 – 200 neighbors, with the assumption that the device receives about one million routes total from all neighbors and sends about eight million routes total to neighbors. For each additional one million incoming routes, the capacity for outgoing routes decreases by about two million.

Basic configuration tasks required for BGP4

The following sections describe how to perform the configuration tasks that are required to use BGP4 on the Brocade device.

Enabling BGP4 on the device

When you enable BGP4 on the device, BGP4 is automatically activated. To enable BGP4 on the device, enter the following commands.

```
device# configure terminal
device(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
device(config-bgp-router)# local-as 10
device(config-bgp-router)# neighbor 10.157.23.99 remote-as 100
device(config-bgp-router)# write memory
```

Changing the device ID

The OSPF and BGP4 protocols use device IDs to identify devices that are running the protocols. A device ID is a valid, unique IP address and sometimes is an IP address configured on the device. The device ID cannot be an IP address in use by another device.

By default, the device ID on a Brocade device is one of the following:

- If the device has loopback interfaces, the default device ID is the IP address on the lowest numbered loopback interface configured on the Brocade device. For example, if you configure loopback interfaces 1, 2, and 3 as follows, the default device ID is 10.9.9.9/24:

  - Loopback interface 1, 10.9.9.9/24
  - Loopback interface 2, 10.4.4.4/24
  - Loopback interface 3, 10.1.1.1/24
- If the device does not have any loopback interfaces, the default device ID is the lowest numbered IP interface address configured on the device.

NOTE
Brocade devices use the same device ID for both OSPF and BGP4. If the device is already configured for OSPF, you may want to use the device ID already assigned to the device rather than set a new one. To display the current device ID, enter the **show ip** command at any CLI level.

To change the device ID, enter a command such as the following.

```
device(config)# ip router-id 10.157.22.26
```

Syntax: [no] ip router-id *ip-addr*

The *ip-addr* can be any valid, unique IP address.

NOTE
You can specify an IP address used for an interface on the Brocade device, but do not specify an IP address that is being used by another device.

Setting the local AS number

The local autonomous system number (ASN) identifies the AS in which the Brocade BGP4 device resides.

To set the local AS number, enter commands such as the following.

```
device(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
device(config-bgp)# local-as 10
device(config-bgp)# write memory
```

Syntax: [no] local-as *num*

The *num* parameter specifies a local AS number in the range 1 through 4294967295. It has no default. AS numbers 64512 - 65535 are the well-known private BGP4 AS numbers and are not advertised to the Internet community.

Setting the local AS number for VRF instances

The local autonomous system (AS) number identifies the AS in which the BGP4 device resides.

In releases prior to NetIron R05.3.00, you can assign only a single BGP AS number for the entire system, including instances of BGP VRF. In NetIron R05.3.00, you can assign different BGP AS numbers for each VRF instance. If you do not assign an AS number, the BGP VRF instances use the default BGP AS number, as in previous releases.

The **local-as** command is available under the "global BGP" CLI level and "address- family ipv4 unicast vrf" CLI level.

To set the local as number for a VRF, enter commands such as the following.

```
device(config-bgp)# address-family ipv4 unicast vrf vrf-name
device(config-bgp)# local-as num
```

Syntax: [no] local-as *num*

The *num* parameter specifies a local AS number in the range 1 - 4294967295. It has no default. AS numbers 64512 - 65535 are the well-known private BGP4 AS numbers and are not advertised to the Internet community.

The configuration takes effect immediately and the BGP VRF instance is reset. All BGP peering within the VRF is reset, and take the new AS number.

The local AS number for the VRF instance, if configured, is displayed in the **show running-config** and **show ip bgp config** command output.

Enter the **show ip bgp config** command:

```
device# show ip bgp config
Current BGP configuration:
router bgp
 local-as 100
 neighbor 10.10.10.10 remote-as 200

 address-family ipv4 unicast
 exit-address-family

 address-family ipv4 multicast
 exit-address-family

 address-family ipv6 unicast
 exit-address-family

 address-family ipv6 multicast
 exit-address-family

 address-family vpnv4 unicast
 exit-address-family

 address-family l2vpn vpls
 exit-address-family

address-family ipv4 unicast vrf vrf_a
 local-as 300
 neighbor 10.111.111.111 remote-as 400
 exit-address-family
```

Adding a loopback interface

You can configure the device to use a loopback interface instead of a specific port or virtual routing interface to communicate with a BGP4 neighbor. A loopback interface adds stability to the network by working around route flap problems that can occur due to unstable links between the device and neighbors.

Loopback interfaces are always up, regardless of the states of physical interfaces. Loopback interfaces are especially useful for IBGP neighbors (neighbors in the same AS) that are multiple hops away from the device. When you configure a BGP4 neighbor on the device, you can specify whether the device uses the loopback interface to communicate with the neighbor. As long as a path exists between the device and the neighbor, BGP4 information can be exchanged. The BGP4 session is not associated with a specific link, but is instead associated with the virtual interfaces.

NOTE
If you configure the Brocade device to use a loopback interface to communicate with a BGP4 neighbor, the peer IP address on the remote device pointing to your loopback address must be configured.

To add a loopback interface, enter commands such as the following.

```
device(config-bgp)# exit
device(config)# int loopback 1
device(config-lbif-1)# ip address 10.0.0.1/24
```

Syntax: [no] interface loopback *num*

The *num* value can be from 1 through 64.

Adding BGP4 neighbors

Because BGP4 does not contain a peer discovery process, for each BGP4 neighbor (peer), you must indicate the IP address and the AS number of each neighbor. Neighbors that are in different autonomous systems communicate using EBGP. Neighbors within the same AS communicate using IBGP.

NOTE
If the device has multiple neighbors with similar attributes, you can simplify configuration by configuring a peer group, then adding individual neighbors to it. The configuration steps are similar, except you specify a peer group name instead of a neighbor IP address when configuring the neighbor parameters, then add individual neighbors to the peer group.

NOTE
The device attempts to establish a BGP4 session with a neighbor as soon as you enter a command specifying the IP address of the neighbor. If you want to completely configure the neighbor parameters before the device establishes a session with the neighbor, you can administratively shut down the neighbor.

To add a BGP4 neighbor with an IP address 10.157.22.26, enter the following command.

```
device(config-bgp-router)# neighbor 10.157.22.26 remote-as 100
```

The neighbor *ip-addr* must be a valid IP address.

The **neighbor** command has additional parameters, as shown in the following syntax:

Syntax: no neighbor {*ip-addr* | *peer-group-name*} {[**activate**] [**advertisement-interval** *seconds* [**allowas-in** *num*] [**bfd holdover-interval** *num*] [**bfd min-tx** *num* **min-rx** *num* **multiplier** *num*][**capability as4** [**enable** | **disable**]] [**capability orf prefixlist** [**send** | **receive**]] [**default-originate** [**route-map** *map-name*]] [**description** *string*] [**distribute-list in** | **out** *num,num,...* | *ACL-num* **localin** | **out**] [**ebgp-btsh**] [**ebgp-multihop** [*num*]] [**enforce-first-as**]] [**local-as** *as-num* [**no-prepend**]] [**maxas-limit in** [*num* | **disable**] [**maximum-prefix** *num* [*threshold*] [**teardown**] [**next-hop-self**] [**password** *string*] [**peer-group** *group-name*] [**prefix-list** *string* **in** | **out**] [**remote-as** *as-number*] [**remove-private-as**] [**route-map in** | **out** *map-name*] [**route-reflector-client**] [**send-community**] [**shutdown** [**generate-rib-out**]] [**soft-reconfiguration inbound**] [**static-network-edge**] [**timers keep-alive** *num* **hold-time** *num*] [**unsuppress-map** *map-name*] [**update-source** *ip-addr* | **ethernet** *slot / portnum* | **loopback** *num* | **ve** *num*] [**weight** *num*] [**send-label**]}

The *ip-addr* and *peer-group-name* parameters indicate whether you are configuring an individual neighbor or a peer group. If you specify a neighbor IP address, you are configuring that individual neighbor. If you specify a peer group name, you are configuring a peer group.

activate allows exchange of routes in the current family mode.

advertisement-interval *seconds* configures an interval in seconds over which the specified neighbor or peer group will hold all route updates before sending them. At the expiration of the timer, the routes are sent as a batch. The default value for this parameter is zero. Acceptable values are 0 to 600 seconds.

NOTE
The device applies the advertisement interval only under certain conditions. The device does not apply the advertisement interval when sending initial updates to a BGP4 neighbor. As a result, the device sends the updates one immediately after another, without waiting for the advertisement interval.

allowas-in *num* disables the AS_PATH check function for routes learned from a specified location. BGP4 usually rejects routes that contain an AS number within an AS_PATH attribute to prevent routing loops. In an MPLS or VPN hub and spoke topology this can prevent legitimate routes from being accepted. The **allowas-in** option stops this blockage. *num* specifies the number of occurrences of the AS number.

capability as4 [**enable** | **disable**] enables the capability of processing AS4s. The optional keywords **enable and disable** specify whether the feature should be changed from its current state. For example, if this neighbor belongs to a peer group that is enabled for AS4s but you want to disable it on the current interface, use the command and include the **disable** keyword.

capability orf prefixlist [**send** | **receive**] configures cooperative device filtering. The **send** and **receive** parameters specify the support you are enabling:

- **send** - The device sends the IP prefix lists as Outbound Route Filters (ORFs) to the neighbor.
- **receive** - The device accepts filters as Outbound Route Filters (ORFs) from the neighbor.

If you do not specify either **send** or **receive** , both capabilities are enabled. The **prefixlist** parameter specifies the type of filter you want to send to the neighbor.

NOTE
The current release supports cooperative filtering only for filters configured using IP prefix lists.

default-originate [**route-map***map-name*] configures the device to send the default route 0.0.0.0 to the neighbor. If you use the route-map *map-name* parameter, the route map injects the default route conditionally, based on the match conditions in the route map.

description *string* specifies a name for the neighbor. You can enter an alphanumeric text string up to 80 characters long.

distribute-list in | out *num,num,...* specifies a distribute list to be applied to updates to or from the specified neighbor. The **in** and **out** keywords specify whether the list is applied on updates received from the neighbor, or sent to the neighbor. The *num,num,...* parameter specifies the list of address-list filters. The device applies the filters in the order in which you list them and stops applying the filters in the distribute list when a match is found.

To use an IP ACL instead of a distribute list, you can specify **distribute-list***ACL-num***in | out** . In this case, *ACL-num* is an IP ACL.

NOTE
By default, if a route does not match any of the filters, the device denies the route. To change the default behavior, configure the last filter as **permit any any** .

NOTE
The address filter must already be configured.

ebgp-btsh enables GTSM protection for the specified neighbor.

ebgp-multihop [*num*] specifies that the neighbor is more than one hop away and that the session type with the neighbor is EBGP-multihop. This option is disabled by default. The *num* parameter specifies the TTL you are adding for the neighbor. You can specify a number from 0 through 255. The default is 0. If you leave the EBGP TTL value set to 0, the software uses the IP TTL value.

enforce-first-as ensures, for this neighbor, that the first AS listed in the AS_SEQUENCE field of an AS path update message from EBGP neighbors is the AS of the neighbor that sent the update.

weight *num* specifies a weight that the device applies to routes received from the neighbor. You can specify a number from 0 through 65535.

NOTE
By default, if an AS-path does not match any of the filters or ACLs, the device denies the route. To change the default behavior, configure the last filter or ACL as **permit any any** .

NOTE
The AS-path filter or ACL must already be configured.

local-as *as-num* assigns a local AS number with the value specified by the *as-num* variable to the neighbor being configured. The *as-num* has no default value. Its range is 1 - 4294967295.

NOTE
When the **local-as** option is used, the device automatically prepends the local AS number to the routes that are received from the EBGP peer; to disable this behavior, include the **no-prepend** keyword.

maxas-limit in *num* |**disable** specifies that the device discard routes that exceed a maximum AS path length received in UPDATE messages. You can specify a value from 0 - 300. The default value is 300. The **disable** keyword is used to stop a neighbor from inheriting the configuration from the peer-group or global and to the use system default value.

maximum-prefix *num* specifies the maximum number of IP network prefixes (routes) that can be learned from the specified neighbor or, if configured at peer group level, per neighbor in the peer group. You can specify a value from 0 through 4294967295. The default is 0 (unlimited).

- The *num* parameter specifies the maximum number. The range is 0 through 4294967295. The default is 0 (unlimited).
- The *threshold* parameter specifies the percentage of the value you specified for the **maximum-prefix** *num* , at which you want the software to generate a Syslog message. You can specify a value from 1 (one percent) to 100 (100 percent). The default is 100.
- The **teardown** parameter tears down the neighbor session if the maximum-prefix limit is exceeded. The session remains shutdown until you clear the prefixes using the **clear ip bgp neighbor all** or **clear ip bgp neighbor** command, or change the maximum prefix configuration for the neighbor. The software also generates a Syslog message.

next-hop-self specifies that the device should list itself as the next hop in updates sent to the specified neighbor. This option is disabled by default.

password *string* specifies an MD5 password for securing sessions between the device and the neighbor. You can enter a string up to 80 characters long. The string can contain any alphanumeric characters and spaces if the words in the password are placed inside quotes.

NOTE
If you want the software to assume that the value you enter is the clear-text form, and to encrypt display of that form, do not enter 0 or 1. Instead, omit the encryption option and allow the software to use the default behavior. If you specify encryption option 1, the software assumes that you are entering the encrypted form of the password or authentication string. In this case, the software decrypts the password or string you enter before using the value for authentication. If you accidentally enter option 1 followed by the clear-text version of the password or string, authentication will fail because the value used by the software will not match the value you intended to use.

The system creates an MD5 hash of the password and uses it for securing sessions between the device and its neighbors. To display the configuration, the system uses a 2-way encoding scheme to be able to retrieve the original password that was entered.

By default, the password is encrypted. If you want the password to appear in clear text, insert a 0 between the password and the string.

```
device(config-bgp)# neighbor 10.157.22.26 password 0 marmalade
```

The system adds an encryption code followed by the encrypted text of the original password. For example, the following portion of the code has the encrypted code “2”.

The system adds an encryption code followed by the encrypted text of the original password. For example, the following portion of the code has the encrypted code “2”.

```
password 2 $IUA2PWc9LW9VIW9zVQ=="
```

One of the following may be displayed:

- 0 = the password is not encrypted and is in clear text
- 1 = the password uses proprietary simple cryptographic 2-way algorithm (only for Brocade NetIron CES devices)
- 2 = the password uses proprietary base64 cryptographic 2-way algorithm (only for Brocade NetIron XMR and Brocade MLX series devices)

peer-group *group-name* assigns the neighbor to the specified peer group.

prefix-list *string* **in** |**out** specifies an IP prefix list. You can use IP prefix lists to control routes to and from the neighbor. IP prefix lists are an alternative method to AS-path filters. The **in** and **out** keywords specify whether the list is applied on updates received from the neighbor or sent to the neighbor. The filters can use the same prefix list or different prefix lists.

remote-as *as-number* specifies the AS in which the remote neighbor resides. The *as-number* has no default value. The range is 1 - 4294967295.

remove-private-as configures the device to remove private AS numbers from update messages the device sends to this neighbor. The device will remove AS numbers 64512 through 65535 (the well-known BGP4 private AS numbers) from the AS-path attribute in update messages the device sends to the neighbor. This option is disabled by default.

route-map in | **out** map-name specifies a route map the device will apply to updates sent to or received from the specified neighbor. The **in** and **out** keywords specify whether the list is applied on updates received from the neighbor or sent to the neighbor.

NOTE
The route map must already be configured.

route-reflector-client specifies that this neighbor is a route-reflector client of the device. Use the parameter only if this device is going to be a route reflector. This option is disabled by default.

send-community enables sending the community attribute in updates to the specified neighbor. By default, the device does not send the community attribute.

shutdown administratively shuts down the session with this neighbor. Shutting down the session lets you configure the neighbor and save the configuration without actually establishing a session with the neighbor.

When a peer is put into the shutdown state, ribout routes are not produced for that peer. You can elect to produce ribout routes using the **generate-rib-out** option. This option is disabled by default.

soft-reconfiguration inbound enables the soft reconfiguration feature, which stores all the route updates received from the neighbor. If you request a soft reset of inbound routes, the software performs the reset by comparing the policies against the stored route updates, instead of requesting the neighbor BGP4 route table or resetting the session with the neighbor.

static-network-edge controls the advertisement of a static BGP4 network to BGP4 neighbors that are configured as Service Edge Devices.

timers keep-alive *num* **hold-time** *num* overrides the global settings for the Keep Alive Time and Hold Time. For the Keep Alive Time, you can specify 0 - 65535 seconds. For the Hold Time, you can specify 0 or a number in the range 3 through 65535 (1 and 2 are not allowed). If you set the Hold Time to 0, the device waits indefinitely for messages from a neighbor without concluding that the neighbor is non-operational. The defaults for these parameters are the currently configured global Keep Alive Time and Hold Time.

unsuppress-map *map-name* removes route suppression from neighbor routes when those routes have been dampened due to aggregation.

update-source *ip-addr* | **ethernet***slot/portnum* | **loopback***num* | **ve***num* configures the device to communicate with the neighbor through the specified interface. There is no default.

weight *num* specifies a weight a device will add to routes received from the specified neighbor. BGP4 prefers larger weights over smaller weights. The default weight is 0.

The **send-label** keyword enables IPv6 label capability for the IPv4 peers.

Removing route dampening from suppressed routes

You can selectively un-suppress specific routes that have been suppressed due to aggregation, and allow these routes to be advertised to a specific neighbor or peer group.

```
device(config-bgp)# aggregate-address 10.1.0.0 255.255.0.0 summary-only
device(config-bgp)# show ip bgp route 10.1.0.0/16 longer
Number of BGP Routes matching display condition : 2
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix           Next Hop        Metric     LocPrf     Weight Status
1      10.1.0.0/16      0.0.0.0                    101        32768  BAL
AS_PATH:
2      10.1.44.0/24     10.2.0.1        1          101        32768  BLS
AS_PATH:
```

In this example, the **aggregate-address** command configures an aggregate address of 10.1.0.0 255.255.0.0. and the **summary-only** parameter prevents the device from advertising more specific routes contained within the aggregate route.

Entering a **show ip bgp route** command for the aggregate address 10.1.0.0/16 shows that the more specific routes aggregated into 10.1.0.0/16 have been suppressed. In this case, the route to 10.1.44.0/24 has been suppressed. If you enter this command, the display shows that the route is not being advertised to the BGP4 neighbors.

```
device(config-bgp)# show ip bgp route 10.1.44.0/24
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix           Next Hop        Metric     LocPrf     Weight Status
1      10.1.44.0/24     10.2.0.1        1          101        32768  BLS
AS_PATH:
Route is not advertised to any peers
```

To override the **summary-only** parameter and allow a specific route to be advertised to a neighbor, enter commands such as the following

```
device(config)# ip prefix-list Unsuppress1 permit 10.1.44.0/24
device(config)# route-map RouteMap1 permit 1
device(config-routemap RouteMap1)# exit
device(config)# router bgp
device(config-bgp)# neighbor 10.1.0.2 unsuppress-map RouteMap1
device(config-bgp)# clear ip bgp neighbor 10.1.0.2 soft-out
```

The **ip prefix-list** command configures an IP prefix list for network 10.1.44.0/24, which is the route you want to unsuppress. The next two commands configure a route map that uses the prefix list as input. The **neighbor** command enables the device to advertise the routes specified in the route map to neighbor 10.1.0.2. The **clear** command performs a soft reset of the session with the neighbor so that the device can advertise the unsuppressed route.

Syntax: [no] neighbor { *ip-addr* | *peer-group-name* } **unsuppress-map** *map-name*

The **show ip bgp route** command verifies that the route has been unsuppressed.

```
device(config-bgp)# show ip bgp route 10.1.44.0/24
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix           Next Hop        MED LocPrf     Weight Status
1      10.1.44.0/24     10.2.0.1        1   101        32768  BLS
AS_PATH:
Route is advertised to 1 peers:
10.1.0.2(4)
```

Encrypting BGP4 MD5 authentication keys

When you configure a BGP4 neighbor or neighbor peer group, you can specify an MD5 authentication string to authenticate packets exchanged with the neighbor or peer group of neighbors.

For added security, by default, the software encrypts the display of the authentication string. The software also provides an optional parameter to disable encryption of the authentication string, on an individual neighbor or peer group basis. By default, MD5 authentication strings are displayed in encrypted format in the output of the following commands:

- **show running-config** (or **write terminal**)
- **show configuration**
- **show ip bgp config**

When encryption of the authentication string is enabled, the string is encrypted in the CLI regardless of the access level you are using.

When you save the configuration to the startup configuration file, the file contains the new BGP4 command syntax and encrypted passwords or strings.

NOTE
Brocade recommends that you save a copy of the startup configuration file for each device you plan to upgrade.

Encryption example

The following commands configure a BGP4 neighbor and a peer group, and specify MD5 authentication strings (passwords) to authenticate packets exchanged with the neighbor or peer group.

```
device(config-bgp)# local-as 2
device(config-bgp)# neighbor xyz peer-group
device(config-bgp)# neighbor xyz password abc
device(config-bgp)# neighbor 10.10.200.102 peer-group xyz
device(config-bgp)# neighbor 10.10.200.102 password test
```

The BGP4 configuration commands appear in the following format as a result of the **show ip bgp configuration** command.

```
device# show ip bgp configuration
Current BGP configuration:
router bgp
 local-as 2
 neighbor xyz peer-group
 neighbor xyz password $b24tbw==
 neighbor 10.10.200.102 peer-group xyz
 neighbor 10.10.200.102 remote-as 1
 neighbor 10.10.200.102 password $on-o
```

In this output, the software has converted the commands that specify an authentication string into the new syntax (described below), and has encrypted display of the authentication strings.

Since the default behavior does not affect the BGP4 configuration itself but does encrypt display of the authentication string, the CLI does not list the encryption options.

Syntax: [no] neighbor { *ip-addr* | *peer-group-name* } **password** *string*

The *ip-addr* | *peer-group-name* parameter indicates whether you are configuring an individual neighbor or a peer group. If you specify the IP address of a neighbor, you are configuring that individual neighbor. If you specify a peer group name, you are configuring a peer group.

If you want the software to assume that the value you enter is the clear-text form and to encrypt the display of that form, do not enter 0 or 1. Instead, omit the encryption option and allow the software to use the default behavior. If you specify encryption option 1, the software assumes that you are entering the encrypted form of the password or authentication string. In this case, the software decrypts the password or string you enter before using the value for authentication. If you accidentally enter option 1 followed by the clear-text version of the password or string, authentication will fail because the value used by the software will not match the value you intended to use.

The **password** *string* parameter specifies an MD5 authentication string to secure sessions between the device and the neighbor. You can enter a string of up to 80 characters. The string can contain any alphanumeric characters, but must be placed inside quotes if it contains a space.

The system creates an MD5 hash of the password and uses it to secure sessions between the device and the neighbors. To display the configuration, the system uses a 2-way encoding scheme to retrieve the original password.

By default, password is encrypted. If you want the password to be in clear text, insert a 0 between **password** and *string.*

```
device(config-bgp)# neighbor 10.157.22.26 password 0 admin
```

The system adds an encryption code followed by the encrypted text of the original password. For example, the following portion of the code has the encrypted code “2”.

```
password 2 $IUA2PWc9LW9VIW9zVQ=="
```

```
device(config-bgp)# neighbor 10.157.22.26 password 0 marmalade
```

Displaying the authentication string

To display the authentication string, enter the following commands.

```
device(config)# enable password-display
device(config)# show ip bgp neighbors
```

The **enable password-display** command enables display of the authentication string, but only in the output of the **show ip bgp neighbors** command. String display is still encrypted in the startup configuration file and running configuration. Enter the command at the global CONFIG level of the CLI.

NOTE
The command also displays SNMP community strings in clear text, in the output of the **show snmp server** command.

Advertising IPv4 routes between IPv6 BGP peers

This feature transports IPv6 routes over an IPv4 BGP session. If you have an existing IPv4 BGP mesh, you can use it to transport IPv6 routes instead of creating a new IPv6 BGP mesh.

First, configure peering using the IPv4 addresses under IPv6 address family, i.e enabling the IPv6 address family for the IPv4 neighbor. Since the advertised next hop is usually unreachable, set the next hop with a static route or with an inbound or outbound route-map.

For example, IPv6 Router 1 (IP address 192.168.1.1) and IPv6 Router 2 (IP address 192.168.12) are connected through an IPv4 network. To configure the IPv4 peers to advertise the IPv6 routes, enter the following commands.

On Device 1, enter the following:

```
device(config-bgp)# show ip bgp config
router bgp
local-as 1
neighbor 192.168.1.2 remote-as 2

device(config-bgp)# address-family ipv6 unicast
device(config-bgp)# neighbor 192.168.1.2 activate
device(config-bgp)# neighbor 192.168.1.2 route-map in t5
exit-address-family

address-family vpnv4 unicast
exit-address-family

 device(config-bgp)# show route-map
route-map t5 permit 1
set ipv6 next-hop 2001:db8::2
```

On Device 2, enter the following:

```
device# show ip bgp config
router bgp
local-as 1
neighbor 192.168.1.1 remote-as 1

address-family ipv6 unicast
redistribute static
neighbor 192.168.1.1 activate
exit-address-family
!
```

Displaying neighbor information

To display IPv6 unicast route summary information, enter the **show ip bgp ipv6 summary** command:

```
device(config-bgp)# show ip bgp ipv6 summary
BGP4 Summary
Router ID: 10.1.1.1 Local AS Number: 1
Confederation Identifier: not configured
Confederation Peers:
Maximum Number of IP ECMP Paths Supported for Load Sharing: 1
Number of Neighbors Configured: 1, UP: 1
Number of Routes Installed: 1, Uses 86 bytes
Number of Routes Advertising to All Neighbors: 0 (0 entries)
Number of Attribute Entries Installed: 1, Uses 90 bytes
Neighbor Address AS# State Time Rt:Accepted Filtered Sent ToSend
192.168.1.2 2 ESTAB 0h 1m51s 1 0 0 0
```

Syntax: show ip bgp ipv6 summary

To display IPv6 unicast device information with respect to the IPv4 neighbor, enter the **show ip bgp ipv6 neighbors** command:

```
device(config-bgp)# show ip bgp ipv6 neighbors
Total number of BGP Neighbors: 1
1 IP Address: 192.168.1.2, AS: 2 (EBGP), RouterID: 10.1.1.2, VRF: default-vrf
State: ESTABLISHED, Time: 0h8m33s, KeepAliveTime: 60, HoldTime: 180
KeepAliveTimer Expire in 17 seconds, HoldTimer Expire in 135 seconds
UpdateSource: Loopback 1
RefreshCapability: Received
.......
Neighbor NLRI Negotiation:
Peer Negotiated IPV6 unicast capability
Peer configured for IPV6 unicast Routes
Neighbor ipv6 MPLS Label Capability Negotiation:
Neighbor AS4 Capability Negotiation:
TCP Connection state: ESTABLISHED, flags:00000033 (0,0)
```

Syntax: show ip bgp ipv6 neighbors [**last-packet-with-error**] [**routes-summary**] [*ip-address*]

The **neighbors** parameter provides details on TCP and BGP neighbor connections. The **last-packet-with-error** parameter displays the last packet received with error. The **routes-summary** parameter displays the routes summary.

The *ip-address* parameter is the neighbor IP address. The following sub-parameters are available for the *ip-address* parameter:

[advertised routes} [flap-statistics] [last-packet-with-error] [received] [received-routes] [rib-out-routes] [routes][routes-summary]

The **advertised-routes** parameter displays routes advertised to a neighbor. The **flap-statistics** parameter displays flap statistics for a neighbor. The **last-packet-with-error** parameter displays the last packet received with error. The **received** parameter displays the received ORF from neighbor. The **received-routes** parameter displays the received routes from neighbor. The **rib-out-routes** parameter

displays RIB-out routes for a neighbor.The **routes** parameter displays routes learned from neighbor. The **routes-summary** parameter displays routes summary for a neighbor.

To display IPv6 multicast route information with respect to IPv4 neighbors, enter the show ip mbgp ipv6 neighbors command.

Syntax: show ip mbgp ipv6 neighbors [**summary**] [**last-packet-with-error**] [**routes-summary**] [*ip-address*]

The **summary** parameter displays a summary of BGP neighbor status. The **last-packet-with-error** parameter displays the last packet received with error. The **routes-summary** parameter displays the routes summary for a neighbor.

The *ip-address* parameter is the neighbor IP address. Use the following sub-parameters to display details on TCP and BGP neighbor connections.

The **advertised-routes** parameter displays routes advertised to a neighbor. The **flap-statistics** parameter displays flap statistics for a neighbor. The **last-packet-with-error** parameter displays the last packet received with error. The **received** parameter displays the received ORF from neighbor. The **received-routes** parameter displays the received routes from neighbor. The **rib-out-routes** parameter displays RIB-out routes for a neighbor.The **routes** parameter displays routes learned from neighbor. The **routes-summary** parameter displays routes summary for a neighbor.

Clearing IPv6 route information

To clear IPv6 unicast route information with respect to IPv4 neighbors, enter the **clear ip bgp ipv6 neighbor** command.

Syntax: clear ip bgp ipv6 [**6pe**] [**dampening**] [**flap-statistics**] [**l2vpn**] [**local**] [**routes**] [**traffic**] [**ipv6**] [**vpnv4**] [**vrf**] [**neighbor**] [*as-number* | *ipaddress* | *peer-group-name* | **all**]

The **6pe** parameter clears information for 6pe address family.

The **dampening** parameter clears route flap dampening information. The **flap-statistics** parameter clears route flap statistics.

The **l2vpn** parameter clears information for l2vpn address family.

The **local** parameter clears local information. The **routes** parameter clears BGP routes. The **traffic** parameter clears BGP traffic counters. The **ipv6** parameter clears information for ipv6 address family. The **vpnv4** parameter clears information for VPNV4 address family. The **vrf** parameter clears information for a VRF instance.

The **neighbor** parameter has the following sub-parameters:

as-number identifies neighbors with the specified AS number, 1-4294967295. *ipaddress* identifies the neighbor IP address. *peer-group-name* clears the peer group name identified using ASCII string. *all* clears all BGP neighbors.

To clear IPv6 multicast route information with respect to IPv4 neighbor, enter the **clear ip mbgp ipv6 neighbor** command.

Syntax: clear ip mbgp ipv6 [neighbor] [*as-number* | *ipaddress* | *peer-group-name* | **all**]

as-number identifies neighbors with the specified AS number, 1-4294967295. *ipaddress* identifies the neighbor IP address. *peer-group-name* clears the peer group name identified using ASCII string. *all* clears all BGP neighbors.

Adding a BGP4 peer group

A peer group is a set of BGP4 neighbors that share common parameters. The benefits of peer groups are:

- Simplified neighbor configuration - You can configure a set of neighbor parameters and then apply them to multiple neighbors. You do not need to configure the common parameters individually on each neighbor.
- Flash memory conservation - Using peer groups instead of individually configuring all the parameters for each neighbor requires fewer configuration commands in the startup configuration file.

You can perform the following tasks on a peer-group basis:

- Reset neighbor sessions
- Perform soft-outbound resets (the device updates outgoing route information to neighbors but does not entirely reset the sessions with those neighbors)
- Clear BGP4 message statistics
- Clear error buffers

Peer group parameters

You can set all neighbor parameters in a peer group. When you add a neighbor to the peer group, the neighbor receives all the parameter settings you set in the group, except parameter values you have explicitly configured for the neighbor. If you do not set a neighbor parameter in the peer group and the parameter also is not set for the individual neighbor, the neighbor uses the default value.

Peer group configuration rules

The following rules apply to peer group configuration:

- You must configure a peer group before you can add neighbors to the peer group.
- If you remove a parameter from a peer group, the value for that parameter is reset to the default for all the neighbors within the peer group, unless you have explicitly set that parameter on individual neighbors. In this case, the value you set on the individual neighbors applies to those neighbors, while the default value applies to neighbors for which you have not explicitly set the value.

NOTE
If you enter a command to remove the remote AS parameter from a peer group, the software makes sure that the peer group does not contain any neighbors. If the peer group contains neighbors, the software does not allow you to remove the remote AS so that the neighbors in the peer group that are using the remote AS do not lose connectivity to the device.

You can override neighbor parameters that do not affect outbound policy on an individual neighbor basis:

- If you do not specify a parameter for an individual neighbor, the neighbor uses the value in the peer group.
- If you set the parameter for the individual neighbor, that value overrides the value you set in the peer group.
- If you add a parameter to a peer group that already contains neighbors, the parameter value is applied to neighbors that do not already have the parameter explicitly set. If a neighbor has the parameter explicitly set, the explicitly set value overrides the value you set for the peer group.
- If you remove the setting for a parameter from a peer group, the value for that parameter changes to the default value for all the neighbors in the peer group that do not have that parameter individually set.

Configuring a peer group

To configure a peer group, enter commands such as the following at the BGP4 configuration level.

```
device(config-bgp)# neighbor PeerGroup1 peer-group
device(config-bgp)# neighbor PeerGroup1 description "EastCoast Neighbors"
device(config-bgp)# neighbor PeerGroup1 remote-as 100
device(config-bgp)# neighbor PeerGroup1 distribute-list out 1
device(config-bgp)# neighbor PeerGroup1 capability as4 enable|disable
```

The commands in this example configure a peer group called "PeerGroup1" and set the following parameters for the peer group:

- A description, "EastCoast Neighbors"
- A remote AS number, 100
- A distribute list for outbound traffic
- The capability of PeerGroup1 to utilize a four-byte AS number

The software applies these parameters to each neighbor you add to the peer group. You can override the description parameter for individual neighbors. If you set the description parameter for an individual neighbor, the description overrides the description configured for the peer group.

Syntax: neighbor *peer-group-name* **peer-group**

The *peer-group-name* parameter specifies the name of the group and can be up to 80 characters long. The name can contain special characters and internal blanks. If you use internal blanks, you must use quotation marks around the name. For example, the command **neighbor "My Three Peers"** peer-group is valid, but the command **neighbor My Three Peers** peer-group is not valid.

Syntax: [no] neighbor *ip-addr* | *peer-group-name* [**advertisement-interval** *num*] [**default-originate** [**route-map** *map-name*]] [**description** *string*] [**distribute-list** { **in** | **out** } *num,num...* | *ACL-num* **in** | **out**] [**ebgp-multihop** [*num*]] [**filter-list in** | **out** *num,num,...* | *acl-num* | **out** | **weight**] [**maxas-limit in** [*num* | **disable**] [**maximum-prefix** *num* [*threshold*] [**teardown**]] [**next-hop-self**] [**password** *string*] [**prefix-list** *string* **in** | **out**] **remote-as** *as-number*] [**remove-private-as**] [**route-map-in** | **out** *map-name*] [**route-reflector-client**] [**send-community**] [**soft-reconfiguration inbound**] [**shutdown**] [**timers keep-alive** *num* **hold-time** *num*] [**update-source loopback** *num* **ethernet** *slotl portnum* | **loopback** *num* | **ve** *num*] [**weight** *num*] [**local-as** *as-num*]

The *ip-addr* and *peer-group-name* parameters indicate whether you are configuring a peer group or an individual neighbor. You can specify a peer group name or IP address with the **neighbor** command. If you specify a peer group name, you are configuring a peer group. If you specify a neighbor IP address, you are configuring that individual neighbor. Use the *ip-addr* parameter if you are configuring an individual neighbor instead of a peer group.

The remaining parameters are the same ones supported for individual neighbors.

Applying a peer group to a neighbor

After you configure a peer group, you can add neighbors to the group. When you add a neighbor to a peer group, you are applying all the neighbor attributes specified in the peer group to the neighbor.

To add neighbors to a peer group, enter commands such as the following.

```
device(config-bgp-router)# neighbor 192.168.1.12 peer-group PeerGroup1
device(config-bgp-router)# neighbor 192.168.2.45 peer-group PeerGroup1
device(config-bgp-router)# neighbor 192.168.3.69 peer-group PeerGroup1
```

The commands in this example add three neighbors to the peer group "PeerGroup1". As members of the peer group, the neighbors automatically receive the neighbor parameter values configured for the peer group. You also can override the parameters on an individual neighbor basis. For neighbor parameters not specified for the peer group, the neighbors use the default values.

Syntax: [no] neighbor *ip-addr* **peer-group** *peer-group-name*

The *ip-addr* parameter specifies the IP address of the neighbor.

The *peer-group-name* parameter specifies the peer group name.

NOTE
You must add the peer group before you can add neighbors to it.

Administratively shutting down a session with a BGP4 neighbor

You can prevent the device from starting a BGP4 session with a neighbor by administratively shutting down the neighbor. This option is very useful for situations in which you want to configure parameters for a neighbor, but are not ready to use the neighbor. You can shut the neighbor down as soon as you have added it to the device, configure the neighbor parameters, then allow the device to reestablish a session with the neighbor by removing the shutdown option from the neighbor.

When you apply the option to shut down a neighbor, the option takes place immediately and remains in effect until you remove it. If you save the configuration to the startup configuration file, the shutdown option remains in effect even after a software reload.

The software also contains an option to end the session with a BGP4 neighbor and clear the routes learned from the neighbor. Unlike this clear option, the option for shutting down the neighbor can be saved in the startup configuration file and can prevent the device from establishing a BGP4 session with the neighbor even after reloading the software.

NOTE
If you notice that a particular BGP4 neighbor never establishes a session with the device, check the running configuration and startup configuration files for that device to see whether the configuration contains a command that is shutting down the neighbor. The neighbor may have been shut down previously by an administrator.

To shut down a BGP4 neighbor, enter commands such as the following.

```
device(config)# router bgp
device(config-bgp-router)# neighbor 10.157.22.26 shutdown
device(config-bgp-router)# write memory
```

Syntax: [no] neighbor *ip-addr* **shutdown** [**generate-rib-out**]

The *ip-addr* parameter specifies the IP address of the neighbor.

Optional BGP4 configuration tasks

The following sections describe how to perform optional BGP4 configuration tasks.

Changing the Keep Alive Time and Hold Time

The Keep Alive Time specifies how frequently the device will send KEEPALIVE messages to its BGP4 neighbors. The Hold Time specifies how long the device will wait for a KEEPALIVE or UPDATE message from a neighbor before concluding that the neighbor is dead. When the device concludes that

a BGP4 neighbor is dead, the device ends the BGP4 session and closes the TCP connection to the neighbor.

The default Keep Alive time is 60 seconds. The default Hold Time is 180 seconds.

NOTE
Generally, you should set the Hold Time to three times the value of the Keep Alive Time.

NOTE
You can override the global Keep Alive Time and Hold Time on individual neighbors.

To change the Keep Alive Time to 30 and Hold Time to 90, enter the following command.

```
device(config-bgp-router)# timers keep-alive 30 hold-time 90
```

Syntax: [no] timers keep-alive *num* **hold-time** *num*

For each keyword, *num* indicates the number of seconds. The Keep Alive Time can be 0 - 65535. The Hold Time can be 0 or 3 - 65535 (1 and 2 are not allowed). If you set the Hold Time to 0, the device waits indefinitely for messages from a neighbor without concluding that the neighbor is dead.

Changing the BGP4 next-hop update timer

By default, the device updates the BGP4 next-hop tables and affected BGP4 routes five seconds after IGP route changes. You can change the update timer to a value from 1 through 30 seconds.

To change the BGP4 update timer value to 15 seconds, for example, enter the **update-time** command at the BGP configuration level of the CLI.

```
device(config-bgp-router)# update-time 15
```

Syntax: [no] update-time *secs*

The *secs* parameter specifies the number of seconds and can be from 0 through 30. The default is 5. The value of 0 permits fast BGP4 convergence for situations such as link-failure or IGP route changes. Setting the value to 0 starts the BGP4 route calculation in sub-second time. All other values from 1 through 30 are still calculated in seconds.

Enabling fast external fallover

BGP4 devices rely on KEEPALIVE and UPDATE messages from neighbors to signify that the neighbors are alive. For BGP4 neighbors that are two or more hops away, such messages are the only indication that the BGP4 protocol has concerning the alive state of the neighbors. As a result, if a neighbor becomes non-operational, the device waits until the Hold Time expires or the TCP connection fails before concluding that the neighbor is not operational and closing its BGP4 session and TCP connection with the neighbor.

The device waits for the Hold Time to expire before ending the connection to a directly-attached BGP4 neighbor that becomes non-operational.

For directly-attached neighbors, the device immediately senses loss of a connection to the neighbor from a change of state of the port or interface that connects the device to the neighbor. For directly-attached EBGP neighbors, the device uses this information to immediately close the BGP4 session and TCP connection to locally attached neighbors that become non-operational.

NOTE
The fast external failover feature applies only to directly attached EBGP neighbors. The feature does not apply to IBGP neighbors.

To enable fast external fallover, enter the following command.

```
device(config-bgp-router)# fast-external-fallover
```

To disable fast external fallover again, enter the following command.

```
device(config-bgp-router)# no fast-external-fallover
```

Syntax: [no] fast-external-fallover

Changing the maximum number of paths for BGP4 Multipath load sharing

Multipath load sharing enables the device to balance traffic to a route across multiple equal-cost paths of the same route type (EBGP or IBGP).

To configure the device to perform BGP4 Multipath load sharing:

- Enable IP load sharing if it is disabled.
- Set the maximum number of BGP4 load sharing paths. The default maximum number is 1, which means no BGP4 load sharing takes place by default.

NOTE
The maximum number of BGP4 load sharing paths cannot be greater than the maximum number of IP load sharing paths.

How Multipath load sharing affects route selection

During evaluation of multiple paths to select the best path to a given destination (for installment in the IP route table), the device performs a final comparison of the internal paths. The following events occur when load sharing is enabled or disabled:

- When load sharing is disabled, the device prefers the path with the lower device ID if the **compare-routerid** command is enabled.
- When load sharing and BGP4 Multipath load sharing are enabled, the device balances the traffic across multiple paths instead of choosing just one path based on device ID.

Refer to How BGP4 selects a path for a route (BGP best path selection algorithm) on page 29 for a description of the BGP4 algorithm.

When you enable IP load sharing, the device can load-balance BGP4 or OSPF routes across up to four equal paths by default. You can change the number load sharing paths to a value from 2 through 32.

Changing the maximum number of shared BGP4 paths

To change the maximum number of BGP4 shared paths, enter commands such as the following.

```
device(config)# router bgp
```

```
device(config-bgp-router)# maximum-paths 4
device(config-bgp-router)# write memory
```

Syntax: [no] maximum-paths *num* | **use-load-sharing**

The *number* parameter specifies the maximum number of paths across which the device can balance traffic to a given BGP4 destination. The *number* value range is 2 through 32 and the default is 1.

NOTE

The maximum number of paths supported by the BR-MLX-10Gx24-DM module is 16.

When the **use-load-sharing** option is used in place of the *number* variable, the maximum IP ECMP path value is determined solely by the value configured using the**ip load-sharing** command.

Customizing BGP4 Multipath load sharing

By default, when BGP4 Multipath load sharing is enabled, both IBGP and EBGP paths are eligible for load sharing, while paths from different neighboring autonomous systems are not eligible. You can change load sharing to apply only to IBGP or EBGP paths, or to support load sharing among paths from different neighboring autonomous systems.

To enable load sharing of IBGP paths only, enter the following command at the BGP4 configuration level of the CLI.

```
device(config-bgp-router)# multipath ibgp
```

To enable load sharing of EBGP paths only, enter the following command at the BGP4 configuration level of the CLI.

```
device(config-bgp-router)# multipath ebgp
```

To enable load sharing of paths from different neighboring autonomous systems, enter the following command at the BGP4 configuration level of the CLI.

```
device(config-bgp)# multipath multi-as
```

Syntax: [no] multipath ebgp | **ibgp** | **multi-as**

The **ebgp, bgp, and multi-as** parameters specify the change you are making to load sharing:

- **ebgp** - Multipath load sharing applies only to EBGP paths. Multipath load sharing is disabled for IBGP paths.
- **ibgp** - Multipath load sharing applies only to IBGP paths. Multipath load sharing is disabled for EBGP paths.
- **multi-as** - Multipath load sharing is enabled for paths from different autonomous systems.

By default, load sharing applies to EBGP and IBGP paths, and does not apply to paths from different neighboring autonomous systems.

Enhancements to BGP4 Multipath load sharing

Enhancements to BGP4 Multipath load sharing allows support for load sharing of BGP4 routes in IP ECMP even if the BGP4 Multipath load sharing feature is not enabled through the **use-load-sharing** option to the **maximum-paths** command. Using the following commands, you can also set separate values for IBGP and EBGP multipath load sharing.

To set the number of equal-cost multipath IBGP routes or paths that will be selected, enter commands such as the following.

```
device(config)# router bgp
device(config-bgp)# maximum-paths ibgp
```

Syntax: [no] maximum-paths ibgp *number*

The *number* variable specifies the number of equal-cost multipath IBGP routes that will be selected. The range is 2 to 32. If the value is set to 1, BGP4 level equal-cost multipath is disabled for IBGP routes.

To set the number of equal-cost multipath EBGP routes or paths that will be selected, enter commands such as the following.

```
device(config)# router bgp
device(config-bgp)# maximum-paths ebgp
```

Syntax: [no] maximum-paths ebgp *num*

The *number* variable specifies the number of equal-cost multipath EBGP routes that will be selected. The range is 2 to 32. If the value is set to 1, BGP4 level equal-cost multipath is disabled for EBGP routes.

Specifying a list of networks to advertise

By default, the device sends BGP4 routes only for the networks you either identify with the **network** command or are redistributed into BGP4 from OSPF, IS-IS, RIP, or connected routes.

NOTE
The exact route must exist in the IP route table before the device can create a local BGP4 route.

To configure the device to advertise network 10.157.22.0/24, enter the following command.

```
device(config-bgp-router)# network 10.157.22.0 255.255.255.0
```

Syntax: [no] network *ip-addr ip-mask* [**route-map** *map-name*] | [**weight** *num*] | [**backdoor**]

The *ip-addr* is the network number and the *ip-mask* specifies the network mask.

The **route-map** *map-name* parameter specifies the name of the route map you want to use to set or change BGP4 attributes for the network you are advertising. The route map must already be configured. If it is not, the default action is to deny redistribution.

The **weight** *num* parameter specifies a weight to be added to routes to this network.

The **backdoor** parameter changes the administrative distance of the route to this network from the EBGP administrative distance (20 by default) to the Local BGP4 weight (200 by default), tagging the route as a backdoor route. Use this parameter when you want the device to prefer IGP routes such as RIP or OSPF routes over the EBGP route for the network.

Specifying a route map when configuring BGP4 network advertising

You can specify a route map when you configure a BGP4 network to be advertised. The device uses the route map to set or change BGP4 attributes when creating a local BGP4 route.

NOTE
You must configure the route map *before* you can specify the route map name in a BGP4 network configuration; otherwise, the route is not imported into BGP4.

To configure a route map, and use it to set or change route attributes for a network you define for BGP4 to advertise, enter commands such as the following.

```
device(config)# route-map set_net permit 1
device(config-routemap set_net)# set community no-export
device(config-routemap set_net)# exit
device(config)# router bgp
device(config-bgp)# network 10.100.1.0/24 route-map set_net
```

The first two commands in this example create a route map named "set_net" that sets the community attribute for routes that use the route map to "NO_EXPORT". The next two commands change the CLI to the BGP4 configuration level. The last command configures a network for advertising from BGP4, and associates the "set_net" route map with the network. When BGP4 originates the 10.100.1.0/24 network, BGP4 also sets the community attribute for the network to "NO_EXPORT".

Changing the default local preference

When the device uses the BGP4 algorithm to select a route to send to the IP route table, one of the parameters the algorithm uses is the local preference. Local preference indicates a degree of preference for a route relative to other routes. BGP4 neighbors can send the local preference value as an attribute of a route in an UPDATE message.

Local preference applies only to routes within the local AS. BGP4 devices can exchange local preference information with neighbors who also are in the local AS, but BGP4 devices do not exchange local preference information with neighbors in remote autonomous systems.

The default local preference is 100. For routes learned from EBGP neighbors, the default local preference is assigned to learned routes. For routes learned from IBGP neighbors, the local preference value is not changed for the route.

When the BGP4 algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen.

NOTE
To set the local preference for individual routes, use route maps.

To change the default local preference to 200, enter the following command.

```
device(config-bgp)# default-local-preference 200
```

Syntax: [no] default-local-preference *num*

The *num* parameter indicates the preference and can be a value from 0 - 4294967295.

Using the IP default route as a valid next-hop for a BGP4 route

By default, the device does not use a default route to resolve a BGP4 next-hop route. If the IP route lookup for the BGP4 next-hop does not result in a valid IGP route (including static or direct routes), the BGP4 next-hop is considered to be unreachable and the BGP4 route is not used.

In some cases, such as when the device is acting as an edge device, you can allow the device to use the default route as a valid next-hop. To do so, enter the following command at the BGP4 configuration level of the CLI.

```
device(config-bgp)# next-hop-enable-default
```

Syntax: [no] next-hop-enable-default

Changing the default MED (Metric) used for route redistribution

The Brocade device can redistribute directly connected routes, static IP routes, RIP routes, IS-IS routes, and OSPF routes into BGP4. By default, BGP4 uses zero (0) for direct connected routes and the metric (MED) value of IGP routes in the IP route table. The MED is a global parameter that specifies the cost that will be applied to all routes, if assigned, when they are redistributed into BGP4. When routes are selected, lower metric values are preferred over higher metric values. The default, the BGP4 MED value is not assigned.

NOTE
RIP, IS-IS, and OSPF also have default metric parameters. The parameters are set independently for each protocol and have different ranges.

To change the default metric to 40, enter the following command.

```
device(config-bgp-router)# default-metric 40
```

Syntax: default-metric *num*

The *num* indicates the metric and can be a value from 0 through 4294967295.

Enabling next-hop recursion

For each BGP4 route learned, the device performs a route lookup to obtain the IP address of the next-hop for the route. A BGP4 route is eligible for addition in the IP route table only if the following conditions are true:

- The lookup succeeds in obtaining a valid next-hop IP address for the route.
- The path to the next-hop IP address is an IGP path or a static route path.

By default, the software performs only one lookup for the next-hop IP address for the BGP4 route. If the next-hop lookup does not result in a valid next-hop IP address, or the path to the next-hop IP address is a BGP4 path, the software considers the BGP4 route destination to be unreachable. The route is not eligible to be added to the IP route table.

The BGP4 route table can contain a route with a next-hop IP address that is not reachable through an IGP route, even though the device can reach a hop farther away through an IGP route. This can occur when the IGPs do not learn a complete set of IGP routes, so the device learns about an internal route through IBGP instead of through an IGP. In this case, the IP route table will not contain a route that can be used to reach the BGP4 route destination.

To enable the device to find the IGP route to the next-hop gateway for a BGP4 route, enable recursive next-hop lookups. With this feature enabled, if the first lookup for a BGP4 route results in an IBGP path that originated within the same AS, rather than an IGP path or static route path, the device performs a lookup on the next-hop IP address for the next-hop gateway. If this second lookup results in an IGP path, the software considers the BGP4 route to be valid and adds it to the IP route table. Otherwise, the device performs another lookup on the next-hop IP address of the next-hop for the next-hop gateway, and so on, until one of the lookups results in an IGP route.

NOTE
You must configure a static route or use an IGP to learn the route to the EBGP multihop peer.

Enabling recursive next-hop lookups

The recursive next-hop lookups feature is disabled by default. To enable recursive next-hop lookups, enter the following command at the BGP4 configuration level of the CLI.

```
device(config-bgp-router)# next-hop-recursion
```

Syntax: [no] next-hop-recursion

Example when recursive route lookups are disabled

The output here shows the results of an unsuccessful next-hop lookup for a BGP4 route. In this case, next-hop recursive lookups are disabled. This example is for the BGP4 route to network 10.0.0.0/24.

```
device# show ip bgp route
Total number of BGP Routes: 5
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix             Next Hop       MED     LocPrf    Weight Status
1      0.0.0.0/0          10.1.0.2       0       100       0      BI
         AS_PATH: 65001 4355 701 80
2      10.10.0.0/24       10.0.0.1       1       100       0      BI
         AS_PATH: 65001 4355 1
3      10.40.0.0/24       10.1.0.2       0       100       0      BI
         AS_PATH: 65001 4355 701 1 189
4      10.0.0.0/24        10.0.0.1       1       100       0      I
         AS_PATH: 65001 4355 3356 7170 1455
5      10.25.0.0/24       10.157.24.1    1       100       0      I
         AS_PATH: 65001 4355 701
```

In this example, the device cannot reach 10.0.0.0/24, because the next-hop IP address for the route is an IBGP route instead of an IGP route, and is considered unreachable by the device. The IP route table entry for the next-hop gateway for the BGP4 route's next-hop gateway (10.0.0.1/24) is shown here.

```
device# show ip route 10.0.0.1
Total number of IP routes: 37
      Network Address     Gateway     Port  Cost  Type
      10.0.0.0                10.0.0.1    1/1   1     B
```

Since the route to the next-hop gateway is a BGP4 route, and not an IGP route, it cannot be used to reach 10.0.0.0/24. In this case, the device tries to use the default route, if present, to reach the subnet that contains the BGP4 route next-hop gateway.

```
device# show ip route 10.0.0.0/24
Total number of IP routes: 37
      Network Address     Gateway      Port  Cost  Type
      0.0.0.0             10.0.0.202   1/1   1     S
```

Example when recursive route lookups are enabled

When recursive next-hop lookups are enabled, the device continues to look up the next-hop gateways along the route until the device finds an IGP route to the BGP4 route destination.

```
device# show ip bgp route
Total number of BGP Routes: 5
```

```
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix          Next Hop        MED      LocPrf    Weight Status
1      0.0.0.0/0       10.1.0.2        0        100       0      BI
         AS_PATH: 65001 4355 701 80
2      10.10.0.0/24    10.0.0.1        1        100       0      BI
         AS_PATH: 65001 4355 1
3      10.40.0.0/24    10.1.0.2        0        100       0      BI
         AS_PATH: 65001 4355 701 1 189
4      10.0.0.0/24   10.0.0.1          1        100       0      BI
         AS_PATH: 65001 4355 3356 7170 1455
5      10.25.0.0/24   10.157.24.1      1        100       0       I
         AS_PATH: 65001 4355 701
```

The first lookup results in an IBGP route, to network 10.0.0.0/24.

```
device# show ip route 10.0.0.1
Total number of IP routes: 38
      Network Address       Gateway        Port   Cost   Type
      10.0.0.0              10.0.0.1       1/1    1      B
         AS_PATH: 65001 4355 1
```

Since the route to 10.0.0.1/24 is not an IGP route, the device cannot reach the next hop through IP, and so cannot use the BGP4 route. In this case, since recursive next-hop lookups are enabled, the device next performs a lookup for the next-hop gateway to 10.0.0.1's next-hop gateway, 10.0.0.1.

```
device# show ip bgp route 10.0.0.0
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix          Next Hop     MED     LocPrf     Weight Status
1      10.0.0.0/24   10.0.0.1       1       100        0      BI
         AS_PATH: 65001 4355 1
```

The next-hop IP address for 10.0.0.1 is not an IGP route, which means the BGP4 route destination still cannot be reached through IP. The recursive next-hop lookup feature performs a lookup on the next-hop gateway for 10.0.0.1

```
device# show ip route 10.0.0.1
Total number of IP routes: 38
Network Address         Gateway        Port     Cost   Type
      10.0.0.0          0.0.0.0        1/1      1      D

        AS_PATH: 65001 4355 1 1
```

This lookup results in an IGP route that is a directly-connected route. As a result, the BGP4 route destination is now reachable through IGP, which means the BGP4 route can be added to the IP route table. The IP route table with the BGP4 route is shown here.

```
device# show ip route 10.0.0.0/24
Total number of IP routes: 38
Network Address       Gateway         Port     Cost   Type
      10.0.0.0         10.0.0.1       1/1      1      B
         AS_PATH: 65001 4355 1
```

The device can use this route because it has an IP route to the next-hop gateway. Without recursive next-hop lookups, this route would not be in the IP route table.

Changing administrative distances

BGP4 devices can learn about networks from various protocols, including the EBGP portion of BGP4, and IGPs such as OSPF, IS-IS, and RIP, the routes to a network may differ depending on the protocol from which the routes were learned.

To select one route over another based on the source of the route information, the device can use the administrative distances assigned to the sources. The administrative distance is a protocol-independent metric that IP devices use to compare routes from different sources.

The device re-advertises a learned best BGP4 route to neighbors even when the route table manager does not also select that route for installation in the IP route table. The best BGP4 route is the BGP4 path that BGP4 selects based on comparison of the paths' BGP4 route parameters.

When selecting a route from among different sources (BGP4, OSPF, RIP, IS-IS, static routes, and so on), the software compares the routes on the basis of the administrative distance for each route. If the administrative distance of the paths is lower than the administrative distance of paths from other sources (such as static IP routes, RIP, or OSPF), the BGP4 paths are installed in the IP route table.

The default administrative distances on the device are:

- Directly connected - 0 (this value is not configurable)
- Static - 1 is the default and applies to all static routes, including default routes. This can be assigned a different value.
- EBGP - 20
- OSPF - 110
- IS-IS - 115
- RIP - 120
- IBGP - 200
- Local BGP4 - 200
- Unknown - 255 (the device will not use this route)

Lower administrative distances are preferred over higher distances. For example, if the device receives routes for the same network from OSPF and from RIP, the device will prefer the OSPF route by default. The administrative distances are configured in different places in the software. The device re-advertises a learned best BGP4 route to neighbors by default, regardless of whether the administrative distance for the route is lower than the administrative distances of other routes from different route sources to the same destination:

- To change the EBGP, IBGP, and Local BGP4 default administrative distances, refer to the instructions in this section.
- To change the default administrative distance for OSPF, RIP, IS-IS, refer to Configuring a static BGP4 network on page 117.

To change the default administrative distances for EBGP, IBGP, and Local BGP4, enter a command such as the following.

```
device(config-bgp-router)# distance 200 200 200
```

Syntax: [no] distance *external-distance internal-distance local-distance*

The *external-distance* sets the EBGP distance and can be a value from 1 through 255.

The *internal-distance* sets the IBGP distance and can be a value from 1 through 255.

The *local-distance* sets the Local BGP4 distance and can be a value from 1 through 255.

Requiring the first AS to be the neighbor AS

By default, the Brocade device does not require the first AS listed in the AS_SEQUENCE field of an AS path update message from EBGP neighbors to be the AS of the neighbor that sent the update. However, you can enable the Brocade device to have this requirement. You can enable this requirement globally for the device, or for a specific neighbor or peer group. This section describes how to enable this requirement.

When you configure the device to require that the AS an EBGP neighbor is in be the same as the first AS in the AS_SEQUENCE field of an update from the neighbor, the device accepts the update only if the AS numbers match. If the AS numbers do not match, the Brocade device sends a notification

message to the neighbor and closes the session. The requirement applies to all updates received from EBGP neighbors.

The hierarchy for enforcement of this feature is: a neighbor will try to use the enforce-first-as value if one is configured; if none is configured, the neighbor will try to use the configured value for a peer group. If neither configuration exists, enforcement is simply that of the global configuration (which is disabled by default).

To enable this feature globally, enter the **enforce-first-as** command at the BGP4 configuration level of the CLI.

```
device(config-bgp)# enforce-first-as
```

Syntax: [no] enforce-first-as

To enable this feature for a specific neighbor, enter the following command at the BGP4 configuration level.

```
device(config-bgp)# neighbor 10.1.1.1 enforce-first-as enable
```

Syntax: [no] neighbor *ip-address* **enforce-first-as** [**enable** | **disable**]

The ip-address value is the IP address of the neighbor.

When the first-as requirement is enabled, its status appears in the output of the **show running configuration** command. The optional last keyword choice of **enable** or **disable** lets you specify whether the output of the **show running configuration** command includes the configuration of the first-as requirement. This option allows the **show running configuration** command output to show what is actually configured.

To enable this feature for a peer group, enter the following command at the BGP4 configuration level.

```
device(config-bgp)# neighbor Peergroup1 enforce-first-as enable
```

Syntax: [no] neighbor *peer-group-name* **enforce-first-as** [**enable** | **disable**]

The *peer-group-name* value is the name of the peer group.

When the first-as requirement is enabled, its status appears in the output of the show running configuration command. The optional last keyword choice, that of **enable** or **disable** , lets you specify whether the output of the show running configuration command includes the configuration of the first-as requirement: this option helps the show running command output to show what you have actually configured.

The following example shows a running configuration with the first-as enforcement items (for global, peer group, and neighbor) in bold.

```
device(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
device(config-bgp)# local-as 1

device(config-bgp)# enforce-first-as
device(config-bgp)# neighbor abc peer-group
device(config-bgp)# neighbor abc remote-as 2
device(config-bgp)# neighbor abc enforce-first-as disable
device(config-bgp)# neighbor 192.168.1.2 peer-group abc
device(config-bgp)# neighbor 192.168.1.2 enforce-first-as enable
```

Disabling or re-enabling comparison of the AS-Path length

AS-Path comparison is Step 5 in the algorithm that BGP4 uses to select the next path for a route. Comparison of the AS-Path length is enabled by default. To disable it, enter the following command at the BGP4 configuration level of the CLI.

```
device(config-bgp)# as-path-ignore
```

Syntax: [no] as-path-ignore

This command disables comparison of the AS-Path lengths of otherwise equal paths. When you disable AS-Path length comparison, the BGP4 algorithm shown in How BGP4 selects a path for a route (BGP best path selection algorithm) on page 29 skips from Step 4 to Step 6.

Enabling or disabling comparison of device IDs

Device ID comparison is Step 11 in the algorithm BGP4 uses to select the next path for a route.

NOTE
Comparison of device IDs is applicable only when BGP4 load sharing is disabled.

When device ID comparison is enabled, the path comparison algorithm compares the device IDs of the neighbors that sent the otherwise equal paths:

- If BGP4 load sharing is disabled (maximum-paths 1), the instructions in this section selects the path that came from the neighbor with the lower device ID.
- If BGP4 load sharing is enabled, the device load shares among the remaining paths. In this case, the device ID is not used to select a path.

NOTE
Device ID comparison is disabled by default.

To enable device ID comparison, enter the **compare-routerid** command at the BGP4 configuration level of the CLI.

```
device(config-bgp-router)# compare-routerid
```

Syntax: [no] compare-routerid

Configuring the device to always compare Multi-Exit Discriminators

A Multi-Exit Discriminator (MED) is a value that the BGP4 algorithm uses when it compares multiple paths received from different BGP4 neighbors in the same AS for the same route. In BGP4, a MED for a route is equivalent to its metric.

BGP4 compares the MEDs of two otherwise equivalent paths if and only if the routes were learned from the same neighboring AS. This behavior is called deterministic MED. Deterministic MED is always enabled and cannot be disabled.

You can enable the device to always compare the MEDs, regardless of the AS information in the paths. For example, if the device receives UPDATES for the same route from neighbors in three autonomous systems, the device can compare the MEDs of all the paths together instead of comparing the MEDs for the paths in each autonomous system individually.

To enable this comparison, enter the **always-compare-med** command at the BGP4 configuration level of the CLI. This option is disabled by default.

NOTE
By default, value 0 (most favorable) is used in MED comparison when the MED attribute is not present. The default MED comparison results in the device favoring route paths that do not have their MEDs. Use the **med-missing-as-worst** command to force the device to regard a BGP4 route with a missing MED attribute as the least favorable route, when comparing the MEDs of the routes.

NOTE
MED comparison is not performed for internal routes originated within the local AS or confederation unless the **compare-med-empty-aspath** command is configured.

To configure the device to always compare MEDs, enter the following command.

```
device(config-bgp-router)# always-compare-med
```

Syntax: [no] always-compare-med

The following BGP4 command directs BGP4 to take the MED value into consideration even if the route has an empty as-path path attribute.

```
device(config)# router bgp
device(config-bgp-router)# compare-med-empty-aspath
```

Syntax: [no] compare-med-empty-aspath

Treating missing MEDs as the worst MEDs

By default, the device favors a lower MED over a higher MED during MED comparison. Since the device assigns the value 0 to a route path MED if the MED value is missing, the default MED comparison results in the device favoring the route paths that are missing their MEDs.

To change this behavior so that the device favors a route that has a MED over a route that is missing its MED, enter the following command at the BGP4 configuration level of the CLI.

```
device(config-bgp-router)# med-missing-as-worst
```

Syntax: [no] med-missing-as-worst

NOTE
This command affects route selection only when route paths are selected based on MED comparison. It is still possible for a route path that is missing its MED to be selected based on other criteria. For example, a route path with no MED can be selected if its weight is larger than the weights of the other route paths.

Configuring route reflection parameters

Normally, all the BGP4 devices within an AS are fully meshed. Since each device has an IBGP session with each of the other BGP4 devices in the AS, each IBGP device has a route for each IBGP neighbor. For large autonomous systems containing many IBGP devices, the IBGP route information in each fully-meshed IBGP device may introduce too much administrative overhead.

To avoid this overhead, you can organize your IGP devices into clusters:

- A cluster is a group of IGP devices organized into route reflectors and route reflector clients. You configure the cluster by assigning a cluster ID on the route reflector and identifying the IGP neighbors that are members of that cluster. All configuration for route reflection takes place on the route reflectors. Clients are unaware that they are members of a route reflection cluster. All members of the cluster must be in the same AS. The cluster ID can be any number from 1 - 4294967295, or an IP address. The default is the device ID expressed as a 32-bit number.

NOTE
If the cluster contains more than one route reflector, you need to configure the same cluster ID on all the route reflectors in the cluster. The cluster ID helps route reflectors avoid loops within the cluster.

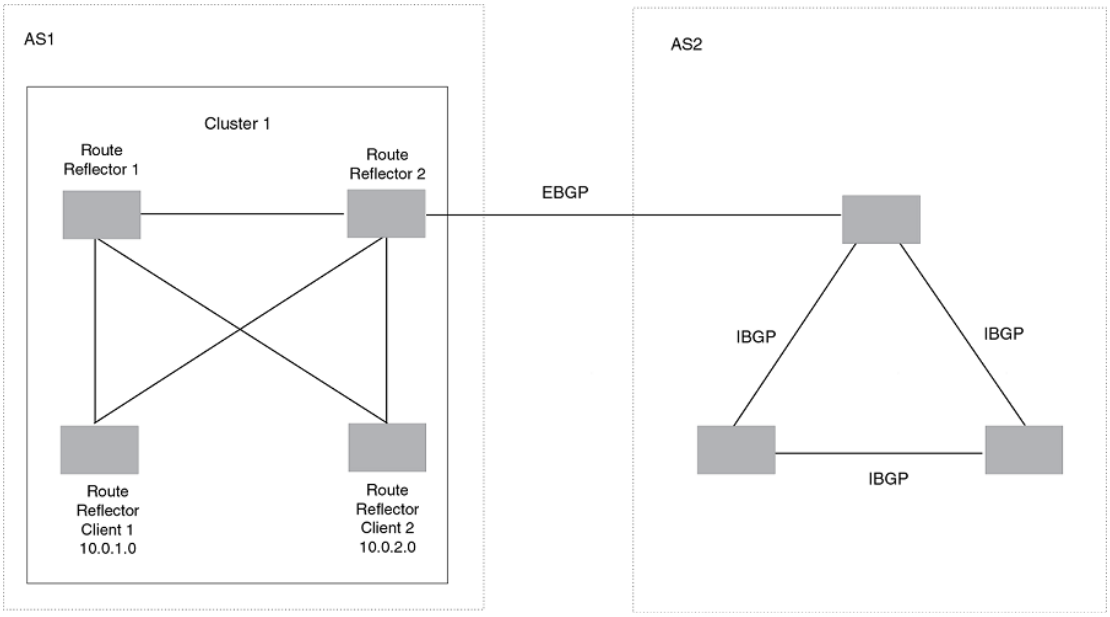
- A route reflector is an IGP device configured to send BGP4 route information to all the clients (other BGP4 devices) within the cluster. Route reflection is enabled on all BGP4 devices by default but does not take effect unless you add route reflector clients to the device.
- A route reflector client is an IGP device identified as a member of a cluster. You identify a device as a route reflector client on the device that is the route reflector, not on the client. The client itself requires no additional configuration. In fact, the client does not know that it is a route reflector client. The client just knows that it receives updates from its neighbors and does not know whether one or more of those neighbors are route reflectors.

NOTE
Route reflection applies only among IBGP devices within the same AS. You cannot configure a cluster that spans multiple autonomous systems.

This is an example of a route reflector configuration. In this example, two devices are configured as route reflectors for the same cluster, which provides redundancy in case one of the reflectors becomes unavailable. Without redundancy, if a route reflector becomes unavailable, the clients for that device are cut off from BGP4 updates.

AS1 contains a cluster with two route reflectors and two clients. The route reflectors are fully meshed with other BGP4 devices, but the clients are not fully meshed and rely on the route reflectors to propagate BGP4 route updates.

FIGURE 5 A route reflector configuration



Support for RFC 4456

Route reflection on Brocade devices is based on RFC 4456. This updated RFC helps eliminate routing loops that are possible in some implementations of the older specification, RFC 1966. These instances include:

- The device adds the route reflection attributes only if it is a route reflector, and only when advertising IBGP route information to other IBGP neighbors. The attributes are not used when communicating with EBGP neighbors.
- A device configured as a route reflector sets the ORIGINATOR_ID attribute to the device ID of the device that originated the route. The route reflector sets this attribute only if this is the first time the route is being reflected (sent by a route reflector).
- If a device receives a route with an ORIGINATOR_ID attribute value that is the same as the ID of the device, the device discards the route and does not advertise it. By discarding the route, the device prevents a routing loop.
- The first time a route is reflected by a device configured as a route reflector, the route reflector adds the CLUSTER_LIST attribute to the route. Other route reflectors that receive the route from an IBGP neighbor add their cluster IDs to the front of the routes CLUSTER_LIST. If the route reflector does not have a cluster ID configured, the device adds its device ID to the front of the CLUSTER_LIST.
- If a device configured as a route reflector receives a route with a CLUSTER_LIST that contains the cluster ID of the route reflector, the route reflector discards the route.

Configuration procedures for BGP4 route reflector

To configure a Brocade device to be a BGP4 route reflector, use either of the following methods.

NOTE
All configuration for route reflection takes place on the route reflectors, not on the clients.

Enter the following commands to configure a Brocade device as route reflector 1. To configure route reflector 2, enter the same commands on the device that will be route reflector 2. The clients require no configuration for route reflection.

```
device(config-bgp)# cluster-id 1
```

Syntax: [no] cluster-id *num* | *ip-addr*

The *num* and *ip-addr* parameters specify the cluster ID and can be a number from 1 - 4294967295, or an IP address. The default is the device ID. You can configure one cluster ID on the device. All route-reflector clients for the device are members of the cluster.

NOTE
If the cluster contains more than one route reflector, you need to configure the same cluster ID on all the route reflectors in the cluster. The cluster ID helps route reflectors avoid loops within the cluster.

To add an IBGP neighbor to the cluster, enter the following command:

```
device(config-bgp)# neighbor 10.0.1.0 route-reflector-client
```

Syntax: [no] neighbor *ip-addr* **route-reflector-client**

Disabling or re-enabling client-to-client route reflection

By default, the clients of a route reflector are not required to be fully meshed. Routes from a client are reflected to other clients. However, if the clients are fully meshed, route reflection is not required between clients.

If you need to disable route reflection between clients, enter the **no client-to-client-reflection** command. When this feature is disabled, route reflection does not occur between clients does still occur between clients and non-clients.

```
device(config-bgp)# no client-to-client-reflection
```

Enter the following command to re-enable the feature.

```
device(config-bgp)# client-to-client-reflection
```

Syntax: [no] client-to-client-reflection

Configuring confederations

A **confederation** is a BGP4 Autonomous System (AS) that has been subdivided into multiple, smaller autonomous systems. Subdividing an AS into smaller autonomous systems simplifies administration and reduces BGP4-related traffic, which in turn reduces the complexity of the Interior Border Gateway Protocol (IBGP) mesh among the BGP4 devices in the AS.

The Brocade implementation of this feature is based on RFC 3065.

Normally, all BGP4 devices within an AS must be fully meshed, so that each BGP4 device has BGP4 sessions to all the other BGP4 devices within the AS. This is feasible in smaller autonomous systems, but becomes unmanageable in autonomous systems containing many BGP4 devices.

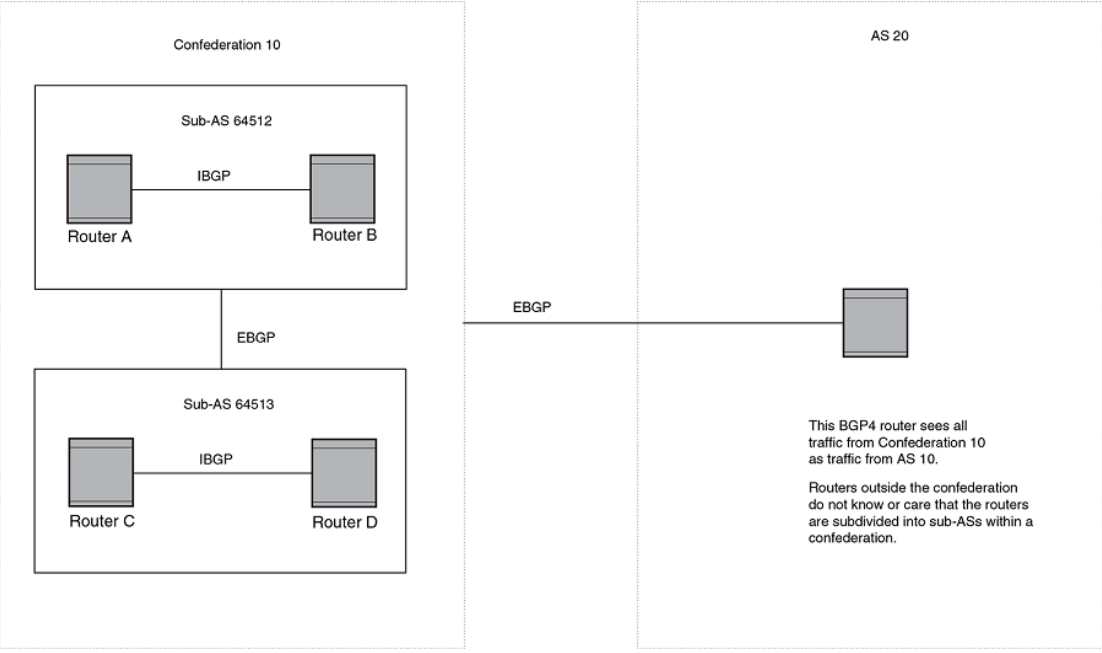
When you configure BGP4 devices into a confederation, all the devices within a sub-AS (a subdivision of the AS) use IBGP and must be fully meshed. However, devices use EBGP to communicate between different sub-autonomous systems.

NOTE
Another way to reduce the complexity of an IBGP mesh is to use route reflection. However, if you want to run different Interior Gateway Protocols (IGPs) within an AS, you must configure a confederation. You can run a separate IGP within each sub-AS.

To configure a confederation, configure groups of BGP4 devices into sub-autonomous systems. A sub-AS is simply an AS. The term "sub-AS" distinguishes autonomous systems within a confederation from autonomous systems that are not in a confederation. For the viewpoint of remote autonomous systems, the confederation ID is the AS ID. Remote autonomous systems do not know that the AS represents multiple sub-autonomous systems with unique AS IDs.

NOTE
You can use any valid AS numbers for the sub-autonomous systems. If your AS is connected to the Internet, Brocade recommends that you use numbers from within the private AS range (64512 through 65535). These are private autonomous system numbers and BGP4 devices do not propagate these AS numbers to the Internet.

FIGURE 6 Example BGP4 confederation



In this example, four devices are configured into two sub-autonomous systems, each containing two of the devices. The sub-autonomous systems are members of confederation 10. Devices within a sub-AS must be fully meshed and communicate using IBGP. In this example, devices A and B use IBGP to communicate. devices C and D also use IBGP. However, the sub-autonomous systems communicate with one another using EBGP. For example, device A communicates with device C using EBGP. The devices in the confederation communicate with other autonomous systems using EBGP.

Devices in other autonomous systems are unaware that devices A through D are configured in a confederation. In fact, when devices in confederation 10 send traffic to devices in other autonomous systems, the confederation ID is the same as the AS number for the devices in the confederation. Thus, devices in other autonomous systems see traffic as coming from AS 10 and are unaware that the devices in AS 10 are subdivided into sub-autonomous systems within a confederation.

Configuring a BGP4 confederation

To configure a BGP4 configuration, perform these configuration tasks on each BGP4 device within the confederation:

- Configure the local AS number. The local AS number indicates membership in a sub-AS. All BGP4 devices with the same local AS number are members of the same sub-AS. BGP4 devices use the local AS number when communicating with other BGP4 devices in the confederation.
- Configure the confederation ID. The confederation ID is the AS number by which BGP4 devices outside the confederation recognize the confederation. A BGP4 device outside the confederation is not aware of, and does not care that BGP4 devices are in multiple sub-autonomous systems. A BGP4 device uses the confederation ID to communicate with devices outside the confederation. The confederation ID must differ from the sub-AS numbers.
- Configure the list of the sub-AS numbers that are members of the confederation. All devices within the same sub-AS use IBGP to exchange device information. Devices in different sub-autonomous systems within the confederation use EBGP to exchange device information.

To configure four devices to be members of confederation 10 (consisting of sub-autonomous systems 64512 and 64513), enter commands such as the following.

Commands for device A

```
deviceA(config)# router bgp
deviceA(config-bgp)# local-as 64512
deviceA(config-bgp)# confederation identifier 10
deviceA(config-bgp)# confederation peers 64512 64513
deviceA(config-bgp)# write memory
```

Syntax: [no] local-as *num*

The *num* parameter with the **local-as** command indicates the AS number for the BGP4 devices within the sub-AS. You can specify a number in the range 1 - 4294967295. I Brocade recommends that you use a number within the range of well-known private autonomous systems, 64512 through 65535.

Syntax: [no] confederation identifier *num*

The num parameter with the **confederation identifier** command indicates the confederation number. The confederation ID is the AS number by which BGP4 devices outside the confederation recognize the confederation. A BGP4 device outside the confederation is not aware of, and does not care that your BGP4 devices are in multiple sub-autonomous systems. BGP4 devices use the confederation ID when communicating with devices outside the confederation. The confederation ID must be different from the sub-AS numbers. For the *num* parameter, you can specify a number in the range 1 - 4294967295.

Syntax: [no] confederation peers *num* [*num* **...**]

The *num* parameter with the **confederation peers** command indicates the sub-AS numbers for the sub-autonomous systems in the confederation. You can list all sub-autonomous systems in the confederation. You must specify all the sub-autonomous systems with which this device has peer sessions in the confederation. All the devices within the same sub-AS use IBGP to exchange device information. Devices in different sub-autonomous systems within the confederation use EBGP to exchange device information. The *num* is a number in the range 1 - 4294967295.

Commands for device B

```
deviceB(config)# router bgp
deviceB(config-bgp)# local-as 64512
deviceB(config-bgp)# confederation identifier 10
deviceB(config-bgp)# confederation peers 64512 64513
deviceB(config-bgp)# write memory
```

Commands for device C

```
deviceC(config)# router bgp
deviceC(config-bgp)# local-as 64513
deviceC(config-bgp)# confederation identifier 10
deviceC(config-bgp)# confederation peers 64512 64513
deviceC(config-bgp)# write memory
```

Commands for device D

```
deviceD(config)# router bgp
deviceD(config-bgp)# local-as 64513
deviceD(config-bgp)# confederation identifier 10
deviceD(config-bgp)# confederation peers 64512 64513
deviceD(config-bgp)# write memory
```

Aggregating routes advertised to BGP4 neighbors

By default, the device advertises individual routes for all networks. The aggregation feature allows you to configure the device to aggregate routes from a range of networks into a single network prefix. For example, without aggregation, the device will individually advertise routes for networks 10.95.1.0/24, 10.95.2.0/24, and 10.95.3.0/24. You can configure the device to end a single, aggregate route for the networks instead. The aggregate route can be advertised as 10.95.0.0/16.

To aggregate routes for 10.157.22.0/24, 10.157.23.0/24, and 10.157.24.0/24, enter the following command.

```
device(config-bgp)# aggregate-address 10.157.0.0 255.255.0.0
```

Syntax: [no] aggregate-address *ip-addr ip-mask* [**as-set**] [**summary-only**] [**suppress-map** *map-name*] [**advertise-map** *map-name*] [**attribute-map** *map-name*]

The *ip-addr* and *ip-mask* parameters specify the aggregate value for the networks. Specify 0 for the host portion and for the network portion that differs among the networks in the aggregate. For example, to aggregate 10.0.1.0/24, 10.0.2.0/24, and 10.0.3.0/24, enter the IP address 10.0.0.0 and the network mask 255.255.0.0.

The **as-set** parameter causes the device to aggregate AS-path information for all the routes in the aggregate address into a single AS-path.

The **summary-only** parameter prevents the device from advertising more specific routes contained within the aggregate route.

The **suppress-map** *map-name* parameter prevents the more specific routes contained in the specified route map from being advertised.

The **advertise-map** *map-name* parameter configures the device to advertise the more specific routes in the specified route map.

The **attribute-map** *map-name* parameter configures the device to set attributes for the aggregate routes based on the specified route map.

NOTE
For the **suppress-map** , **advertise-map** , and **attribute-map** parameters, the route map must already be defined.

Configuring BGP4 restart

BGP4 restart can be configured for a global routing instance or for a specified Virtual Routing and Forwarding (VRF) instance. The following sections describe how to enable the BGP4 restart feature.

Configuring BGP4 Restart for the global routing instance

Use the following command to enable the BGP4 Restart feature globally on a device.

```
device(config)# router bgp
device(config-bgp-router)# graceful-restart
```

Syntax: [no] graceful-restart

Configuring BGP4 Restart for a VRF

Use the following command to enable the BGP4 Restart feature for a specified VRF.

```
device(config)# router bgp
device(config-bgp-router)# address-family ipv4 unicast vrf blue
device(config-bgp-ipv4u-vrf)# graceful-restart
```

Syntax: [no] graceful-restart

Configuring timers for BGP4 Restart (optional)

You can optionally configure the following timers to change their values from the default values:

- Restart Timer
- Stale Routes Timer
- Purge Timer

The *seconds* variable sets the maximum restart wait time advertised to neighbors. Possible values are 1- 3600 seconds. The default value is 120 seconds.

Configuring the restart timer for BGP4 Restart

Use the following command to specify the maximum amount of time a device will maintain routes from and forward traffic to a restarting device.

```
device(config-bgp)# graceful-restart restart-time 150
```

Syntax: [no] graceful-restart restart-time *seconds*

The *seconds* variable sets the maximum restart wait time advertised to neighbors. Possible values are 1 through 3600 seconds. The default value is 120 seconds.

Configuring BGP4 Restart stale routes timer

Use the following command to specify the maximum amount of time a helper device will wait for an end-of-RIB message from a peer before deleting routes from that peer.

```
device(config-bgp)# graceful-restart stale-routes-time 120
```

Syntax: [no] graceful-restart stale-routes-time *seconds*

The *seconds* variable sets the maximum time before a helper device cleans up stale routes. Possible values are 1 through 3600 seconds. The default value is 360 seconds.

Configuring BGP4 Restart purge timer

Use the following command to specify the maximum amount of time a device will maintain stale routes in its routing table before purging them.

```
device(config-bgp)# graceful-restart purge-time 900
```

Syntax: [no] graceful-restart purge-time *seconds*

The *seconds* variable sets the maximum time before a restarting device cleans up stale routes. Possible values are 1 - 3600 seconds. The default value is 600 seconds.

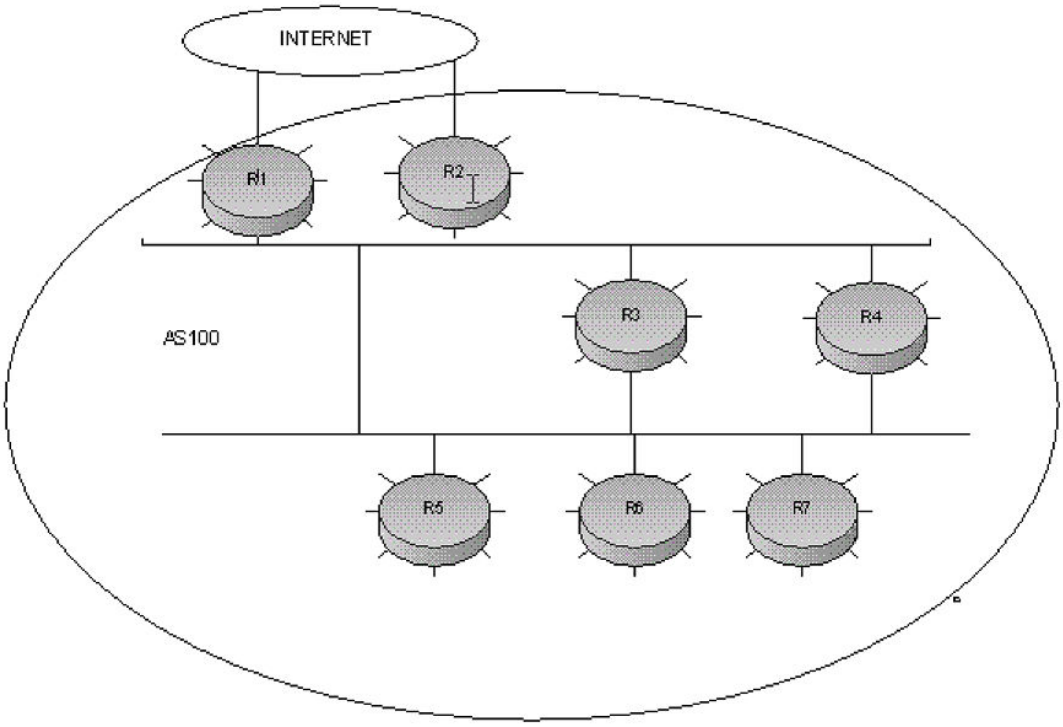
BGP4 null0 routing

BGP4 considers the null0 route in the routing table (for example, static route) as a valid route, and can use the null0 route to resolve the next hop. If the next hop for BGP4 resolves into a null0 route, the BGP4 route is also installed as a null0 route in the routing table.

The null0 routing feature allows network administrators to block certain network prefixes using null0 routes and route-maps, directing a remote device to drop all traffic for a network prefix by redistributing a null0 route into BGP4.

This example shows a topology for a null0 routing application example.

FIGURE 7 SAMPLE null0 routing application



Configuring BGP4 null0 routing

The following example configures a null0 routing application to stop denial of service attacks from remote hosts on the Internet.

1. Select a device, for example, device 6, to distribute null0 routes throughout the BGP4 network.
2. To configure a route-map perform the following step.
   - Configure a route-map to match a particular tag (50) and set the next-hop address to an unused network address (192.168.0.1).
3. Set the local-preference to a value higher than any possible internal or external local-preference (50).
4. Complete the route map by setting origin to IGP.
5. On device 6, redistribute the static routes into BGP4, using route-map *route-map-name* (redistribute static route-map block user).
6. To configure a route-map perform the following step.
   - On device 1, (the device facing the Internet), configure a null0 route matching the next-hop address in the route-map (ip route 192.168.0.1/32 null0).
7. Repeat step 3 for all devices interfacing with the Internet (edge corporate devices). In this case, device 2 has the same null0 route as device 1.
8. On device 6, configure the network prefixes associated with the traffic you want to drop. The static route IP address references a destination address. You must point the static route to the egress port, (for example, Ethernet 3/7), and specify the tag 50, matching the route-map configuration.

Configuration examples

Device 6

The following configuration defines specific prefixes to filter:

```
device(config)# ip route 10.0.0.40/29 ethernet 3/7 tag 50
device(config)# ip route 10.0.0.192/27 ethernet 3/7 tag 50
device(config)# ip route 10.014.0/23 ethernet 3/7 tag 50
```

The following configuration redistributes routes into BGP4.

```
device(config)# router bgp
device(config-bgp-router)# local-as 100
device(config-bgp-router)# neighbor router1_int_ip address remote-as 100
device(config-bgp-router)# neighbor router2_int_ip address remote-as 100
device(config-bgp-router)# neighbor router3_int_ip address remote-as 100
device(config-bgp-router)# neighbor router4_int_ip address remote-as 100
device(config-bgp-router)# neighbor router5_int_ip address remote-as 100
device(config-bgp-router)# neighbor router7_int_ip address remote-as 100
device(config-bgp-router)# redistribute static route-map blockuser
device(config-bgp-router)# exit
```

The following configuration defines the specific next hop address and sets the local preference to preferred.

```
device(config)# route-map blockuser permit 10
device(config-routemap blockuser)# match tag 50
device(config-routemap blockuser)# set ip next-hop 192.168.0.1
device(config-routemap blockuser)# set local-preference 1000000
device(config-routemap blockuser)# set origin igp
device(config-routemap blockuser)# exit
```

NOTE
A match tag can take up to 16 tags. During the execution of a route-map, a match on any tag value in the list is considered a successful match.

Device 1

The following configuration defines the null0 route to the specific next hop address. The next hop address 192.168.0.1 points to 10.178.1.101, which gets blocked.

```
device(config)# ip route 192.168.0.1/32 null0
device(config)# router bgp
device(config-bgp-router)# local-as 100
device(config-bgp-router)# neighbor router2_int_ip address remote-as 100
device(config-bgp-router)# neighbor router3_int_ip address remote-as 100
device(config-bgp-router)# neighbor router4_int_ip address remote-as 100
device(config-bgp-router)# neighbor router5_int_ip address remote-as 100
device(config-bgp-router)# neighbor router6_int_ip address remote-as 100
device(config-bgp-router)# neighbor router7_int_ip address remote-as 100
```

Device 2

The following configuration defines a null0 route to the specific next hop address. The next hop address 192.168.0.1 points to 10.178.1.101, which gets blocked.

```
device(config)# ip route 192.168.0.1/32 null0
device(config)# router bgp
device(config-bgp-router)# local-as 100
device(config-bgp-router)# neighbor router1_int_ip address remote-as 100
device(config-bgp-router)# neighbor router3_int_ip address remote-as 100
device(config-bgp-router)# neighbor router4_int_ip address remote-as 100
device(config-bgp-router)# neighbor router5_int_ip address remote-as 100
device(config-bgp-router)# neighbor router6_int_ip address remote-as 100
device(config-bgp-router)# neighbor router7_int_ip address remote-as 100
```

Show commands for BGP4 null 0 routing

After configuring the null0 application, you can display the output using **show** commands.

Device 6

Show ip route static output for device 6.

```
device# show ip route static
Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
          Destination          Gateway         Port        Cost       Type
1       10.0.0.40/29           DIRECT          eth 3/7     1/1        S
2       10.0.0.192/27          DIRECT          eth 3/7     1/1        S
3       10.0.14.0/23           DIRECT          eth 3/7     1/1        S
device#
```

Device 1 and 2

Show ip route static output for device 1 and device 2.

```
device# show ip route static
 Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
            Destination        Gateway         Port        Cost       Type
1           192.168.0.1/32     DIRECT          drop        1/1        S
device#
```

Device 6

The following is the **show ip bgp route** output for Device-6

```
device# show ip bgp route
Total number of BGP Routes: 126
```

```
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED s:STALE
       Prefix           Next Hop       MED     LocPrf     Weight Status
1      10.0.1.0/24      10.4.1.3       0        100        0      BI
         AS_PATH:
.             ..                            .      .           .      .
9      10.0.0.16/30     10.9.1.3              100        0      I
         AS_PATH: 85
10     10.0.0.40/29     192.168.0.1    1     1000000    32768     BL
         AS_PATH:
11     10.0.0.80/28     10.9.1.3              100        0      I
 .          ..                          .            .          .      .
 .          ..                                       .          .      .
36     10.0.0.96/28     10.3.1.3              100        0      I
         AS_PATH: 50
37     10.0.0.192/27    192.168.0.1       10000000      32768     BL
         AS_PATH:
 .             ..                         .           .          .      .
64     10.0.7.0/24      10.7.1.3              100        0      I
         AS_PATH: 10
65     10.0.14.0/23     192.168.0.1/        1000000 32768  BL
         AS_PATH: ..                    .           .           .     .
```

Device 1 and 2

The **show ip route** output for device 1 and device 2 shows "drop" under the Port column for the network prefixes you configured with null0 routing

```
device# show ip route
Brocade#show ip route
Total number of IP routes: 133
 Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
        Destination       Gateway       Port        Cost      Type
1       10.9.1.24/32      DIRECT        loopback 1  0/0       D
2       10.30.1.0/24      DIRECT        eth 2/7     0/0       D
3       10.40.1.0/24      DIRECT        eth 2/1     0/0       D
.
13      10.110.0.6/31     10.90.1.3     eth 2/2     20/1      B
14      10.110.0.16/30    10.90.1.3     eth 2/2     20/1      B
15      10.110.0.40/29    DIRECT        drop        200/0     B
.          ..               .              .          .       .
42      10.115.0.192/27   DIRECT        drop        200/0     B
43      10.115.1.128/26   10.30.1.3     eth 2/7     20/1      B
.          ..               .              .          .       .
69      10.120.7.0/24     10.70.1.3     eth 2/10    20/1      B
70      10.120.14.0/23    DIRECT        drop        200/0     B
.          ..               .              .          .       .
.          ..               .              .          .       .
131     10.144.0.0/12     10.80.1.3      eth 3/4     20/1      B
132     12.168.0.1/32     DIRECT         drop        1/1       S
```

Modifying redistribution parameters

By default, the route information between BGP4 and the IP IGPs (RIP, IS-IS, and OSPF) is not redistributed. You can configure the device to redistribute OSPF, IS-IS, or RIP routes, directly connected routes, or static routes into BGP4.

To enable redistribution of all OSPF routes and directly attached routes into BGP4, enter the following commands.

```
device(config)# router bgp
device(config-bgp)# redistribute ospf
device(config-bgp)# redistribute connected
device(config-bgp)# write memory
```

Syntax: [no] redistribute connected | **ospf** | **rip** | **isis** | **static**

The **connected** parameter indicates that you are redistributing routes to directly attached devices into BGP4.

The **ospf** parameter indicates that you are redistributing OSPF routes into BGP4.

NOTE
Entering **redistribute ospf** simply redistributes internal OSPF routes. To redistribute external OSPF routes also, use the **redistribute ospf match external** command.

The **rip** parameter indicates that you are redistributing RIP routes into BGP4.

The **isis** parameter indicates that you are redistributing IS-IS routes into BGP4.

The **static** parameter indicates that you are redistributing static routes into BGP4.

Redistributing connected routes

To configure BGP4 to redistribute directly connected routes, enter the following command.

```
device(config-bgp)# redistribute connected
```

Syntax: [no] redistribute connected [**metric** *num*] [**route-map** *map-name*]

The **connected** parameter indicates that you are redistributing routes to directly attached devices into BGP4.

The **metric** *num* parameter changes the metric. You can specify a value from 0 through 4294967295. The default is not assigned.

The **route-map** *map-name* parameter specifies a route map to be consulted before adding the RIP route to the BGP4 route table.

NOTE
The route map you specify must already be configured on the device.

Redistributing RIP routes

To configure BGP4 to redistribute RIP routes and add a metric of 10 to the redistributed routes, enter the following command.

```
device(config-bgp)# redistribute rip metric 10
```

Syntax: [no] redistribute rip [**metric** *num*] [**route-map** *map-name*]

The **rip** parameter indicates that you are redistributing RIP routes into BGP4.

The **metric** *num* parameter changes the metric. You can specify a value from 0 - 4294967295. The default is not assigned.

The **route-map** *map-name* parameter specifies a route map to be consulted before adding the RIP route to the BGP4 route table.

NOTE
The route map you specify must already be configured on the device.

Redistributing OSPF external routes

To configure the device to redistribute OSPF external type 1 routes, enter the following command.

```
device(config-bgp)# redistribute ospf match external1
```

Syntax: [no] redistribute ospf [**match internal** | **external1** | **external2**] [**metric** *num*] [**route-map** *map-name*]

The **ospf** parameter indicates that you are redistributing OSPF routes into BGP4.

The match **internal, external1, and external2** parameters apply only to OSPF. These parameters specify the types of OSPF routes to be redistributed into BGP4. The default is internal.

NOTE
If you do not enter a value for the **match** parameter, (for example, you enter r**edistribute ospf** only) then only internal OSPF routes will be redistributed.

The **metric** *num* parameter changes the metric. You can specify a value from 0 through 4294967295. The default is not assigned.

The **route-map** *map-name* parameter specifies a route map to be consulted before adding the OSPF route to the BGP4 route table.

NOTE
The route map you specify must already be configured on the device.

NOTE
If you use both the **redistribute ospf route-map** command and the **redistribute ospf match internal** command, the software uses only the route map for filtering.

Redistributing static routes

To configure the device to redistribute static routes, enter the following command.

```
device(config-bgp)# redistribute static
```

Syntax: [no] redistribute static [**metric** *num*] [**route-map** *map-name*]

The **static** parameter indicates that you are redistributing static routes into BGP4.

The **metric***num* parameter changes the metric. You can specify a value from 0 - 4294967295. The default is 0.

The **route-map** *map-name* parameter specifies a route map to be consulted before adding the static route to the BGP4 route table.

NOTE
The route map you specify must already be configured on the device.

Redistributing IBGP routes

By default, the device does not allow redistribute IBGP routes from BGP4 into RIP, OSPF, or IS-IS. This behavior helps eliminate routing loops. In non-default VRF instances, by default, the device does allow redistribution IBGP routes from BGP4 into RIP, OSPF.

To enable the device to redistribute BGP4 routes into OSPF, RIP, or IS-IS enter the following command.

```
device(config-bgp)# bgp-redistribute-internal
```

Syntax: [no] bgp-redistribute-internal

To disable redistribution of IBGP routes into RIP, IS-IS, and OSPF, enter the **bgp-redistribute-internal** command.

Filtering

This section describes the following:

- AS-path filtering
- Route-map continue clauses for BGP4 routes
- Defining and applying IP prefix lists
- Defining neighbor distribute lists
- Defining route maps
- Router-map continue clauses for BGP4 routes
- Configuring cooperative BGP4 route filtering

AS-path filtering

You can filter updates received from BGP4 neighbors based on the contents of the AS-path list accompanying the updates. For example, to deny routes that have the AS 10.3.2.1 in the AS-path from entering the BGP4 route table, you can define a filter.

The device provides the following methods for filtering on AS-path information:

- AS-path filters
- AS-path ACLs

NOTE
The device cannot support AS-path filters and AS-path ACLs at the same time. Use one method or the other, but do not mix methods.

NOTE
Once you define a filter or ACL, the default action for updates that do not match a filter is **deny** . To change the default action to **permit** , configure the last filter or ACL as **permit any any** .

AS-path filters or AS-path ACLs can be referred to by the filter list number of a BGP4 neighbor as well as by match clauses in a route map.

Defining an AS-path ACL

To configure an AS-path list that uses "acl 1", enter a command such as the following.

```
device(config)# ip as-path access-list acl1 permit 100
device(config)# router bgp
device(config-bgp)# neighbor 10.10.10.1 filter-list acl1 in
```

Syntax: [no] ip as-path access-list *string* [**seq s** *eq-value*] **deny** | **permit** *regular-expression*

The **ip as-path** command configures an AS-path ACL that permits routes containing AS number 100 in their AS paths. The **neighbor** command then applies the AS-path ACL to advertisements and updates received from neighbor 10.10.10.1. In this example, the only routes the device permits from neighbor 10.10.10.1 are those whose AS-paths contain AS-path number 100.

The *string* parameter specifies the ACL name. (If you enter a number, the CLI interprets the number as a text string.)

The **seq***seq-value* parameter is optional and specifies the sequence number for the AS-path list. If you do not specify a sequence number, the software numbers in increments of 5, beginning with number 5. The software interprets the entries in an AS-path list in numerical order, beginning with the lowest sequence number.

The **deny** and **permit** parameters specify the action the software takes if the AS-path list for a route matches a match clause in this ACL. To configure the AS-path match clauses in a route map, use the match as-path command.

The *regular-expression* parameter specifies the AS path information you want to permit or deny to routes that match any of the match clauses within the ACL. You can enter a specific AS number or use a regular expression.

The **neighbor** command uses the **filter-list** parameter to apply the AS-path ACL to the neighbor.

Using regular expressions

Use a regular expression for the *as-path* parameter to specify a single character or multiple characters as a filter pattern. If the AS-path matches the pattern specified in the regular expression, the filter evaluation is true; otherwise, the evaluation is false.

You can also include special characters that influence the way the software matches the AS-path against the filter value.

To filter on a specific single-character value, enter the character for the *as-path* parameter. For example, to filter on AS-paths that contain the letter "z", enter the following command:

```
device(config-bgp)# ip as-path access-list acl1 permit z
```

To filter on a string of multiple characters, enter the characters in brackets. For example, to filter on AS-paths that contain "x", "y", or "z", enter the following command.

```
device(config-bgp)# ip as-path access-list acl1 permit [xyz]
```

BGP4 Special characters

When you enter a single-character expression or a list of characters, you also can use the special characters listed in "Using regular expressions." The description for each character includes an example. Some special characters must be placed in front of the characters they control and others must be placed after the characters they control. The examples show where to place the special character.

TABLE 2 BGP4 special characters for regular expressions

Character	Operation
.	The period matches on any single character, including a blank space. For example, the following regular expression matches for "aa", "ab", "ac", and so on, but not just "a". a.
*	The asterisk matches on zero or more sequences of a pattern. For example, the following regular expression matches on an AS-path that contains the string "1111" followed by any value: 1111*
+	The plus sign matches on one or more sequences of a pattern. For example, the following regular expression matches on an AS-path that contains a sequence of "g"s, such as "deg", "degg", "deggg", and so on: deg+
?	The question mark matches on zero occurrences or one occurrence of a pattern. For example, the following regular expression matches on an AS-path that contains "dg" or "deg": de?g
^	A caret (when not used within brackets) matches on the beginning of an input string. For example, the following regular expression matches on an AS-path that begins with "3": ^3
$	A dollar sign matches on the end of an input string. For example, the following regular expression matches on an AS-path that ends with "deg": deg$
_	An underscore matches on one or more of the following: • , (comma) • { (left curly brace) • } (right curly brace) • ((left parenthesis) •) (right parenthesis) • The beginning of the input string • The end of the input string • A blank space For example, the following regular expression matches on "100" but not on "1002", "2100", and so on. _100_

TABLE 2 BGP4 special characters for regular expressions (Continued)

Character	Operation
[ ]	Square brackets enclose a range of single-character patterns. For example, the following regular expression matches on an AS-path that contains "1", "2", "3", "4", or "5": [1-5] You can use the following expression symbols within the brackets. These symbols are allowed only inside the brackets: • ^ - The caret matches on any characters except the ones in the brackets. For example, the following regular expression matches on an AS-path that does not contain "1", "2", "3", "4", or "5":[^1-5] • - The hyphen separates the beginning and ending of a range of characters. A match occurs if any of the characters within the range is present. Refer to the example above.
|	A vertical bar (sometimes called a pipe or a "logical or") separates two alternative values or sets of values. The AS-path can match one or the other value. For example, the following regular expression matches on an AS-path that contains either "abc" or "defg": (abc)|(defg) **NOTE** The parentheses group multiple characters to be treated as one value. Refer to the following row for more information about parentheses.
( )	Parentheses allow you to create complex expressions. For example, the following complex expression matches on "abc", "abcabc", or "abcabcabcdefg", but not on "abcdefgdefg": ((abc)+)|((defg)?)

To filter for a special character instead of using the special character as described in "Using regular expressions," enter "\" (backslash) in front of the character. For example, to filter on AS-path strings containing an asterisk, enter the asterisk portion of the regular expression as "*".

```
device(config-bgp-router)# ip as-path access-list acl2 deny \*
```

To use the backslash as a string character, enter two slashes. For example, to filter on AS-path strings containing a backslash, enter the backslash portion of the regular expression as "\\".

```
device(config-bgp-router)# ip as-path access-list acl2 deny \\
```

BGP4 filtering communities

You can filter routes received from BGP4 neighbors based on community names.

A community is an optional attribute that identifies the route as a member of a user-defined class of routes. Community names are arbitrary values made of two five-digit integers joined by a colon. You determine what the name means when you create the community name as a route attribute. Each string in the community name can be a number from 0 through 65535.

This format allows you to easily classify community names. For example, a common convention used in community naming is to configure the first string as the local AS and the second string as the unique community within that AS. Using this convention, communities 1:10, 1:20, and 1:30 can be easily identified as member communities of AS 1.

The device provides the following methods for filtering on community information.

- Community filters
- Community list ACLs

NOTE
The device cannot actively support community filters and community list ACLs at the same time. Use one method or the other but do not mix methods.

NOTE
Once you define a filter or ACL, the default action for communities that do not match a filter or ACL is **deny** . To change the default action to **permit** , configure the last filter or ACL entry as **permit any any** .

Community filters or ACLs can be referred to by match clauses in a route map.

Defining a community ACL

To configure community ACL 1, enter a command such as the following. This command configures a community ACL that permits routes that contain community 123:2.

```
device(config)# ip community-list 1 permit 123:2
```

Syntax: no ip community-list standard *string* [**seq seq-value**] **deny** | **permit** *community-num*

Syntax: no ip community-list standard *string* [**seq seq-value**] **deny** | **permit** *community-num* | *regular-expression*

The *string* parameter specifies the ACL name. (If you enter a number, the CLI interprets the number as a text string.)

The **standard** or **extended** parameter specifies whether you are configuring a standard or extended community ACL. The difference between standard and extended communities is that a standard community ACL does not support regular expressions and an extended one does.

The **seq** *seq-value* parameter is optional and specifies the sequence number for the community list. You can configure up to 199 entries in a community list. If you do not specify a sequence number, the software numbers the entries in increments of 5, beginning with number 5. The software interprets the entries in a community list in numerical order, beginning with the lowest sequence number.

The **deny** and **permit** parameters specify the action the software takes if a route community list matches a match clause in this ACL. To configure the community-list match clauses in a route map, use the **match community** command.

The *community-num* parameter specifies the community type or community number. This parameter can have the following values:

- **num:num** - A specific community number
- **internet** - The Internet community
- **no-export** - The community of sub-autonomous systems within a confederation. Routes with this community can be exported to other sub-autonomous systems within the same confederation but cannot be exported outside the confederation to other autonomous systems or otherwise sent to EBGP neighbors.
- **local-as** - The local sub-AS within the confederation. Routes with this community can be advertised only within the local subAS.
- **no-advertise** - Routes with this community cannot be advertised to any other BGP4 devices at all.

The *regular-expression* parameter specifies a regular expression for matching on community names.

To use a community-list filter, use route maps with the **match community** parameter.

Defining and applying IP prefix lists

An IP prefix list specifies a list of networks. When you apply an IP prefix list to a neighbor, the device sends or receives only a route whose destination is in the IP prefix list. The software interprets the prefix lists in order, beginning with the lowest sequence number.

To configure an IP prefix list and apply it to a neighbor, enter commands such as the following.

```
device(config)# ip prefix-list Routesfor20 permit 10.20.0.0/24
device(config)# router bgp
device(config-bgp)# neighbor 10.10.10.1 prefix-list Routesfor20 out
```

These commands configure an IP prefix list named Routesfor20, which permits routes to network 10.20.0.0/24. The **neighbor** command configures the device to use IP prefix list Routesfor20 to determine which routes to send to neighbor 10.10.10.1. The device sends routes that go to 10.20.x.x to neighbor 10.10.10.1 because the IP prefix list explicitly permits these routes to be sent to the neighbor.

Syntax: [no] ip prefix-list *name* [**seq** *seq-value*] [**description** *string*] **deny** | **permit** *network-addr l mask-bits* [**ge** *ge-value*] [**le** *le-value*]

The *name* parameter specifies the prefix list name. Use this name when applying the prefix list to a neighbor.

The **description** *string* parameter is a text string describing the prefix list.

The **seq** *seq-value* parameter is optional and specifies the sequence number of the IP prefix list. If you do not specify a sequence number, the software numbers the entries in increments of 5, beginning with prefix list entry 5. The software interprets the prefix list entries in numerical order, beginning with the lowest sequence number.

The **deny and permit** parameters specify the action the software takes if a neighbor route is in this prefix list.

The *network-addr* and *mask-bits* parameters specify the network number and the number of bits in the network mask.

You can specify a range of prefix length for prefixes that are more specific than *network-addr* and *mask-bits* .

The prefix-list matches only on this network unless you use the **ge** *ge-value* or **le** *le-value* parameters.

- If you specify only **ge** *ge-value*, the mask-length range is from *ge-value* to 32.
- If you specify only **le** *le-value*, the mask-length range is from length to *le-value* .

The *ge-value* or *le-value* you specify must meet the following condition:

length < ge-value <= le-value <= 32

If you do not specify **ge** *ge-value* or **le** *le-value* , the prefix list matches only on the exact network prefix you specified with the *network-addr* and *mask-bits* parameters.

In the following example, only default routes are allowed:

```
device(config)# ip prefix-list match-default-routes permit 0.0.0.0/0
```

In the following example, only default routes are denied:

```
device(config)# ip prefix-list match-default-routes deny 0.0.0.0/0
```

In the following example, all routes are allowed, including all subnet masks and all prefixes:

```
device(config)# ip prefix-list match-all-routes permit 0.0.0.0/0 le 32
```

NOTE
Be careful to determine exactly which routes you want to allow using a prefix list.

Defining neighbor distribute lists

A neighbor distribute list is a list of BGP4 address filters or ACLs that filter the traffic to or from a neighbor.

To configure a distribute list that uses ACL 1, enter a command such as the following.

```
device(config-bgp)# neighbor 10.10.10.1 distribute-list 1 in
```

This command configures the device to use ACL 1 to select the routes that the device will accept from neighbor 10.10.10.1.

Syntax: [no] neighbor *ip-addr* **distribute-list** *name-or-num* **in** | **out**

The *ip-addr* parameter specifies the neighbor.

The *name-or-num* parameter specifies the name or number of a standard or named ACL.

The **in** and **out** parameters specify whether the distribute list applies to inbound or outbound routes:

- **in** - controls the routes the device will accept from the neighbor.
- **out** - controls the routes sent to the neighbor.

Defining route maps

A route map is a named set of match conditions and parameter settings that the device can use to modify route attributes and to control redistribution of the routes into other protocols. A route map consists of a sequence of instances . If you think of a route map as a table, an instance is a row in that table. The device evaluates a route according to route map instances in ascending numerical order. The route is first compared against instance 1, then against instance 2, and so on. When a match is found, the device stops evaluating the route.

Route maps can contain match clauses and **set** statements. Each route map contains a **permit** or **deny** action for routes that match the match clauses:

- If the route map contains a **permit** action, a route that matches a match statement is permitted; otherwise, the route is denied.
- If the route map contains a **deny** action, a route that matches a match statement is denied.
- If a route does not match any match statements in the route map, the route is denied. This is the default action. To change the default action, configure the last match statement in the last instance of the route map to **permit any any** .
- If there is no match statement, the software considers the route to be a match.
- For route maps that contain address filters, AS-path filters, or community filters, if the action specified by a filter conflicts with the action specified by the route map, the route map action takes precedence over the filter action.

If the route map contains set clauses, routes that are permitted by the route map match statements are modified according to the set clauses.

Match statements compare the route against one or more of the following:

- The route BGP4 MED (metric)
- A sequence of AS-path filters
- A sequence of community filters
- A sequence of address filters
- The IP address of the next hop device
- The route tag
- For OSPF routes only, the route type (internal, external type-1, or external type-2)
- An AS-path ACL
- A community ACL
- An IP prefix list
- An IP ACL

For routes that match all of the match statements, the route map set clauses can perform one or more of the following modifications to the route attributes:

- Prepend AS numbers to the front of the route AS-path. By adding AS numbers to the AS-path, you can cause the route to be less preferred when compared to other routes based on the length of the AS-path.
- Add a user-defined tag an automatically calculated tag to the route.
- Set the community attributes.
- Set the local preference.
- Set the MED (metric).
- Set the IP address of the next-hop device.
- Set the origin to IGP or INCOMPLETE.
- Set the weight.
- Set a BGP4 static network route.

When you configure parameters for redistributing routes into BGP4, one of the optional parameters is a route map. If you specify a route map as one of the redistribution parameters, the device matches the route against the match statements in the route map. If a match is found and if the route map contains set clauses, the device sets the attributes in the route according to the set clauses.

To create a route map, you define instances of the map by a sequence number.

To define a route map, use the procedures in the following sections.

Entering the route map into the software

To add instance 1 of a route map named "GET_ONE" with a permit action, enter the following command.

```
device(config)# route-map GET_ONE permit 1
device(config-routemap GET_ONE)#
```

Syntax: [no] route-map *map-name* **permit** | **deny** *num*

As shown in this example, the command prompt changes to the route map level. You can enter the match and set clauses at this level.

The *map-name* is a string of characters that names the map. Map names can be up to 80 characters in length.

The **permit and deny** parameters specify the action the device will take if a route matches a match statement:

- If you specify **deny**, the device does not advertise or learn the route.
- If you specify **permit**, the device applies the match and set clauses associated with this route map instance.

The *num* parameter specifies the instance of the route map you are defining.

To delete a route map, enter a command such as the following. When you delete a route map, all the permit and deny entries in the route map are deleted.

```
device(config)# no route-map Map1
```

This command deletes a route map named Map1. All entries in the route map are deleted.

To delete a specific instance of a route map without deleting the rest of the route map, enter a command such as the following.

```
device(config)# no route-map Map1 permit 10
```

This command deletes the specified instance from the route map but leaves the other instances of the route map intact.

Specifying the match conditions

Use the following command to define the match conditions for instance 1 of the route map GET_ONE. This instance compares the route updates against BGP4 address filter 11.

```
device(config-routemap GET_ONE)# match address-filters 11
```

Syntax: [no] match [**as-path** *name*] [**community** *acl* **exact-match**] | [**extcommunity** *acl*] | [**ip address** *acl* | **prfix-list** *string*] | [**ip route-source** *acl* | **prefix** *name*] [**metric** *num*]]| [**next-hop** *address-filter-list*] | [**route-type internal** | **external-type1** | **external-type2**] | [**level-1** | **level-2** | **level-1-2**] [**tag tag-value**] | **interface** *interface interface interface* **.. protocol bgp static-networkprotocol bgp externalprotocol bgp internal**

The **as-path***num* parameter specifies an AS-path ACL. You can specify up to five AS-path ACLs. To configure an AS-path ACL, use the **ip as-path access-list** command.

The **community** *num* parameter specifies a community ACL.

NOTE
The ACL must already be configured.

The **community***acl***exact-match** parameter matches a route if (and only if) the route community attributes field contains the same community numbers specified in the match statement.

The **extcommunity** *acl* parameter identifies a set of sites and VRFs that may receive routes that are tagged with the configured route target and site of origin.

The **ip address, next-hop** *acl-num***, prefix-list, and** *string* parameters specify an ACL or IP prefix list. Use this parameter to match based on the destination network or next-hop gateway. To configure an IP ACL for use with this command, use the **ip access-list** command. To configure an IP prefix list, use the **ip prefix-list** command.

The **ip route-source***acl and* **prefix***name* parameters match based on the source of a route (the IP address of the neighbor from which the device learned the route).

The **metric***num* parameter compares the route MED (metric) to the specified value.

The **next-hop** *address-filter-list* parameter compares the IP address of the route next-hop to the specified IP address filters. The filters must already be configured.

The **route-type internal, external-type1, and external-type2** parameters apply only to OSPF routes. These parameters compare the route type to the specified value.

The **level-1** parameter compares IS-IS routes only with routes within the same area. The **level-2** parameter compares IS-IS routes only with routes in different areas, but within a domain. The **level-1-2** parameter compares IS-IS routes with routes in the same area and in different areas, but within a domain.

The **tag***tag-value* parameter compares the route tag to the specified tag value.

The *protocol bgp static-network* parameter matches on BGP4 static network routes.

The *protocol bgp external* parameter matches on eBGP (external) routes.

The *protocol bgp internal* parameter matches on iBGP (internal) routes.

Match examples using ACLs

The following sections contain examples of how to configure route maps that include match statements that match on ACLs.

Matching based on AS-path ACL

To construct a route map that matches based on AS-path ACL 1, enter the following commands.

```
device(config)# route-map PathMap permit 1
device(config-routemap PathMap)# match as-path 1
```

Syntax: [no] match as-path *string*

The *string* parameter specifies an AS-path ACL and can be a number from 1 through 199. You can specify up to five AS-path ACLs.

Matching based on community ACL

To construct a route map that matches based on community ACL 1, enter the following commands.

```
device(config)# ip community-list 1 permit 123:2
device(config)# route-map CommMap permit 1
device(config-routemap CommMap)# match community 1
```

Syntax: [no] match community *string*

The *string* parameter specifies a community list ACL. To configure a community list ACL, use the **ip community-list** command.

Matching based on extcommunity ACL

To construct a route map that matches based on BGP Extended Community attributes in the incoming BGP routes, enter the following commands.

```
device(config)# ip extcommunity-list 1 permit rt 1:3 soo 1:1
device(config)# route-map ExtCommMap permit 20
device(config-routemap CommMap)# match community 1
```

Syntax: [no] match extcommunity *string*

The *string* parameter specifies a community list ACL. To configure a community list ACL, use the **ip community-list** command.

Matching based on destination network

You can use the results of an IP ACL or an IP prefix list as the match condition.

To construct a route map that matches based on destination network, enter commands such as the following.

```
device(config)# route-map NetMap permit 1
device(config-routemap NetMap)# match ip address 1
```

Syntax: [no] match ip address *ACL-name-or-num*

Syntax: [no] match ip address prefix-list *name*

The *ACL-name-or-num* parameter with the first command specifies an IP ACL and can be a number from 1 through 199 or the ACL name if it is a named ACL. Multiple ACLs may be added when seperated by spaces. To configure an IP ACL, use the **ip access-list** or **access-list** command.

The *name* parameter with the second command specifies an IP prefix list name.

Matching based on next-hop device

You can use the results of an IP ACL or an IP prefix list as the match condition.

To construct a route map that matches based on the next-hop device, enter commands such as the following.

```
device(config)# route-map HopMap permit 1
device(config-routemap HopMap)# match ip next-hop 2
```

Syntax: [no] match ip next-hop *string*

Syntax: [no] match ip next-hop prefix-list *name*

The *string* parameter with the first command specifies an IP ACL and can be a number from 1 through 199 or the ACL name if it is a named ACL. To configure an IP ACL, use the **ip access-list** or **access-list** command.

The *name* parameter with the second command specifies an IP prefix list name.

Matching based on the route source

To match a BGP4 route based on its source, use the **match ip route-source** command.

```
device(config)# access-list 10 permit 192.168.6.0 0.0.0.255
device(config)# route-map bgp1 permit 1
device(config-routemap bgp1)# match ip route-source 10
```

The first command configures an IP ACL that matches on routes received from 192.168.6.0/24. The remaining commands configure a route map that matches on all BGP4 routes advertised by the BGP4 neighbors whose addresses match addresses in the IP prefix list. You can add a set clause to change a route attribute in the routes that match. You also can use the route map as input for other commands, such as the **neighbor** and **network** commands and some show commands.

Syntax: [no] match ip route-source *ACL* | **prefix-list** *name*

The *acl* and **prefix-list** *name* parameters specify the name or ID of an IP ACL, or an IP prefix list.

Matching on routes containing a specific set of communities

The device can match routes based on the presence of a community name or number in a route. To match based on a set of communities, configure a community ACL that lists the communities, then compare routes against the ACL.

```
device(config)# ip community-list standard std_1 permit 12:34 no-export
```

```
device(config)# route-map bgp2 permit 1
device(config-routemap bgp2)# match community std_1 exact-match
```

The first command configures a community ACL that contains community number 12:34 and community name no-export. The remaining commands configure a route map that matches the community attributes field in BGP4 routes against the set of communities in the ACL. A route matches the route map only if the route contains all the communities in the ACL and no other communities.

Syntax: [no] match community *ACL* **exact-match**

The *ACL* parameter specifies the name of a community list ACL. You can specify up to five ACLs. Separate the ACL names or IDs with spaces.

Here is another example.

```
device(config)# ip community-list standard std_2 permit 23:45 56:78
device(config)# route-map bgp3 permit 1
device(config-routemap bgp3)# match community std_1 std_2 exact-match
```

These commands configure an additional community ACL, std_2, that contains community numbers 23:45 and 57:68. Route map bgp3 compares each BGP4 route against the sets of communities in ACLs std_1 and std_2. A BGP4 route that contains either but not both sets of communities matches the route map. For example, a route containing communities 23:45 and 57:68 matches. However, a route containing communities 23:45, 57:68 and 12:34, or communities 23:45, 57:68, 12:34, and no-export does not match. To match, the route communities must be the same as those in exactly one of the community ACLs used by the match community statement.

Matching based on BGP4 static network

The**match** option has been added to the **route-map** command that allows you to match on a BGP4 static network. In the following example, the route-map is configured to match on the BGP4 static network. The device is then configured to advertise to the core BGP4 peer (IP address 192.168.6.0) only the BGP4 static routes and nothing else.

```
device(config)# route-map policygroup3 permit 10
device(config-routemap policygroup3)# match protocol bgp static-network
device(config-routemap policygroup3)# set local-preference 150
device(config-routemap policygroup3)# set community no-export
device(config-routemap policygroup3)# exit
device(config)# router bgp
device(config-bgp)# neighbor 192.168.6.0 route-map out policymap3
```

Syntax: [no] match protocol bgp [**external** | **internal** | **static-network**]

The **match protocol bgp external** option will match the eBGP routes.

The **match protocol bgp internal** option will match the iBGP routes.

The **match protocol bgp static-network** option will match the static-network BGP4 route, applicable at BGP4 outbound policy only.

Matching based on interface

The *match* option has been added to the **route-map** command that distributes any routes that have their next hop out one of the interfaces specified. This feature operates with the following conditions:

- The **match interface** option can only use the interface name (for example ethernet 1/2) and not the IP address as an argument.
- The **match interface** option is only effective during redistribution and does not apply for other route map usage such as: bgp outbound route update policy.
- The **match interface** option can be applied to other types of redistribution such as redistributing OSPF routes to BGP4, or filtering out all OSPF routes that point to a specific interface.

To configure the match-interface option, use the following command.

```
device(config)# route-map test-route permit 99
device(config-routemap test-route)# match interface ethernet 1/1 eth 3/2
device(config-routemap test-route)# exit
```

Syntax: [no] match interface *interface interface* **...**

The *interface* variable specifies the interface that you want to use with the **match interface** command. Up to 5 interfaces of the following types can be specified:

- **ethernet** *slot/port*
- **loopback** *loopback-number*
- **null0**
- **tunnel** *tunnel-ID*
- **ve** *ve-ID*

Setting parameters in the routes

Use the following command to define a set clause that prepends an AS number to the AS path on each route that matches the corresponding match statement.

```
device(config-routemap GET_ONE)# set as-path prepend 65535
```

Syntax: [no] set [**as-path** [**prepend** *as-num,as-num,...*]] | [**automatic-tag**] | [**comm-list** *acl* **delete**] | [**community** *num* **:** *num* | *num* | **additive** | **local-as** | **no-advertise** | **no-export**] | [**dampening** [*half-life reuse suppress max-suppress-time*]] [**ip next hop** *ip-addr*] [**ip next-hop peer-address**] | [**local-preference** *num*] | [**metric** [**+** | **-**] *num* | **none**] | [**metric-type type-1** | **type-2**] | **external** [**metric-type internal**] | [**next-hop** *ip-addr*] | [**origin igp** | **incomplete**] | [**tag**] | [**weight** *num*]

The **as-path prepend***num,num,...* parameter adds the specified AS numbers to the front of the AS-path list for the route. The range of num values is 1 - 65535 for two-byte ASNs and 1 - 4294967295 if AS4s have been enabled.

The **automatic-tag** parameter calculates and sets an automatic tag value for the route.

NOTE
This parameter applies only to routes redistributed into OSPF.

The **comm-list** parameter deletes a community from the community attributes field for a BGP4 route.

The **community** parameter sets the community attribute for the route to the number or well-known type you specify.

The **dampening** [*half-life reuse suppress max-suppress-time*] parameter sets route dampening parameters for the route. The *half-life* parameter specifies the number of minutes after which the route penalty becomes half its value. The *reuse* parameter specifies how low a route penalty must become before the route becomes eligible for use again after being suppressed. The *suppress* parameter specifies how high a route penalty can become before the device suppresses the route. The *max-suppress-time* parameter specifies the maximum number of minutes that a route can be suppressed regardless of how unstable it is.

The **ip next hop** *ip-addr* parameter sets the next-hop IP address for route that matches a match statement in the route map.

The **ip next-hop peer-address** parameter sets the BGP4 next hop for a route to the neighbor address.

The **local-preference** *num* parameter sets the local preference for the route. You can set the preference to a value from 0 through 4294967295.

The **metric** [**+** | **-**] *num* | **none** parameter sets the MED (metric) value for the route. The default MED value is 0. You can set the preference to a value from 0 through 4294967295.

- **set metric** *num* - Sets the metric for the route to the number you specify.
- **set metric +** *num* - Increases route metric by the number you specify.
- **set metric -** *num* - Decreases route metric by the number you specify.
- **set metric none** - Removes the metric from the route (removes the MED attribute from the BGP4 route).

The **metric-type type-1** and **type-2** parameters change the metric type of a route redistributed into OSPF.

The **metric-type internal** parameter sets the route MED to the same value as the IGP metric of the BGP4 next-hop route. The parameter does this when advertising a BGP4 route to an EBGP neighbor.

The **next-hop** *ip-addr* parameter sets the IP address of the route next-hop device.

The **origin igp incomplete** parameter sets the route origin to IGP or INCOMPLETE.

The **tag***tag-value* parameter sets the route tag. You can specify a tag value from 0 through 4294967295.

NOTE
This parameter applies only to routes redistributed into OSPF.

NOTE
You also can set the tag value using a table map. The table map changes the value only when the device places the route in the IP route table instead of changing the value in the BGP4 route table.

The **weight** *num* parameter sets the weight for the route. The range for the weight value is 0 through 4294967295.

Setting a BGP4 route MED to equal the next-hop route IGP metric

To set a route's MED to the same value as the IGP metric of the BGP4 next-hop route, when advertising the route to a neighbor, enter commands such as the following.

```
device(config)# access-list 1 permit 192.168.9.0 0.0.0.255
device(config)# route-map bgp4 permit 1
device(config-routemap bgp4)# match ip address 1
device(config-routemap bgp4)# set metric-type internal
```

The first command configures an ACL that matches on routes with destination network 192.168.9.0. The remaining commands configure a route map that matches on the destination network in ACL 1, then sets the metric type for those routes to the same value as the IGP metric of the BGP4 next-hop route.

Syntax: no set metric-type internal

Setting the next-hop of a BGP4 route

To set the next-hop address of a BGP4 route to a neighbor address, enter commands such as the following.

```
device(config)# route-map bgp5 permit 1
device(config-routemap bgp5)# match ip address 1
device(config-routemap bgp5)# set ip next-hop peer-address
```

These commands configure a route map that matches on routes whose destination network is specified in ACL 1, and sets the next hop in the routes to the neighbor address (inbound filtering) or the local IP address of the BGP4 session (outbound filtering).

Syntax: [no] set ip next-hop peer-address

The value that the software substitutes for **peer-address** depends on whether the route map is used for inbound filtering or outbound filtering:

- When you use the **set ip next-hop peer-address** command in an inbound route map filter, **peer-address** substitutes for the neighbor IP address.
- When you use the **set ip next-hop peer-address** command in an outbound route map filter, **peer-address** substitutes for the local IP address of the BGP4 session.

NOTE
You can use this command for a peer group configuration.

Deleting a community from a BGP4 route

To delete a community from a BGP4 route's community attributes field, enter commands such as the following.

```
device(config)# ip community-list standard std_3 permit 12:99 12:86
device(config)# route-map bgp6 permit 1
device(config-routemap bgp6)# match ip address 1
device(config-routemap bgp6)# set comm-list std_3 delete
```

The first command configures a community ACL containing community numbers 12:99 and 12:86. The remaining commands configure a route map that matches on routes whose destination network is specified in ACL 1, and deletes communities 12:99 and 12:86 from those routes. The route does not need to contain all the specified communities in order for them to be deleted. For example, if a route contains communities 12:86, 33:44, and 66:77, community 12:86 is deleted.

Syntax: [no] set comm-list *ACL* **delete**

The *ACL* parameter specifies the name of a community list ACL.

Using a table map to set the tag value

Route maps that contain set statements change values in routes when the routes are accepted by the route map. For inbound route maps (route maps that filter routes received from neighbors), the routes are changed before they enter the BGP4 route table.

For tag values, if you do not want the value to change until a route enters the IP route table, you can use a table map to change the value. A table map is a route map that you have associated with the IP routing table. The device applies the set statements for tag values in the table map to routes before adding them to the route table.

To configure a table map, you first configure the route map, then identify it as a table map. The table map does not require separate configuration. You can have one table map.

NOTE
Use table maps only for setting the tag value. Do not use table maps to set other attributes. To set other route attributes, use route maps or filters.

To create a route map and identify it as a table map, enter commands such as following. These commands create a route map that uses an address filter. For routes that match the IP prefix list filter,

the route map changes the tag value to 100 and is then considered as a table map. This route map is applied only to routes the device places in the IP route table. The route map is not applied to all routes. This example assumes that IP prefix list p11 has already been configured.

```
device(config)# route-map TAG_IP permit 1
device(config-routemap TAG_IP)# match ip address prefix-list p11
device(config-routemap TAG_IP)# set tag 100
device(config-routemap TAG_IP)# router bgp

device(config-bgp)# table-map TAG_IP
```

Configuring cooperative BGP4 route filtering

By default, the device performs all filtering of incoming routes locally, on the device itself. You can use cooperative BGP4 route filtering to cause the filtering to be performed by a neighbor before it sends the routes to the device. Cooperative filtering conserves resources by eliminating unnecessary route updates and filter processing. For example, the device can send a deny filter to a neighbor, which the neighbor uses to filter out updates before sending them to the device. The neighbor saves the resources it would otherwise use to generate the route updates, and the device saves the resources it would use to filter out the routes.

When you enable cooperative filtering, the device advertises this capability in its Open message to the neighbor when initiating the neighbor session. The Open message also indicates whether the device is configured to send filters, receive filters, or both, and the types of filters it can send or receive. The device sends the filters as Outbound Route Filters (ORFs) in route refresh messages.

To configure cooperative filtering, perform the following tasks on the device and on the BGP4 neighbor:

- Configure the filter.

NOTE
Cooperative filtering is currently supported only for filters configured using IP prefix lists.

- Apply the filter as an inbound filter to the neighbor.
- Enable the cooperative route filtering feature on the device. You can enable the device to send ORFs to the neighbor, to receive ORFs from the neighbor, or both. The neighbor uses the ORFs you send as outbound filters when it sends routes to the device. Likewise, the device uses the ORFs it receives from the neighbor as outbound filters when sending routes to the neighbor.
- Reset the BGP4 neighbor session to send and receive ORFs.
- Perform these steps on the other device.

NOTE
If the device has inbound filters, the filters are still processed even if equivalent filters have been sent as ORFs to the neighbor.

Enabling cooperative filtering

To configure cooperative filtering, enter commands such as the following.

```
device(config)# ip prefix-list Routesfrom10234 deny 10.20.0.0/24
device(config)# ip prefix-list Routesfrom10234 permit 0.0.0.0/0 le 32
device(config)# router bgp
device(config-bgp)# neighbor 10.2.3.4 prefix-list Routesfrom1234 in
device(config-bgp)# neighbor 10.2.3.4 capability orf prefixlist send
```

The first two commands configure statements for the IP prefix list Routesfrom1234. The first command configures a statement that denies routes to 10.20.20./24. The second command configures a statement that permits all other routes. Once you configure an IP prefix list statement, all routes not explicitly permitted by statements in the prefix list are denied.

The next two commands change the CLI to the BGP4 configuration level, then apply the IP prefix list to neighbor 10.2.3.4. The last command enables the device to send the IP prefix list as an ORF to neighbor 10.2.3.4. When the device sends the IP prefix list to the neighbor, the neighbor filters out the 10.20.0.x routes from its updates to the device. This assumes that the neighbor is also configured for cooperative filtering.

Syntax: [no] neighbor *ip-addr* | *peer-group-name* **capability orf prefixlist** [**send** | **receive**]

The ip-addr | peer-group-name parameters specify the IP address of a neighbor or the name of a peer group of neighbors.

The **send and receive** parameters specify the support you are enabling:

- **send** - The device sends the IP prefix lists to the neighbor.
- **receive** - The device accepts filters from the neighbor.

If you do not specify the capability, both capabilities are enabled.

The **prefixlist** parameter specifies the type of filter you want to send to the neighbor.

NOTE
The current release supports cooperative filtering only for filters configured using IP prefix lists.

Sending and receiving ORFs

Cooperative filtering affects neighbor sessions that start after the filtering is enabled, but do not affect sessions that are already established.

To activate cooperative filtering, reset the session with the neighbor. This is required because the cooperative filtering information is exchanged in Open messages during the start of a session.

To place a prefix-list change into effect after activating cooperative filtering, perform a soft reset of the neighbor session. A soft reset does not end the current session, but sends the prefix list to the neighbor in the next route refresh message.

NOTE
Make sure cooperative filtering is enabled on the device and on the neighbor before you send the filters.

To reset a neighbor session and send ORFs to the neighbor, enter a command such as the following.

```
device# clear ip bgp neighbor 10.2.3.4
```

This command resets the BGP4 session with neighbor 10.2.3.4 and sends the ORFs to the neighbor. If the neighbor sends ORFs to the device, the device accepts them if the send capability is enabled.

To perform a soft reset of a neighbor session and send ORFs to the neighbor, enter a command such as the following.

```
device# clear ip bgp neighbor 10.2.3.4 soft in prefix-list
```

Syntax: clear ip bgp neighbor *ip-addr* [**soft in prefix-filter** | **soft in prefix-list**]

If you use the **soft in prefix-filter** parameter, the device sends the updated IP prefix list to the neighbor as part of its route refresh message to the neighbor.

NOTE
If the device or the neighbor is not configured for cooperative filtering, the command sends a normal route refresh message.

Displaying cooperative filtering information

You can display the following cooperative filtering information:

- The cooperative filtering configuration on the device.
- The ORFs received from neighbors.

To display the cooperative filtering configuration on the device, enter a command such as the following.

```
device# show ip bgp neighbor 10.10.10.1
1   IP Address: 10.10.10.1, AS: 65200 (IBGP), RouterID: 10.10.10.1
    State: ESTABLISHED, Time: 0h0m7s, KeepAliveTime: 60, HoldTime: 180
       RefreshCapability: Received
       CooperativeFilteringCapability: Received
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
       Sent    : 1       0       1         0            1
       Received: 1       0       1         0            1
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                 Tx: ---         ---            Rx: ---        ---
    Last Connection Reset Reason:Unknown
    Notification Sent:     Unspecified
    Notification Received: Unspecified
    TCP Connection state: ESTABLISHED
       Byte Sent:   110, Received: 110
       Local host:  10.10.10.2, Local  Port: 8138
       Remote host: 10.10.10.1, Remote Port: 179
       ISentSeq:        460  SendNext:        571  TotUnAck:          0
       TotSent:         111  ReTrans:           0  UnAckSeq:        571
       IRcvSeq:        7349  RcvNext:        7460  SendWnd:       16384
       TotalRcv:        111  DupliRcv:          0  RcvWnd:        16384
       SendQue:           0  RcvQue:            0  CngstWnd:       5325
```

Syntax: show ip bgp neighbor *ip-addr*

To display the ORFs received from a neighbor, enter a command such as the following:

```
device# show ip bgp neighbor 10.10.10.1 received prefix-filter
ip prefix-list 10.10.10.1: 4 entries
     seq 5 permit 10.10.0.0/16 ge 18 le 28
     seq 10 permit 10.20.10.0/24
     seq 15 permit 10.0.0.0/8 le 32
     seq 20 permit 10.10.0.0/16 ge 18
```

Syntax: show ip bgp neighbor *ip-addr* **received prefix-filter**

Four-byte Autonomous System Numbers (AS4)

This section describes the reasons for enabling four-byte autonomous system numbers (AS4s). AS4s are supported by default. You can specify and view AS4s by default and using the enable facility described in this section. However, not all devices in a network are always capable of utilizing AS4s. The act of enabling them on the local device initiates a facility for announcing the capability and negotiating its use with neighbors. If you do not enable AS4s on a device, other devices do not know that this device is sending them.

The system uses a hierarchy to prioritize the utilization of the AS4 capability. The prioritization depends on the CLI configuration commands. AS4s can be enabled and configured at the level of a neighbor, a peer group, or globally for the entire device, according to the following bottom-up hierarchy:

- If a neighbor has no configuration for AS4s but it belongs to a peer group, the neighbor uses the configuration from the peer group. For example, if you configure a neighbor but do not include a specification for AS4s, one of the following applies:
  - The neighbor uses the AS4 configuration for a peer group if it belongs to a peer group.
  - The neighbor uses the device configuration if it does not belong to a peer group or the peer group has no AS4 configuration.
- If a peer group has no configuration for AS4s, it can use the global configuration of the device. If the device has no configuration for AS4s, then a neighbor or peer group without a configuration for AS4s use the device default--no announcement or negotiation of AS4s.
- If a neighbor belongs to peer group with an AS4 configuration but you want that neighbor to be disabled or have a different AS4 configuration, the neighbor AS4 configuration overrides the peer group configuration. For example, you can ensure that neighbor has no AS4 announcement and negotiation activity even though the peer group is enabled for AS4 capability.

NOTE
The configuration for AS4 can be enabled, disabled, or can have no explicit configuration.

CLI commands allow you to disable AS4s on an entity whose larger context has AS4s enabled. For example, you can use a CLI command to disable AS4s on a neighbor that is a member of a peer group that is enabled for AS4s.

Normally, AS4s are sent only to a device, peer group, or neighbor that is similarly configured for AS4s. If a AS4 is configured for a local-autonomous systemS, the system signals this configuration by sending AS_TRANS in the My Autonomous System field of the OPEN message. However, if the AS4 capability for a neighbor is disabled, the local device does not send the four-byte Autonomous System number capability to the neighbor.

Enabling AS4 numbers

This section describes how to enable the announcement and negotiation of AS4s and describes the different types of notation that you can use to represent a AS4.

You can enable AS4s on a device, a peer group, and a neighbor. For global configuration, the **capability** command in the BGP4 configuration context enables or disables AS4 support. For a peer group or a neighbor, **capability** is a keyword for the **neighbor** command. In addition to enabling AS4s for a neighbor or a peer group, you can also use the combination of the **capability** keyword and the optional **enable** or **disable** keyword to disable this feature in a specific case where the AS4s are enabled for a larger context. The Neighbor configuration of AS4s section illustrates this capability.

Global AS4 configuration

To enable AS4s globally, use the **capability** command in the BGP4 configuration context as shown.

```
device(config-bgp)# capability as4 enable
```

Syntax: [no] capability as4 enable | **disable**

The **no** form of the **capability** command deletes the announcement and negotiation configuration of AS4s (if it has been enabled) at the global level. Using the regular form of the command with the **disable** keyword has the same effect on the global configuration. Disabling or using the **no** form of the command does not affect the configuration at the level of a peer or neighbor.

The consequences of choosing between the **enable** or **disable** keyword are reflected in the output of the **show running configuration** command.

Peer group configuration of AS4s

To enable AS4s for a peer group, use the **capability** keyword with the **neighbor** command in the BGP4 configuration context, as the following example for the Peergroup_1 peer group illustrates.

```
device(config-bgp)# neighbor Peergroup_1 capability as4 enable
```

Syntax: [no] neighbor *peer-group-name* **capability as4 enable** | **disable**

The **no** form of the **neighbor** command along with the **capability and as4** keywords disables the announcement and negotiation of AS4s in the named peer group. Using the regular form of the command with the **disable** keyword has the same effect on the neighbor configuration.

The consequences using the **enable** or **disable** keywords are reflected in the output of the **show running configuration** command. However, if the peer group configuration omits an explicit AS4 argument, the **show running configuration** output will not contain AS4 information.

Neighbor configuration of AS4s

To enable AS4s for a neighbor, use the **capability** and **as4** keywords with the **neighbor** command in the BGP4 configuration context, as the following example for IP address 1.1.1.1 illustrates.

```
device(config-bgp)# neighbor 1.1.1.1 capability as4 enable
```

Syntax: [no] neighbor *IPaddress* **capability as4 enable** | **disable**

The **no** form of the **neighbor** command with the **capability** and **as4** keywords deletes the neighbor-enable for AS4s.

The consequences of using the **enable** or **disable** keywords are reflected in the output of the **show running configuration** command. However, if the neighbor configuration omits an explicit AS4 argument, the **show running configuration** output will not contain AS4 information.

To disable AS4s on a particular neighbor within a peer group that is enabled for AS4s, enter a command similar to the following.

```
device(config-bgp)# neighbor 1.1.1.1 capability as4 disable
```

Specifying the local AS number

The local autonomous system number (ASN) identifies the autonomous system where the BGP4 device resides.

Normally, AS4s are sent only to a device, peer group, or neighbor that is similarly configured for AS4s. Typically, if you try to set up a connection from an AS4-enabled device to a device that processes only two-byte ASNs, the connection fails to come up unless you specify the reserved ASN 23456 as the local ASN to send to the far-end device.

To set the local autonomous system number, enter commands such as the following.

```
device(config)# router bgp
BGP4: Please configure 'local-as' parameter in order to enable BGP4.
device(config-bgp)# local-as 100000
device(config-bgp)# write memory
```

Syntax: [no] local-as *num*

The *num* parameter specifies a local ASN in the range 1 - 4294967295. No default exists for *num* . ASNs 64512 - 65535 are the well-known private BGP4 autonomous system numbers and are not advertised to the Internet community.

Route-map set commands and AS4s

You can prepend an AS4 number to an autonomous system path or make the autonomous system number a tag attribute for a route map as shown here.

```
device(config-routemap test)# set as-path prepend 7701000
```

Syntax: [no] set as-path prepend *num,num* **, ...** | **tag**

Use the **no** form of this command to remove the configuration.

NOTE
If the autononous system path for a route map has prepended ASNs and you want to use the **no** form of the command to delete the configuration, you must include the prepended ASNs in the **no set as-path** entry. For example, if 70000 and 70001 have been prepended to a route map, enter **no set as-path prepend 70000 70001** . As a shortcut, in the configuration context of a particular route map, you can also copy and paste ASNs from the output of **show** commands, such as **show route-map** or **show ip bgp route** .

Use the **prepend** keyword to prepend one or more ASNs. The maximum number of ASNs that you can prepend is 16. The range for each ASN is 1 - 4294967295.

Entering the **tag** keyword sets the tag as an AS-path attribute.

You can specify a route target (rt) or a site of origin (soo) for an extended community, as shown in the following example.

```
device(config-routemap test)# set extcommunity rt 7701000:10
```

Syntax: [no] set extcommunity rt *asn:nn* | *ip-address:nn* |**soo** *asn:nn* | *ip-address:nn*

The **rt** keyword specifies a route target in the form of a route ID. The route ID can be an ASN or IP address. The second part of the route ID is a user-specific numeric variable *nn*. The ASN can be a maximum of 4 bytes (in the range 1 – 4294967295). If you specify an AS4 or IP address, the *nn* variable is limited to a maximum length of 2 bytes. If the feature for announcing and negotiating AS4 is disabled, *nn* can be 4 bytes.

The **soo** keyword specifies a site or origin in the form of a route ID. The route ID can be an ASN4 or IP address. The second part of the route ID is a user-specific numeric variable *nn*. The AS4 can be a maximum of 4 bytes (in the range 1 – 4294967295). If you specify an AS4 or IP address, the *nn* variable is limited to a maximum length of 2 bytes. If the feature for announcing and negotiating AS4 is disabled, *nn* can be 4 bytes.

Clearing BGP4 routes to neighbors

You can clear BGP4 connections using the AS4 as an argument with the **clear ip bgp neighbor** command in the configuration context level of the CLI. as shown.

```
device(config)# clear ip bgp neighbor 80000
```

Syntax: clear ip bgp neighbor all | *ip-addr* | *peer-group-name* | *as-num* [**last-packet-with-error** | **notification-errors** | [**soft** [**in** | **out**] | **soft-outbound**]

The neighbor specification is either all, *ip-addr* , *peer-group-name* , or *as-num* . The **all** parameter specifies all neighbors. The *ip-addr* parameter specifies a neighbor by its IP interface with the device. The *peer-group-name* specifies all neighbors in a specific peer group. The *as-num* parameter specifies all neighbors within the specified AS. After choosing one mandatory parameter, you can choose an optional parameter.

The **soft in** and **soft out** parameters determine whether to refresh the routes received from the neighbor or the routes sent to the neighbor. If you do not specify **in** or **out** , the device performs a soft refresh in both options:

- **soft in** performs one of the following actions on inbound routes, according to other configuration settings:
  - If you enabled soft reconfiguration for the neighbor or peer group, **soft in** updates the routes by comparing the route policies against the route updates that the device has stored. Soft reconfiguration does not request additional updates from the neighbor or otherwise affect the session with the neighbor.
  - If you did not enable soft reconfiguration, **soft in** requests the entire BGP4 route table on the neighbor (Adj-RIB-Out), then applies the filters to add, change, or exclude routes.
  - If a neighbor does not support dynamic refresh, **soft in** resets the neighbor session.
- **soft out** updates all outbound routes and then sends the entire BGP4 route table for the device (Adj-RIB-Out) to the neighbor after the device changes or excludes the routes affected by the filters.
- The **soft-outbound** parameter updates all outbound routes by applying the new or changed filters, but sends only the existing routes affected by the new or changed filters to the neighbor.

NOTE
Use **soft-outbound** only if the outbound policy is changed. The **soft-outbound** parameter updates all outbound routes by applying the new or changed filters. However, the device sends to the neighbor only the existing routes that are affected by the new or changed filters.The **soft out** parameter updates all outbound routes and then sends the entire BGP4 route table on the device to the neighbor after the device changes or excludes the routes affected by the filters.

AS4 notation

A AS4 can appear in either a plain or a dot notation format in the output of **show** commands. To select one of these formats, specify the format before entering the **show** command. This section defines these formats and describes how to select a format. The following notations are currently supported:

- With the default **asplain** , the ASN is a decimal integer in the range 1 - 4294967295.
- With **asdot** +, all ASNs are two integer values joined by a period character in the following format:

<high order 16-bit value in decimal>.<low order 16-bit value in decimal>

Using the asdot+ notation, an autonomous system number of value 65526 is represented as the string "0.65526," and an autonomous system number of value 65546 is represented as the string "1.10."

- With **asdot,** an ASN less than 65536 uses the asplain notation (and represents autonomous system number values equal to or greater than 65536 using the asdot+ notation). Using the asdot notation, ASN 65526 is represented as the string "65526," and ASN 65546 is represented as the string "1.10".

NOTE
You can enter autonomous system numbers in any format. However, if you want the **asdot** or the **asdot+** format to appear in the output of a **show** command, you must specify these in the CLI.

NOTE
Remember that autonomous system path matching that uses regular expression is based on the configured autonomous system format.

The following command sequences show how to enable the different notations for AS4s and how these notations appear in the output display.

To see ASNs in asplain, use the **show ip bgp** command.

```
device(config)# show ip bgp
Total number of BGP Routes: 1
Status codes:s suppressed, d damped, h history, * valid, > best, i internal, S stale
Origin codes: i - IGP, e - EGP,? - incomplete
Network          Next Hop      Metric LocPrf Weight Path
*>  47.1.1.0/24   192.168.1.5   1      100    0      90000 100 200 65535 65536 65537
65538 65539 75000 ?
```

To specify **asdot** notation before displaying IP BGP4 information, use the **as-format** command.

```
device(config)# as-format asdot
device(config)# show ip bgp
Total number of BGP Routes: 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal, S stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network          Next Hop      Metric LocPrf Weight Path
*>  10.1.1.0/24  192.168.1.5   1      100    0      1.24464 100 200 65535
1.0 1.1 1.2 1.3 1.9464 ?
```

Syntax: [no] as-format asplain | **asdot** | **asdot+**

The default is **asplain** and can be restored using the **no** version of the command, if the CLI is currently using **asdot** or **asdot+** .

To activate **asdot+** notation, enter **as-format asdot+** in the CLI.

```
device(config)# as-format asdot+
device(config)# show ip bgp
Total number of BGP Routes: 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal, S stale
Origin codes: i - IGP, e - EGP,? - incomplete
    Network        Next Hop      Metric LocPrf Weight Path
*>  10.1.1.0/24    192.168.1.5   1      100    0      1.24464 0.100 0.200
0.65535 1.0 1.1 1.2 1.3 1.9464 ?
```

BGP4 AS4 attribute errors

This section describes the handling of the confederation path segments in the AS4_PATH attribute, and also specifies the error handling for the new attributes.

To support AS4, the following attributes: AS4_PATH and AS4_Aggregator were specified in RFC 4893. Confederation path segments in an AS4_PATH are discarded and if there are any other errors such as: *attribute length* , *flag* , confederation segments after AS_SEQ/AS_SET, Invalid segment types and More than one AS4_PATH in these new attributes, the attribute is discarded and the error is logged.

Error logs

The device generates a log when it encounters attribute errors in AS4_PATH and AS4_AGGREGATOR.

NOTE
Logging of errors is rate-limited to not more than one message for every two minutes. Some errors may be lost due to this rate-limiting.

Sample log messages for various attribute errors are shown here.

Attribute length error (ignore the AS4_PATH)

```
SYSLOG: Sep  9 19:02:03:<11>mu2, BGP: From Peer 192.168.1.1 received invalid
AS4_PATH attribute length (3) - entire AS4_PATH ignored
```

Attribute flag error (ignore the AS4_PATH)

```
SYSLOG: Sep  9 19:02:03:<11>mu2, BGP: From Peer 192.168.1.1 received invalid
AS4_PATH attribute flag (0x40) - entire AS4_PATH ignored
```

Confederation segments after AS_SEQ/AS_SET (ignore the AS4_PATH)

```
SYSLOG: Sep  9 19:02:03:<11>mu2, BGP: From Peer 192.168.1.1 received invalid Confed
info in AS4_PATH (@byte 43) - entire AS4_PATH not ignored
```

Invalid segment types (ignore the AS4_ PATH)

```
SYSLOG: Sep  9 19:02:03:<11>mu2, BGP: From Peer 192.168.1.1 received incorrect Seq
type/len in AS4_PATH (@byte 41) - entire AS4_PATH ignored
```

More than one AS4_PATH (Use the first one and ignore the others)

```
SYSLOG: Sep  9 19:02:03:<11>mu2, BGP: From Peer 192.168.1.1 received multiple
AS4_PATH attributes - used first AS4_PATH attribute only
```

Configuring route flap dampening

A route flap is a change in the state of a route, from up to down or down to up. A route state change causes changes in the route tables of the devices that support the route. Frequent route state changes can cause Internet instability and add processing overhead to the devices that support the route.

Route flap dampening helps reduce the impact of route flap by changing the way a BGP4 device responds to route state changes. When route flap dampening is configured, the device suppresses unstable routes until the number of route state changes drops enough to meet an acceptable degree of stability. The Brocade implementation of route flap dampening is based on RFC 2439.

Route flap dampening is disabled by default. You can enable the feature globally or on an individual route basis using route maps.

NOTE
The device applies route flap dampening only to routes learned from EBGP neighbors.

The route flap dampening mechanism is based on penalties. When a route exceeds a configured penalty value, the device stops using that route and stops advertising it to other devices. The mechanism also allows route penalties to reduce over time if route stability improves.

The route flap dampening mechanism uses the following parameters:

- Suppression threshold - Specifies the penalty value at which the device stops using the route. Each time a route becomes unreachable or is withdrawn by a BGP4 UPDATE from a neighbor, the route receives a penalty of 1000. By default, when a route penalty is greater than 2000, the device stops using the route. By default, if a route goes down more than twice, the device stops using the route. You can set the suppression threshold to a value from 1 through 20000. The default is 2000.
- Half-life - Once a route has been assigned a penalty, the penalty decreases exponentially and decreases by half after the half-life period. The default half-life period is 15 minutes. The software reduces route penalties every five seconds. For example, if a route has a penalty of 2000 and does not receive any more penalties during the half-life, the penalty is reduced to 1000 after the half-life expires. You can configure the half-life to be from 1 - 45 minutes. The default is 15 minutes.
- Reuse threshold - Specifies the minimum penalty a route can have and still be suppressed by the device. If the route penalty falls below this value, the device un-suppresses the route and can use it again. The software evaluates the dampened routes every ten seconds and un-suppresses the routes that have penalties below the reuse threshold. You can set the reuse threshold to a value from 1 through 20000. The default is 750.
- Maximum suppression time - Specifies the maximum number of minutes a route can be suppressed regardless of how unstable the route has been before this time. You can set the parameter to a value from 1 through 20000 minutes. The default is four times the half-life. When the half-life value is set to its default (15 minutes), the maximum suppression time defaults to 60 minutes.

You can configure route flap dampening globally or for individual routes using route maps. If you configure route flap dampening parameters globally and also use route maps, the settings in the route maps override the global values.

Globally configuring route flap dampening

Route flap dampening reduces the amount of route state changes propagated by BGP4 due to unstable routes. This in turn reduces processing requirements.

To enable route flap dampening using the default values, enter the following command.

```
device(config-bgp)# dampening
```

Syntax: [no] dampening [*half-life reuse suppress max-suppress-time*]

The *half-life* parameter specifies the number of minutes after which the penalty for a route becomes half its value. The route penalty allows routes that have remained stable for a period despite earlier instability to eventually become eligible for use again. The decay rate of the penalty is proportional to the value of the penalty. After the half-life expires, the penalty decays to half its value. A dampened route that is no longer unstable can eventually again become eligible for use. You can configure the half-life to be from 1 through 45 minutes. The default is 15 minutes.

The *reuse* parameter specifies how low a penalty for a route must be before the route becomes eligible for use again, after being suppressed. You can set the reuse threshold to a value from 1 through 20000. The default is 750 (0.75, or three-fourths, of the penalty assessed for a one flap).

The *suppress* parameter specifies how high the penalty for a route can be before the device suppresses the route. You can set the suppression threshold to a value from 1 through 20000. The default is 2000 (more than two flaps).

The *max-suppress-time* parameter specifies the maximum number of minutes that a route can be suppressed regardless of how unstable it is. You can set the maximum suppression time to a value from 1 through 255 minutes. The default is 40 minutes.

This example shows how to change the dampening parameters.

```
device(config-bgp)# dampening 20 200 2500 40
```

This command changes the half-life to 20 minutes, the reuse threshold to 200, the suppression threshold to 2500, and the maximum number of minutes a route can be dampened to 40.

NOTE
To change any of the parameters, you must specify all the parameters with the command. To want to leave any parameters unchanged, enter their default values.

Using a route map to configure route flap dampening for specific routes

Route maps enable you to fine tune route flap dampening parameters for individual routes. To configure route flap dampening parameters using route maps, configure BGP4 address filters for each route you want to set the dampening parameters for, then configure route entries that set the dampening parameters for those routes.

Using a route map to configure route flap dampening for a specific neighbor

You can use a route map to configure route flap dampening for a specific neighbor by performing the following tasks:

- Configure an empty route map with no match or set clauses. This route map does not specify particular routes for dampening but does allow you to enable dampening globally when you refer to this route map from within the BGP4 configuration level.
- Configure another route map that explicitly enables dampening. Use a set clause within the route map to enable dampening. When you associate this route map with a specific neighbor, the route map enables dampening for all routes associated with the neighbor. You also can use match clauses within the route map to selectively perform dampening on some routes from the neighbor.

NOTE
You still need to configure the first route map to enable dampening globally. The second route map does not enable dampening by itself; it just applies dampening to a neighbor.

- Apply the route map to the neighbor.

To enable route flap dampening for a specific BGP4 neighbor, enter commands such as the following.

```
device(config)# route-map DAMPENING_MAP_ENABLE permit 1
device(config-routemap DAMPENING_MAP_ENABLE)# exit
device(config)# route-map DAMPENING_MAP_NEIGHBOR_A permit 1
device(config-routemap DAMPENING_MAP_NEIGHBOR_A)# set dampening
device(config-routemap DAMPENING_MAP_NEIGHBOR_A)# exit
device(config)# router bgp
device(config-bgp)# dampening route-map DAMPENING_MAP_ENABLE
device(config-bgp)# neighbor 10.10.10.1 route-map in DAMPENING_MAP_NEIGHBOR_A
```

In this example, the first command globally enables route flap dampening. This route map does not contain any match or set clauses. At the BGP4 configuration level, the **dampening route-map** command refers to the DAMPENING_MAP_ENABLE route map created by the first command, thus enabling dampening globally.

The third and fourth commands configure a second route map that explicitly enables dampening. Notice that the route map does not contain a match clause. The route map implicitly applies to all routes. Since the route map will be applied to a neighbor at the BGP4 configuration level, the route map will apply to all routes associated with the neighbor.

Although the second route map enables dampening, the first route map is still required. The second route map enables dampening for the neighbors to which the route map is applied. However, unless dampening is already enabled globally by the first route map, the second route map has no effect.

The last two commands apply the route maps. The **dampening route-map** command applies the first route map, which enables dampening globally. The **neighbor** command applies the second route map to neighbor 10.10.10.1. Since the second route map does not contain match clauses for specific routes, the route map enables dampening for all routes received from the neighbor.

Removing route dampening from a route

You can un-suppress routes by removing route flap dampening from the routes. The device allows you to un-suppress all routes at once or un-suppress individual routes.

To un-suppress all the suppressed routes, enter the following command at the Privileged EXEC level of the CLI.

```
device# clear ip bgp dampening
```

Syntax: clear ip bgp dampening [*ip-addr ip-mask*]

The **ip-addr** parameter specifies a particular network.

The **ip-mask** parameter specifies the network mask.

To un-suppress a specific route, enter a command such as the following.

```
device# clear ip bgp dampening 10.157.22.0 255.255.255.0
```

This command un-suppresses only the routes for network 10.157.22.0/24.

Displaying and clearing route flap dampening statistics

The software provides many options for displaying and clearing route flap statistics.

Displaying route flap dampening statistics

To display route dampening statistics or all the dampened routes, enter the following command at any CLI level.

```
device# show ip bgp flap-statistics
Total number of flapping routes: 414
    Status Code  >:best d:damped h:history *:valid
    Network            From         Flaps Since    Reuse    Path
h>  10.50.206.0/23   10.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 701
h>  10.255.192.0/20  10.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 7018
h>  10.252.165.0/24  10.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 7018
h>  10.50.208.0/23   10.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 701
h>  10.33.0.0/16     10.90.213.77   1     0 :0 :13 0 :0 :0  65001 4355 1 701
*>  10.17.220.0/24   10.90.213.77   1     0 :1 :4  0 :0 :0  65001 4355 701 62
```

Syntax: show ip bgp flap-statistics [**regular-expression** *regular-expression* | *address mask* [**longer-prefixes**] | **neighbor** *ip-addr*] **as-path-filter** *num*

The **regular-expression** *regular-expression* parameter is a regular expression. Regular expressions are the same ones supported for BGP4 AS-path filters.

The *address mask* parameters specify a particular route. If you also use the optional **longer-prefixes** parameter, all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, if you specify **10.157.0.0 longer** , all routes with the prefix 10.157. or longer (such as 10.157.22.) are displayed.

The **neighbor***ip-addr* parameter displays route flap dampening statistics only for routes learned from the specified neighbor. You also can display route flap statistics for routes learned from a neighbor by entering the following command: **show ip bgp neighbor flap-statistics** .

The **as-path-filter** *num* parameter specifies one or more filters. Only the routes that have been dampened and that match the specified filter or filters are displayed.

TABLE 3 show ip bgp flap-statistics output descriptions

This field	Displays
Total number of flapping routes	The total number of routes in the BGP4 route table that have changed state and have been marked as flapping routes.
Status code	Indicates the dampening status of the route, which can be one of the following: • > - This is the best route among those in the BGP4 route table to the route destination. • d - This route is currently dampened, and unusable. • h - The route has a history of flapping and is unreachable now. • * - The route has a history of flapping but is currently usable.
Network	The destination network of the route.
From	The neighbor that sent the route to the device.
Flaps	The number of flaps the route has experienced.
Since	The amount of time since the first flap of this route.
Reuse	The amount of time remaining until this route will be un-suppressed and can be used again.
Path	Shows the AS-path information for the route.

You also can display all dampened routes by entering the **show ip bgp dampened-paths** command.

Clearing route flap dampening statistics

Clearing the dampening statistics for a route does not change the dampening status of the route. To clear all the route dampening statistics, enter the following command at any level of the CLI.

```
device# clear ip bgp flap-statistics
```

Syntax: clear ip bgp flap-statistics [**regular-expression** *regular-expression* | *address mask* | **neighbor** *ip-addr*]

The parameters are the same as those for the **show ip bgp flap-statistics** command (except the **longer-prefixes** option is not supported).

NOTE
The **clear ip bgp dampening** command not only clears statistics but also un-suppresses the routes.

Generating traps for BGP4

You can enable and disable SNMP traps for BGP4. BGP4 traps are enabled by default.

To enable BGP4 traps after they have been disabled, enter the following command.

```
device(config)# snmp-server enable traps bgp
```

Syntax: [no] snmp-server enable traps bgp

Use the **no** form of the command to disable BGP4 traps.

Configuring BGP4

Once you activate BGP4, you can configure the BGP4 options. There are two configuration levels: global and address family.

At the *global level* , all BGP4 configurations apply to IPv4 and IPv6. Enter this layer using the **device BGP4** command

Under the global level, you specify an address family . Address families separate IPv4 and IPv6 BGP4 configurations. Go to this level by entering the **address-family** command at the device BGP4 level. The command requires you to specify the IPv4 or IPv6 network protocol.

The **address-family** command also requires you to select a sub-address family, which is the type of routes for the configuration. Specify unicast routes.

TABLE 4 IPv4 BGP4 commands for different configuration levels

Command	Global (iPv4 and IPv6)	IPv4 address family unicast	IPv4 address family multicast
address-family	x	x	x
aggregate-address		x	x
always-compare-med	x		
always-propagate		x	
as-path-ignore	x		
bfd		x	
bfd-enable		x	
bgp-redistribute-internal	x		

TABLE 4 IPv4 BGP4 commands for different configuration levels (Continued)

Command	Global (iPv4 and IPv6)	IPv4 address family unicast	IPv4 address family multicast
client-to-client-reflection	x	x	x
cluster-id		x	
compare-med-empty-aspath	x		
compare-routerid	x		
confederation	x		
dampening		x	x
default-information-originate		x	x
default-local-preference	x		
default-metric		x	x
distance	x		
enforce-first-as	x		
exit-address-family	x	x	x
fast-external-fallover	x		
graceful-restart		x	
install-igp-cost		x	
local-as	x		
log-dampening-debug		x	
maxas-limit		x	
maximum-paths		x	
med-missing-as-worst	x		
multipath		x	
neighbor	x	x	x
network		x	x
next-hop-enable-default		x	

TABLE 4 IPv4 BGP4 commands for different configuration levels (Continued)

Command	Global (iPv4 and IPv6)	IPv4 address family unicast	IPv4 address family multicast
next-hop-mpls		x	
next-hop-recursion		x	
redistribute		x	x
rib-route-limit		x	
show	x	x	x
static-network			
table-map		x	x
timers	x		
update-time		x	x

Entering and exiting the address family configuration level

The BGP4 address family contains a unicast or multicast sub-level.

To go to the IPv4 BGP4 unicast address family configuration level, enter the following command.

```
device(config-bgp)# address-family ipv4 unicast
device(config-bgp)#
```

NOTE
The CLI prompt for the global BGP4 level and the BGP4 address-family IPv4 unicast level is the same.

To go to the IPv4 BGP4 multicast address family configuration level, enter the following command.

```
device(config-bgp)# address-family ipv4 multicast
device(config-bgp-ipv4m)#
```

Syntax: [no] address-family ipv4 unicast [**vrf** *vrf-name*] | **ipv4 multicast**

The default is the IPv4 unicast address family level.

The **vrf** option allows you to configure a unicast instance for the VRF specified by the *vrf-name* variable.

To exit an address family configuration level, enter the following command.

```
device(config-bgp)# exit-address-family
device(config-bgp)#
```

Syntax: [no] exit-address-family

BGP route reflector

A BGP device selects a preferred BGP4 route for a specific prefix learned from multiple peers by using the BGP best path selection algorithm, and installs the BGP4 route in the Routing Table Manager (RTM). The BGP device marks the preferred BGP4 route as the best route, and advertises the route to other BGP4 neighbors. Generally, the RTM route table size is larger than the number of unique BGP4 routes in the BGP4 route table. All preferred BGP4 routes are installed in RTM and are marked as the best BGP4 routes.

However, in certain configurations it is possible that the total number of preferred BGP4 routes may exceed the RTM route table size limit. Therefore, some preferred BGP4 routes may not be installed in the RTM, and the BGP device is not able to forward traffic correctly for those BGP4 routes. Those BGP4 routes are not considered as the best BGP4 routes, and are not advertised to other BGP4 neighbors because traffic miss-forwarding or packet drop can occur.

When a BGP device is configured as only a route reflector server, and is not placed directly in the forwarding path, it is possible to mark all preferred BGP4 routes as the best routes to be advertised to other BGP4 neighbors even if the routes are not installed in the RTM. To support the behavior of a BGP device as a route reflector server in such a scenario, use the **always-propagate** command and the **rib-route-limit** command.

NOTE
The **always-propagate** command and the **rib-route-limit** command are supported on Brocade MLX series, Brocade NetIron XMR, Brocade NetIron CER, and Brocade NetIron CES devices.

Configuring BGP route reflector

The **always-propagate** command enables a device to mark a preferred BGP4 route not installed in the RTM as the best route, and advertise the route to other BGP4 neighbors. The same process for outbound route policy continues to apply to all best BGP4 routes. The **rib-route-limit** command limits the number of BGP4 Routing Information Base (RIB) routes that can be installed in the RTM. The RTM must be able to reserve enough entries for Interior Gateway Protocol (IGP) routes because the IGP routes are required by BGP4 to resolve BGP4 next-hop entries. If the RTM is not able to reserve enough entries for IGP routes, BGP4 RIB routes can fill the entire RTM with only BGP4 route entries. The **rib-route-limit** command enables IGP and BGP4 route entries to be installed in the RTM.

NOTE
The **always-propagate** command and the **rib-route-limit** command are configurable in any order under the BGP4 address family configuration level.

Perform the following steps to advertise a preferred BGP4 route not installed in the RTM.

1. Configure a BGP4 unicast route. Enter a command such as the following.

```
device(config-bgp)# address-family ipv4 unicast
```

Syntax: address-family ipv4 unicast [**vrf** *vrf-name*] | **ipv4 multicast**| **ipv6 unicast** | **ipv6 multicast**

NOTE
To configure a BGP4 unicast route for a specified VRF instance, use the **vrf** *vrf-name* parameter. The **vrf***vrf-name* parameter allows you to create a VPN routing or forwarding instance specified by

the *vrf-name* variable. The *vrf-name* variable specifies the name of the VRF instance you want to create.

2. Enter the **always-propagate** command to enable a preferred BGP4 route (not installed in the RTM) to be advertised to other BGP4 neighbors.

```
device(config-bgp)# always-propagate
```

Syntax: always-propagate

3. Enter the **rib-route-limit** command to set the maximum number of BGP4 rib routes that can be installed in the RTM.

```
device(config-bgp)# rib-route-limit 500
```

Syntax: rib-route-limit *decimal*

The *decimal* variable specifies the maximum number of BGP4 rib routes that can be installed in the RTM. The user may enter any number for the *decimal* variable for the **rib-route-limit** command. By default, there is no limit. If the **rib-route-limit** command is set to 0, no BGP4 routes are installed in the RTM. If a BGP4 route is not installed in the RTM because of the configuration set by the **rib-route-limit** command, the **always-propagate** command must be enabled for preferred BGP4 routes to be advertised to the BGP4 neighbors.

If the **rib-route-limit** command is configured to a value that is below the number of BGP4 routes already installed in the RTM, the following warning message is displayed on the console.

```
device(config-bgp)# rib-route-limit 250
The new limit is below the current bgp rib route count. Please use Clear ip bgp
routes command to remove bgp rib routes.
```

You can only use one of the following commands to clear all BGP4 routes in the RTM, and reset the routes for preferred BGP4 routes to be reinstalled in the RTM. Depending on the type of route the **rib-route-limit** command is used for, select from one of the following commands:

- **clear ip bgp routes** command. This command is used to clear IPv4 BGP unicast routes.
- **clear ipv6 bgp routes** command. This command is used to clear IPv6 BGP unicast routes.
- **clear ip mbgp routes** command. This command is used to clear IPv4 MBGP multicast routes.
- **clear ipv6 mbgp routes** command. This command is used to clear IPv6 MBGP multicast routes.

NOTE
It is not guaranteed that the same number of preferred BGP4 routes will be reinstalled in the RTM.

4. Perform the following step to:

- the BGP4 unicast or multicast address family configuration.

```
device(config-bgp-ipv4u)# exit-address-family
```

Syntax: exit-address-family

When you enter the **exit-address-family** command at the address family configuration level, you return to the BGP4 unicast address family configuration level (the default BGP4 level).

Displaying configuration for BGP route reflector

To display the configuration for preferred BGP4 routes not installed in the RTM, use the **show ip bgp route** command as shown in the following example.

```
device(config-bgp)# show ip bgp route
Total number of BGP Routes: 333422
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH m:NOT-INSTALLED-MULTIPATH
       S:SUPPRESSED F:FILTERED s:STALE
```

```
       Prefix           Next Hop       MED    LocPrf   Weight Status
...5      10.12.0.0/24   10.100.100.4          100      0      E
         AS_PATH:48 1994 65148 21948 6461 1239 4837 4808 17431 18245...
```

Syntax: show ip bgp route

In the previous output, BGP4 receives 333,422 routes and the **rib-route-limit** command is configured to 300,000 routes. The **always-propagate** command has not been enabled. However, because the **rib-route-limit** command is configured to allow for 300,000 routes in the RTM, BGP4 installs only 300,000 routes of the 333,422 routes received in the RTM. When the **always-propagate** command is enabled, a preferred BGP4 route not installed in the RTM is now considered as the best BGP4 route to be advertised to other peers. The route is identified by the letter "b" (for NOT-INSTALLED-BEST) in the Status field. However, when the **always-propagate** command is not enabled, the status field displays only the default letter "E", as displayed for BGP4 route 10.12.0.0/24. The letter "B" or "b" is missing from the Status field.

NOTE
The description of the status "b: NOT-INSTALLED-BEST" has changed. The status description for "b: NOT-INSTALLED-BEST" is now: The routes received from the neighbor are the best BGP4 routes to their destinations, but were nonetheless not installed in the IP route table due to the **rib-route-limit** option (or RTM route table size limit), and the **always-propagate** option to allow the propagating of those best BGP routes.

NOTE
Traffic loss on a BGP4 route occurs when a device is advertising preferred BGP4 routes not installed in the RTM as part of the forwarding path.

Because the BGP4 route 10.12.0.0/24 is not considered as the best BGP4 route, the route is not advertised to other BGP4 neighbors.

```
device(config-bgp)# show ip bgp route 10.12.0.0/24
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH m:NOT-INSTALLED-MULTIPATH
       S:SUPPRESSED F:FILTERED s:STALE
       Prefix          Next Hop       MED    LocPrf   Weight Status
1      10.12.0.0/24    10.100.100.4          100      0      E
         AS_PATH: 48 1994 65148 21948 6461 1239 4837 4808 17431 18245
       Last update to IP routing table: 0h16m2s
       No path is selected as BEST route
```

Syntax: show ip bgp route *ip-address*/*prefix*

After enabling the **always-propagate** command, the BGP4 route is now considered the best BGP4 route, even though the route is not installed in the RTM. Because the **rib-route-limit** command was configured to allow for only 300,000 routes in the RTM some preferred BGP4 routes are not installed in the RTM, and are not advertised to other BGP4 neighbors. By enabling the **always-propagate** command, the device is now able to advertise those preferred BGP4 routes to other BGP4 neighbors. In the following example, the Status field displays "bE" indicating that the route is now considered the best BGP4 route for forwarding and will be advertised to other BGP4 neighbors.

```
device(config-bgp)# show ip bgp route 10.12.0.0/24
Number of BGP Routes matching display condition : 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH m:NOT-INSTALLED-MULTIPATH
       S:SUPPRESSED F:FILTERED s:STALE
       Prefix          Next Hop       MED     LocPrf    Weight Status
1      10.12.0.0/24    10.100.100.4           100       0      bE
         AS_PATH: 48 1994 65148 21948 6461 1239 4837 4808 17431 18245
       Last update to IP routing table: 0h12m53s
       Route is to be sent to 1 peers:
        10.0.0.14(6)
```

For an explanation of the fields displayed in the output of the **show ip bgp route** command, refer to Displaying information for a specific route on page 145.

Specifying a maximum AS path length

You can use the **maxas-limit in** command to configure a device running BGP4 to discard routes that exceed a specified AS path limit. This limit can be configured globally, for peer groups, and for BGP neighbors.

When you configure **maxas-limit in** , the behavior of the device changes to first check the length of the AS paths in the UPDATE messages and then to apply the inbound policy. If the AS path exceeds the configured length, then the device performs the following actions:

- Does not store the route in the RIB and does not forward the NLRIs and attributes contained in the UPDATE message for that route
- Logs an error
- Processes the withdrawn NLRIs in the same update message

If a route from a peer exceeds the configured Maximum AS path limit, the device also removes the same route from that peer, if it exists, from its own RIB.

After a maximum AS path length is configured, the maximum AS path limit applies to all new inbound routes. To update previously stored routes, you must perform an inbound soft reset for all of the address families activated for that particular BGP neighbor session.

NOTE
If the neighbor soft-reconfiguration feature is enabled, you must perform a hard reset on the device to impose the maximum length limit.

NOTE
maxas-limit in is checked against the received AS_PATH and AS4_PATH attributes.

BGP devices check for and, if configured, apply **maxas-limit in** in the following order:

1. Neighbor value
2. Peer group value
3. Global value

   In a case where a neighbor has no maximum AS limit, a peer group has a value of 3 configured, and the system has a value of 9 configured, all of the devices in the peer group will only use the peer group value; the global value will never be used.

Setting a global maximum AS path limit

The syntax for the global maximum AS path limit command is:

Syntax: [no] maxas-limit in *num*

The **maxas-limit** keyword specifies the limit on the AS numbers in the as-path attribute. The **in** keyword allows the as-path attribute from any neighbor imposing a limit on AS numbers received. The default maximum length for the global system is 300. The range is 0 - 300. The **no** keyword removes the configuration at the global level.

NOTE
The device applies the BGP4 maximum AS path limit on a per virtual device basis.

To configure the global Maximum AS path limit to 15, enter the following command:

```
device(config-bgp)# maxas-limit in 15
```

Setting a maximum AS path limit for a peer group or neighbor

To set maximum AS path limit for a peer group or a neighbor, the syntax is:

Syntax: neighbor { *ip-addr* | *peer-group-name* } **maxas-limit in** [**num** | **disable**]

By default, neighbors or peer groups have no configured maximum values. The range is 0 - 300. The **disable** keyword is used to stop a neighbor from inheriting the configuration from the peer-group or global and to the use system default value.

To configure a peer group named "PeerGroup1" and set a maximum AS path value of 7, enter the following commands:

```
device(config-bgp)# neighbor PeerGroup1 peer-group
device(config-bgp)# neighbor PeerGroup1 maxas-limit in 7
```

BGP4 max-as error messages

This section lists error log messages that you might see when the device receives routes that exceed the configured AS segment limit or the internal memory limit. The log messages can contain a maximum of 30 ASNs. If a message contains more than 30 ASNs, the message is truncated and an ellipsis appears.

Maximum AS path limit error

```
SYSLOG: <11>Jan  1 00:00:00 mu1, BGP: From Peer 192.168.1.2 received Long AS_PAT H=
AS_CONFED_SET(4) 1 2 3 AS_CONFED_SEQUENCE(3) 4 AS_SET(1) 5 6 7 AS_SEQ(2) 8 9
attribute length (9) More than configured MAXAS-LIMIT 7
```

Memory limit error

```
SYSLOG: <11>Jan  1 00:00:00 mu1, BGP: From Peer 192.168.1.2 received Long AS_PAT H=
AS_CONFED_SET(4) 1 2 3 AS_CONFED_SEQUENCE(3) 4 AS_SET(1) 5 6 7 AS_SEQ(2) 8 9
attribute length (9) Exceeded internal memory limit
```

NOTE
The device generates a log message one time every two minutes. Because of this rate limit, it is possible that some errors might not appear in the log. In this case, you can use the **debug ip bgp events** command to view errors pertaining to the **maxas-limit** value and the actual AS path attributes received.

Originating the default route

By default, the device does not originate and advertise a default route using BGP4. A BGP4 default route is the IP address 0.0.0.0 and the route prefix 0 or network mask 0.0.0.0. For example, 0.0.0.0/0 is a default route.

NOTE
The device checks for the existence of an IGP route for 0.0.0.0/0 in the IP route table before creating a local BGP4 route for 0.0.0.0/0.

To configure the device to originate and advertise a default BGP4 route, enter this command.

```
device(config-bgp)# default-information-originate
```

Syntax: [no] default-information-originate

Changing the default metric used for route cost

By default, BGP4 uses the BGP MED value as the route cost when adding the route to the RTM. However, you can configure BGP4 to use the IGP cost instead.

NOTE
It is recommended that you change the default to IGP cost only in mixed-vendor environments, and that you change it on all Brocade devices in the environment.

To change the route cost default from BGP MED to IGP cost, enter a command such as the following:

```
device(config-bgp)# install-igp-cost
```

Syntax: [no] install-igp-cost

Use the **no** form of the command to revert to the default of BGP MED.

Configuring a static BGP4 network

This feature allows you to configure a static network in BGP4, creating a stable BGP4 network in the core. While a route configured with this feature will never flap unless it is manually deleted, a "static" BGP4 network will not interrupt the normal BGP4 decision process on other learned routes being installed into the RTM (Routing Table Manager). Consequently, when there is a route that can be resolved, it will be installed into the RTM.

To configure a static BGP4 network, enter commands such as the following.

```
device(config)# router bgp
device(config-bgp)# static-network 10.157.22.26/16
```

Syntax: [no] static-network *ipAddressPrefixImask*

The *ipAddress* and *mask* variables are the IPv4 address prefix and mask of the static BGP4 network you are creating.

Using the **no** option uninstalls a route (that was previously installed) from BGP4 RIB-IN and removes the corresponding drop route from the RTM. If there is a new best route, it is advertised to peers if necessary. Otherwise, a withdraw message is sent.

NOTE
The BGP4 network route and the BGP4 static network route are mutually exclusive. They cannot be configured with the same prefix and mask.

When you configure a route using the **static-network** command, BGP4 automatically generates a local route in BGP4 RIB-IN, and installs a NULL0 route in the RTM if there is no other valid route with the same prefix/mask learned from any peer. Otherwise, the learned BGP4 route will be installed in the RTM. In either situation, the new locally generated route will be the best route in RIB-IN and will be advertised to peers if it passes the per-peer outbound policies.

Setting an administrative distance for a static BGP4 network

When a static BGP4 network route is configured, its type is **local BGP4 route** and has a default administrative distance value of 200. To change the administrative distance value, change the value of all local BGP4 routes using the **distance** command at the router bgp level of the CLI, and set a new value for local routes. You can also assign a specific administrative distance value for each static network using the **distance** option as shown.

```
device(config)# router bgp
device(config-bgp)# static-network 10.157.22.26/16 distance 100
```

Syntax: [no] static-network *ipAddressPrefixImask* **distance** *distance-value*

The *ipAddress* and *mask* variables are the IPv4 address prefix and mask of the static BGP4 network for which you are setting an administrative distance.

The *distance-value* sets the administrative distance of the static BGP4 network route. The range for this value is 1 - 255.

Limiting advertisement of a static BGP4 network to selected neighbors

You can control the advertisement of a static BGP4 network to BGP4 neighbors that are configured as Service Edge Devices. When this feature is configured for a BGP4 neighbor, static BGP4 network routes that are installed in the routing table as DROP routes are not advertised to that neighbor. When this feature is configured, the route is only advertised to identified Service Edge devices if it is installed as a forward route, such as the routes described in these steps.

1. There is a learned route from a customer BGP4 peering.
2. There is a valid learned route from another Services Edge device as a result of a customer route present on that device.

   To configure a BGP4 neighbor to limit the advertisement of Static BGP4 Network routes, enter the **static-network-edge** command as shown.

   ```
   device(config)# router bgp
   device(config-bgp)# neighbor 10.2.3.4 static-network-edge
   ```

   Syntax: [no] neighbor *ip-address* | *peer-group-name* **static-network-edge**

The *ip-addr* and *peer-group-name* variables indicate whether you are configuring an individual neighbor or a peer group. If you specify a neighbor IP address, you are configuring that individual neighbor. If you specify a peer group name, you are configuring a peer group.

Route-map continue clauses for BGP4 routes

A continuation clause in a route-map directs program flow to skip over route-map instances to another, user-specified instance. If a matched instance contains a continue clause, the system looks for the instance that is identified in the continue clause.

The continue clause in a matching instance initiates another traversal at the instance that you specify in the continue clause. The system records all of the matched instances and, if no deny statements are encountered, proceeds to execute the set clauses of the matched instances.

If the system scans all route map instances but finds no matches, or if a deny condition is encountered, then it does not update the routes. Whenever a matched instance contains a deny parameter, the current traversal terminates, and none of the updates specified in the set clauses of the matched instances in both current and previous traversals are applied to the routes.

This feature supports a more programmable route map configuration and route filtering scheme for BGP4 peering. It can also execute additional instances in a route map after an instance is executed with successful match clauses. You can configure and organize more modular policy definitions to reduce the number of instances that are repeated within the same route map.

This feature currently applies to BGP4 routes only. For protocols other than BGP4, continue statements are ignored.

Specifying route-map continuation clauses

This section describes the configuration of route-map continuation clauses. The following sequence of steps (with referenced items in the screen output in bold) is described:

- The configuration context for a route-map named *test* is entered.
- Two route-map **continue** statements are added to route-map *test* .
- The **show route-map** output displays the modified route-map *test* .
- Subsequent **neighbor** commands identify the route map *test* in the inbound and outbound directions for the neighbor at 10.8.8.3.
- The **show ip bgp config** output shows inbound and outbound route-map *test* for the neighbor at 10.8.8.3.

```
device(config-bgp)# route-map test permit 1
device(config-routemap test)# match metric 10
device(config-routemap test)# set weight 10
device(config-routemap test)# continue 2
device(config-routemap test)# route-map test permit 2
device(config-routemap test)# match tag 10
device(config-routemap test)# set weight 20
device(config-routemap test)# continue 3
device(config-routemap test)# router bgp
device(config-bgp)# exit
device(config-bgp)# show route-map test
route-map test permit 1
 match metric 10
 set weight 10
 continue 2
route-map test permit 2
 match tag 10
 set weight 20
 continue 3
device(config-bgp)# neighbor 10.8.8.3 route-map in test
device(config-bgp)# neighbor 10.8.8.3 route-map out test
device(config-bgp)# show ip bgp config
```

```
Current BGP configuration:
router bgp
 local-as 100
 neighbor 10.8.8.3 remote-as 200
 address-family ipv4 unicast
 neighbor 10.8.8.3 route-map in test
 neighbor 10.8.8.3 route-map out test
 exit-address-family
 address-family ipv4 multicast
 exit-address-family
 address-family ipv6 unicast
 exit-address-family
 address-family ipv6 multicast
 exit-address-family
 address-family vpnv4 unicast
 exit-address-family
end of BGP configuration
```

Syntax: [no] route-map *map-name* **permit** | **deny** *num*

The **no** form of the command deletes the route map. The map-name is a string of up to 80 characters that specifies the map.

The **permit** option means the device applies match and set clauses associated with this route map instance.

The **deny** option means that any match causes the device to ignore the route map.

The *num* parameter specifies the instance of the route map defined in the route-map context that the CLI enters. Routes are compared to the instances in ascending numerical order. For example, a route is compared to instance 1, then instance 2, and so on.

Syntax: [no] continue [*instance-number*]

The **continue** command is entered in the context of a route-map instance. The **no** form of the command deletes the continue clause specified by *instance-number* . The instance number range is 0 - 4294967295, and the occurrences of *instance-number* must be in ascending numeric order. If you specify a continue clause without an instance number, it means "continue to the next route-map instance."

Syntax: [no] neighbor *ip-addr* | *peer-group-name* [**route-map in** | **out** *map-name*]

This syntax shows only the **neighbor** parameters that apply to this example. The *ip-addr* or *peer-group-name* identifies the neighbor, and the **route-map in** and **out***map-name* options let you specify a route map and direction to apply to the neighbor.

Dynamic route filter update

Routing protocols use various route filters to control the distribution of routes. Route filters are used to filter routes received from and advertised to other devices. Protocols also use route-map policies to control route redistribution from other routing protocols. In addition, route filter policies are used to select routes to be installed in the routing tables, and used by forwarding engine to forward traffic.

There are currently 5 different types of route filters defined for use in a device:

- Access List (ACL)
- Prefix-List
- BGP4 as-path Access-list
- BGP4 community-list
- BGP4 extended community-list
- Route-map

Not every protocol uses all of these route filters. A protocol will usually use two or three filter types.

TABLE 5 Route filters used by each protocol

Protocol	Route map	Prefix list	Community-list	Extended community-list	As-path access- list	ACL
BGP4	X	X	BGP4 does not use Community-List filters directly. It does use them indirectly through route-map filters that contain Community-List filters.	BGP4 does not directly use Extended Community-List filters, but indirectly uses them through route-map filters that contain Extended Community-List filters.	X	X
OSPF	X	X	X	X	X	X
RIP	X	X	X	X	X	
IS-IS	X	X	X	X	X	
RIPng		X				
OSPFv3	X	X	X		X	
MSDP	X					
MCast						X

When a route filter is changed (created, modified or deleted) by a user, the filter change notification will be sent to all relevant protocols, so that protocols can take appropriate actions. For example if BGP4 is using a route-map (say MapX) to control the routes advertised to a particular peer, the change of route-map (MapX) will cause BGP4 to re-evaluate the advertised routes, and make the appropriate advertisements or withdrawals according to the new route-map policy.

A route filter change action can happen in three ways.

1. A new filter is defined (created).

   This filter name may be already referenced by an application. The application needs to be notified of the addition of the new filter, and will bind to and use the new filter. In general, if a filter name is referenced by an application, but is not actually defined, the application assumes the default **deny** action for the filter.

2. An existing filter is undefined (removed).

   If the deleted filter is already used and referenced by an application, the application will unbind itself from the deleted filter.

3. An existing filter is modified (updated).

   If the filter is already used and referenced by an application, the application will be notified.

Protocols are automatically notified when a route filter is created, deleted or modified. In addition, when a protocol is notified of a filter change, appropriate steps are taken to apply the new or updated filter to existing routes.

Commands for dynamic route filter updating

In order to allow multiple filter updates to be processed together by applications, the device waits 10 seconds by default before notifying applications of the filter change. You can force an immediate update notification or modify the time delay from when a change is made to a route filter to when the protocols are notified.

Route filter update delay settings can be configured using the commands shown here.

Setting a time delay for route filter update notification

Set the amount of time that the device waits before sending filter addition, deletion and modification notification to protocols using the following command.

```
device(config)# filter-change-update-delay 100
```

Syntax: [no] filter-change-update-delay *delay-time*

The *delay-time* variable specifies the amount of time in seconds that the device waits before sending filter addition, deletion and modification notification to protocols. The valid range is 0 to 600 seconds. If you set the value to 0, filter change notifications will not be automatically sent to protocols. The default value is 10 seconds.

NOTE
The **filter-change-update-delay** command also affects a route map that is being used in a PBR policy.

Performing an immediate route filter update

To force an immediate filter update to the relevant protocols, use the following command.

```
device(config)# clear filter-change-update
```

Syntax: clear filter-change-update

This command forces an immediate filter update regardless of the filter-change-update-delay setting. It can also be used to simultaneously submit multiple change notifications when the filter-change-update-delay is set to 0. When changes are complete, run the **clear filter-change-update** command to update protocols.

NOTE
There may be delays in sending route filter change notifications to applications, and delays in applying the new or updated filter to all existing routes retroactively. However any *new* routes or changes to existing routes will be subject to the new filters.

Filter update delay and BGP

The **filter-changes-update-delay** command applies (remove only) to changes of filters that are already used or referenced by applications. If the content of a filter is changed, the new filter action

takes effect after **filter-changes-update-delay** for existing routes. The notification delay also applies to situations where the usage or reference of a filter is changed in BGP.

For example, the following BGP neighbor command sets or changes the route-map filter on a neighbor:

```
device(config-bgp)# neighbor x.x.x.x route-map map_abc out
```

In this case, the device applies the route-map "map_abc" to the peer, and updates the neighbor out-bound routes after a delay.

If the *delay-time* is 0, BGP does not start peer out-bound policy updates immediately.

Use the **clear filter-change-update** or **clear ip bgp neighbor soft-out** commands to trigger BGP policy updates.

Similarly, the **filter-changes-update-delay** command also applies to the neighbor in-bound policy change.

NOTE
The auto-update action for a BGP peer filter is newly introduced in release 5.2. In previous releases, a user needs to manually issue the **clear ip bgp neighbor soft out** command to cause the device to apply the new route-map retroactively to existing routes.

The general guideline is to define a policy *first* , then apply it to a BGP peer.

BGP4 policy processing order

The order of application of policies when processing inbound and outbound route advertisements on the device is:

1. Ip prefix-list
2. Outbound Ip prefix-list ORF, if negotiated
3. Outbound extended-community ORF, if negotiated
4. Filter-list (using As-path access-list)
5. Distribute list (using IP ACL - IPv4 unicast only)
6. Route-map

Generalized TTL Security Mechanism support

The device supports the Generalized TTL Security Mechanism (GTSM) as defined in RFC 3682. GTSM protects the device from attacks of invalid BGP4 control traffic that is sent to overload the CPU or hijack the BGP4 session. GTSM protection applies to EBGP neighbors only.

When GTSM protection is enabled, BGP4 control packets sent by the device to a neighbor have a Time To Live (TTL) value of 255. In addition, the device expects the BGP4 control packets received from the neighbor to have a TTL value of either 254 or 255. For multihop peers (where the **ebgp-multihop** option is configured for the neighbor), the device expects the TTL for BGP4 control packets received from the neighbor to be greater than or equal to 255, minus the configured number of hops to the neighbor. If the BGP4 control packets received from the neighbor do not have the anticipated value, the device drops them.

For more information on GTSM protection, refer to RFC 3682.

To enable GTSM protection for neighbor 192.168.9.210 (for example), enter the following command.

```
device(config-bgp-router)# neighbor 192.168.9.210 ebgp-btsh
```

Syntax: [no] neighbor *ip-addr* | *peer-group-name* **ebgp-btsh**

NOTE
For GTSM protection to work properly, it must be enabled on both the device and the neighbor.

show metro mp-vlp-queue

Displays priority information about management processor virtual line card (MP-VLP) queues on Brocade NetIron CER Series devices.

Syntax **show metro mp-vlp-queue**

Modes Privileged EXEC mode

Usage Guidelines Use this command to view statistics about messages from the MP are that are queued in the VLP to dequeue.

NOTE
If the Dequeue Time is less than 1 millisecond, it is not recorded in the **show metro mp-vlp-queue** statistics. The corresponding timestamp is also not recorded. The initial timestamp is shown as "0000.00.00-00:00:00.000".

Command Output The **show metro mp-vlp-queue** command displays the following information:

Output field	Description
MP => VLP Queue	The queue priority: high, medium, or low.
Queue Size	The maximum amount of packet counts that the queue can handle at a given time.
Total Pkt Count	The total count of messages queued in each queue.
Current Pkt Count	The count of messages queued at a specific moment in each queue.
Pkt High WM	The maximum messages reached in the queue at any point of time.
Pkt drop Count	The amount of messages that were dropped because the queue was full.
Dequeue High WM(msec)	The longest period of time, in milliseconds, that a message remained in that queue.
Timestamp Pkt High WM(High)	The timestamp for the time when the high water mark for the number of messages in the high priority queue is reached.
Timestamp Pkt High WM(Medium)	The timestamp for the time when the high water mark for the number of messages in the medium priority queue is reached.
Timestamp Pkt High WM(Low)	The timestamp for the time when the high water mark for the number of messages in the low priority queue is reached.
Timestamp Dequeue Time HWM(High)	The timestamp for the time when the most delay is observed in the high priority queue.
Timestamp Dequeue Time HWM(Medium)	The timestamp for the time when the most delay is observed in the medium priority queue.

Output field	Description
Timestamp Dequeue Time HWM(Low)	The timestamp for the time when the most delay is observed in the low priority queue.

Examples

This example shows sample output from the **show metro mp-vlp-queue** command. Three MP-VLP queues are shown with priority High, Medium and Low. The messages from the MP are queued in these queues for the VLP to dequeue.

```
LP-1# show metro mp-vlp-queue

MP => VLP Queue       :         High       Medium          Low
Queue Size            :         2000         2000         2000
Total Pkt Count       :      2160279            0     61210672
Current Pkt Count     :            0            0            0
Pkt High WM           :           13            0         1992
Pkt drop count        :            0            0            0
Dequeue Time HWM(msec):        12000            0        12675

Timestamp Pkt High WM(High)       : [          13]: 2015.02.25-08:07:16.533
Timestamp Pkt High WM(Medium)     : [           0]: 0000.00.00-00:00:00.000
Timestamp Pkt High WM(Low)        : [        1992]: 2015.02.25-08:07:17.223

Timestamp Dequeue Time HWM(High)  : [       12000]: 2015.02.25-08:07:17.230
Timestamp Dequeue Time HWM(Medium): [           0]: 0000.00.00-00:00:00.000
Timestamp Dequeue Time HWM(Low)   : [       12675]: 2015.02.25-08:07:17.800
```

This example shows sample output from the **show metro mp-vlp-queue** command after statistics have been cleared using the **clear metro mp-vlp-queue** command.

```
LP-1# show metro mp-vlp-queue

MP => VLP Queue       :         High       Medium          Low
Queue Size            :         2000         2000         2000
Total Pkt Count       :            0            0            0
Current Pkt Count     :            0            0            0
Pkt High WM           :            0            0            0
Pkt drop count        :            0            0            0
Dequeue Time HWM(msec):            0            0            0

Timestamp Pkt High WM(High)       : [           0]: 0000.00.00-00:00:00.000
Timestamp Pkt High WM(Medium)     : [           0]: 0000.00.00-00:00:00.000
Timestamp Pkt High WM(Low)        : [           0]: 0000.00.00-00:00:00.000

Timestamp Dequeue Time HWM(High)  : [           0]: 0000.00.00-00:00:00.000
Timestamp Dequeue Time HWM(Medium): [           0]: 0000.00.00-00:00:00.000
Timestamp Dequeue Time HWM(Low)   : [           0]: 0000.00.00-00:00:00.000
```

History

Release version	Command history
5.8.00a	This command was introduced.

clear metro mp-vlp-queue

Resets the management processor virtual line card (MP-VLP) queue statistics on Brocade NetIron CER Series devices.

Syntax **clear metro mp-vlp-queue**

Modes Privileged EXEC mode

Usage Guidelines

Examples This example clears all the counters in the MP-VLP queue statistics.

```
LP-1# clear metro mp-vlp-queue
```

History

Release version	Command history
5.8.00a	This command was introduced.

Displaying BGP4 information

You can display the following configuration information and statistics for BGP4 protocol:

- Summary BGP4 configuration information for the device
- Active BGP4 configuration information (the BGP4 information in the running configuration)
- Neighbor information
- Peer-group information
- Information about the paths from which BGP4 selects routes
- Summary BGP4 route information
- Virtual Routing and Forwarding (VRF) instance information
- The device's BGP4 route table
- Route flap dampening statistics
- Active route maps (the route map configuration information in the running configuration)
- BGP4 graceful restart neighbor Information
- AS4 support and asdot notation

Displaying summary BGP4 information

You can display the local AS number, the maximum number of routes and neighbors supported, and some BGP4 statistics. You can also display BGP4 memory usage for:

- BGP4 routes installed
- Routes advertising to all neighbors (aggregated into peer groups)
- Attribute entries installed

The **show ip bgp summary** command output has the following limitations:

- If a BGP4 peer is not configured for an address-family, the peer information is not displayed.
- If a BGP4 peer is configured for an address-family but not negotiated for an address-family after the BGP4 peer is in the established state, the **show ip bgp summary** command output shows (**NoNeg**) at the end of the line for this peer.
- If a BGP4 peer is configured and negotiated for that address-family, its display is the same as in previous releases.

To view summary BGP4 information for the device, enter the following command at any CLI prompt

```
device# show ip bgp summary
  BGP4 Summary
  Router ID: 10.10.1.14 Local AS Number: 100
  Confederation Identifier: not configured
  Confederation Peers:
  Maximum Number of IP ECMP Paths Supported for Load Sharing: 1
  Number of Neighbors Configured: 67, UP: 67
  Number of Routes Installed: 258088, Uses 22195568 bytes
  Number of Routes Advertising to All Neighbors:17,035844(3,099146 entries),
Uses 192,147052 bytes
  Number of Attribute Entries Installed: 612223, Uses 55100070 bytes
  Neighbor Address  AS#   State   Time    Rt:Accepted Filtered Sent ToSend
  10.0.100.2       100   ESTABp  0h28m24s   0        0        258087 0
  10.0.101.2       100   ESTAB   0h28m24s   0        0        258087 0
  10.2.3.4         200   ADMDN   0h44m56s   0        0        0      2
```

Syntax: show ip bgp summary

TABLE 6 show ip bgp summary output descriptions

This field	Displays
Router ID	The device ID.
Local AS Number	The BGP4 AS number for the device.
Confederation Identifier	The AS number of the confederation in which the device resides.
Confederation Peers	The numbers of the local autonomous systems contained in the confederation. This list matches the confederation peer list you configure on the device.
Maximum Number of Paths Supported for Load Sharing	The maximum number of route paths across which the device can balance traffic to the same destination. The feature is enabled by default but the default number of paths is 1. You can increase the number from 2 through 8 paths.
Number of Neighbors Configured	The number of BGP4 neighbors configured on this device, and currently in established state.
Number of Routes Installed	The number of BGP4 routes in the device BGP4 route table and the route or path memory usage.
Number of Routes Advertising to All Neighbors	The total of the RtSent and RtToSend columns for all neighbors, the total number of unique ribout group entries, and the amount of memory used by these groups.
Number of Attribute Entries Installed	The number of BGP4 route-attribute entries in the device route-attributes table and the amount of memory used by these entries.
Neighbor Address	The IP addresses of the BGP4 neighbors for this device.

TABLE 6 show ip bgp summary output descriptions (Continued)

This field	Displays
AS#	The AS number.
State	The state of device sessions with each neighbor. The states are from this perspective of the device, not the neighbor. State values are based on the BGP4 state machine values described in RFC 1771 and can be one of the following for each device: • IDLE - The BGP4 process is waiting to be started. Usually, enabling BGP4 or establishing a neighbor session starts the BGP4 process. A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • ADMND - The neighbor has been administratively shut down. • CONNECT - BGP4 is waiting for the connection process for the TCP neighbor session to be completed. • ACTIVE - BGP4 is waiting for a TCP connection from the neighbor. **Note** : If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection. • OPEN SENT - BGP4 is waiting for an Open message from the neighbor. • OPEN CONFIRM - BGP4 has received an Open message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. If the device receives a KEEPALIVE message from the neighbor, the state changes to Established. If the message is a NOTIFICATION, the state changes to Idle. • ESTABLISHED - BGP4 is ready to exchange UPDATE packets with the neighbor. **Operational States:** Additional information regarding the operational states of the BGP4 states described above may be added as described in the following: • **(+)** - is displayed if there is more BGP4 data in the TCP receiver queue. **Note** : If you display information for the neighbor using the **show ip bgp neighbor***ip-addr* command, the TCP receiver queue value will be greater than 0. • **(-)** - indicates that the session has gone down and the software is clearing or removing routes. • **(*)** - indicates that the inbound or outbound policy is being updated for the peer. • **(s)** - indicates that the peer has negotiated restart, and the session is in a stale state. • **(r)** - indicates that the peer is restarting the BGP4 connection, through restart. • **(^)** - on the standby MP indicates that the peer is in the ESTABLISHED state and has received restart capability (in the primary MP). • **(<)** - indicates that the device is waiting to receive the "End of RIB" message the peer. • (p**)** - indicates that the neighbor ribout group membership change is pending or in progress
Time	The time that has passed since the state last changed.
Accepted	The number of routes received from the neighbor that this device installed in the BGP4 route table. Usually, this number is lower than the RoutesRcvd number. The difference indicates that this device filtered out some of the routes received in the UPDATE messages.

TABLE 6 show ip bgp summary output descriptions (Continued)

This field	Displays
Filtered	The routes or prefixes that have been filtered out: • If soft reconfiguration is enabled, this field shows how many routes were filtered out (not placed in the BGP4 route table) but retained in memory. • If soft reconfiguration is not enabled, this field shows the number of BGP4 routes that have been filtered out.
Sent	The number of BGP4 routes the device has sent to the neighbor.
ToSend	The number of routes the device has queued to advertise and withdraw to a neighbor.

Displaying the active BGP4 configuration

To view the active BGP4 configuration information contained in the running configuration without displaying the entire running configuration, enter the following command at any level of the CLI.

```
device# show ip bgp config
router bgp
  local-as 200
neighbor 10.102.1.1 remote-as 200
neighbor 10.102.1.1 ebgp-multihop
neighbor 10.102.1.1 update-source loopback 1
neighbor 192.168.2.1 remote-as 100
neighbor 10.200.2.2 remote-as 400
neighbor 2001:db8::1:1 remote-as 200
neighbor 2001:db8::1:2 remote-as 400
neighbor 2001:db8::1 remote-as 300

address-family ipv4 unicast
no neighbor 2001:db8::1:1 activate
no neighbor 2001:db8::1:2 activate
no neighbor 2001:db8::1 activate
exit-address-family

address-family ipv4 multicast
exit-address-family

address-family ipv6 unicast
redistribute static
neighbor 2001:db8::1:1 activate
neighbor 2001:db8::1:2 activate
neighbor 2001:db8::1 activate
exit-address-family
end of BGP configuration
```

Syntax: show ip bgp config

Displaying summary neighbor information

The **show ip bgp neighbor** command output has the following limitations.

1. If BGP4 peer is not configured for an address-family, the peer information will NOT be displayed.
2. If BGP4 peer is configured for an address-family, it will display the same as in previous releases.

   To display summary neighbor information, enter a command such as the following at any level of the CLI.

```
device# show ip bgp neighbor 192.168.4.211 routes-summary
```

```
1   IP Address: 192.168.4.211
Routes Accepted/Installed:1,  Filtered/Kept:11,  Filtered:11
   Routes Selected as BEST Routes:1
      BEST Routes not Installed in IP Forwarding Table:0
   Unreachable Routes (no IGP Route for NEXTHOP):0
   History Routes:0

NLRIs Received in Update Message:24,  Withdraws:0 (0),  Replacements:1
   NLRIs Discarded due to
      Maximum Prefix Limit:0, AS Loop:0
      Invalid Nexthop:0, Invalid Nexthop Address:0.0.0.0
      Duplicated Originator_ID:0, Cluster_ID:0

Routes Advertised:0,  To be Sent:0,  To be Withdrawn:0
NLRIs Sent in Update Message:0,  Withdraws:0,  Replacements:0

Peer Out of Memory Count for:
   Receiving Update Messages:0, Accepting Routes(NLRI):0
   Attributes:0, Outbound Routes(RIB-out):0
```

Syntax: show ip bgp neighbors [*ip-addr*] | [**route-summary**]

TABLE 7 show ip bgp neighbors route-summary output descriptions

This field	Displays
IP Address	The IP address of the neighbor.
Routes Received	How many routes the device has received from the neighbor during the current BGP4 session: • Accepted or Installed - Number of received routes the device accepted and installed in the BGP4 route table. • Filtered or Kept - Number of routes that were filtered out, but were retained in memory for use by the soft reconfiguration feature. • Filtered - Number of received routes filtered out.
Routes Selected as BEST Routes	The number of routes that the device selected as the best routes to their destinations.
BEST Routes not Installed in IP Forwarding Table	The number of routes received from the neighbor that are the best BGP4 routes to their destinations, but were not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes).
Unreachable Routes	The number of routes received from the neighbor that are unreachable because the device does not have a valid RIP, OSPF, or static route to the next-hop.
History Routes	The number of routes that are down but are being retained for route flap dampening purposes.
NLRIs Received in Update Message	The number of routes received in Network Layer Reachability (NLRI) format in UPDATE messages: • Withdraws - Number of withdrawn routes the device has received. • Replacements - Number of replacement routes the device has received.

TABLE 7 show ip bgp neighbors route-summary output descriptions (Continued)

This field	Displays
NLRIs Discarded due to	Indicates the number of times the device discarded an NLRI for the neighbor due to the following reasons: • Maximum Prefix Limit - The configured maximum prefix amount had been reached. • AS Loop - An AS loop occurred. An AS loop occurs when the BGP4 AS-path attribute contains the local AS number. • maxas-limit aspath - The number of route entries discarded because the AS path exceeded the configured maximum length or exceeded the internal memory limits. • Invalid Nexthop - The next-hop value was not acceptable. • Duplicated Originator_ID - The originator ID was the same as the local device ID. • Cluster_ID - The cluster list contained the local cluster ID, or the local device ID if the cluster ID is not configured.
Routes Advertised	The number of routes the device has advertised to this neighbor: • To be Sent - The number of routes queued to send to this neighbor. • To be Withdrawn - The number of NLRIs for withdrawing routes the device has queued to send to this neighbor in UPDATE messages.
NLRIs Sent in Update Message	The number of NLRIs for new routes the device has sent to this neighbor in UPDATE messages: • Withdraws - Number of routes the device has sent to the neighbor to withdraw. • Replacements - Number of routes the device has sent to the neighbor to replace routes the neighbor already has.
Peer Out of Memory Count for	Statistics for the times the device has run out of BGP4 memory for the neighbor during the current BGP4 session: • Receiving Update Messages - The number of times UPDATE messages were discarded because there was no memory for attribute entries. • Accepting Routes (NLRI) - The number of NLRIs discarded because there was no memory for NLRI entries. This count is not included in the Receiving Update Messages count. • Attributes - The number of times there was no memory for BGP4 attribute entries. • Outbound Routes (RIB-out) - The number of times there was no memory to place a "best" route into the neighbor route information base (Adj-RIB-Out) for routes to be advertised.

Displaying BGP4 neighbor information

You can display configuration information and statistics for BGP4 neighbors of the device.

To view BGP4 neighbor information, including the values for all the configured parameters, enter the following command.

NOTE
The display shows all the configured parameters for the neighbor. Only the parameters that have values different from their defaults are shown.

```
device(config-bgp)# show ip bgp neighbor 10.4.0.2
    Total number of BGP neighbors:
1   IP Address: 10.4.0.2, AS: 5 (EBGP), RouterID: 10.0.0.1
       Description: neighbor 10.4.0.2
    Local AS: 101
```

```
State: ESTABLISHED, Time: 0h1m0s, KeepAliveTime: 0, HoldTime: 0
   PeerGroup: pg1
   Multihop-EBGP: yes, ttl: 1
   RouteReflectorClient: yes
   SendCommunity: yes
   NextHopSelf: yes
   DefaultOriginate: yes (default sent)
   MaximumPrefixLimit: 90000
   RemovePrivateAs: : yes
   RefreshCapability: Received
Route Filter Policies:
   Distribute-list: (out) 20
   Filter-list: (in) 30
   Prefix-list: (in) pf1
   Route-map: (in) setnp1  (out) setnp2
Messages:    Open    Update  KeepAlive Notification Refresh-Req
   Sent    : 1       1       1         0            0
   Received: 1       8       1         0            0
Last Update Time: NLRI       Withdraw          NLRI       Withdraw
              Tx: 0h0m59s    ---           Rx: 0h0m59s    ---
Last Connection Reset Reason:Unknown
  Notification Sent:     Unspecified
  Notification Received: Unspecified
TCP Connection state: ESTABLISHED
   Local host:  10.4.0.1, Local  Port: 179
   Remote host: 10.4.0.2, Remote Port: 8053
   ISentSeq:   52837276  SendNext:   52837392  TotUnAck:          0
   TotSent:         116  ReTrans:          0  UnAckSeq:   52837392
   IRcvSeq:  2155052043  RcvNext:  2155052536  SendWnd:       16384
   TotalRcv:        493  DupliRcv:         0  RcvWnd:        16384
   SendQue:           0  RcvQue:           0  CngstWnd:       1460
```

This example shows how to display information for a specific neighbor, by specifying the neighbor's IP address with the command. Since none of the other display options are used, all of the information is displayed for the neighbor. The number in the far left column indicates the neighbor for which information is displayed. When you list information for multiple neighbors, this number makes the display easier to read.

The TCP statistics at the end of the display show status for the TCP session with the neighbor. Most of the fields show information stored in the Transmission Control Block (TCB) for the TCP session between the device and the neighbor. These fields are described in detail in section 3.2 of RFC 793, "Transmission Control Protocol Functional Specification".

Syntax: show ip bgp neighbors [*ip-addr* [**advertised-routes** [**detail** [*ip-add* [*/ mask-bits*]]]] | [**attribute-entries** [**detail**]] | [**flap-statistics**] | [**last-packet-with-error**] | [**received prefix-filter**] | [**received-routes**] | [**routes** [**best**] | [**detail** [**best**] | [**not-installed-best**] | [**unreachable**]] | [**rib-out-routes** [*ip-addrlmask-bits* | *ip-addr net-mask* | **detail**]] | [**routes-summary**]]

The *ip-addr* option lets you narrow the scope of the command to a specific neighbor.

The **advertised-routes** option displays only the routes that the device has advertised to the neighbor during the current BGP4 session.

The **attribute-entries** option shows the attribute-entries associated with routes received from the neighbor.

The **flap-statistics** option shows the route flap statistics for routes received from or sent to the neighbor.

The **last-packet-with-error** option displays the last packet from the neighbor that contained an error. The packet contents are displayed in decoded (human-readable) format.

The **received prefix-filter** option shows the Outbound Route Filters (ORFs) received from the neighbor. This option applies to cooperative route filtering.

The **received-routes** option lists all the route information received in route updates from the neighbor since the soft reconfiguration feature was enabled.

The **routes** option lists the routes received in UPDATE messages from the neighbor. You can specify the following additional options:

- **best** - Displays the routes received from the neighbor that the device selected as the best routes to their destinations.
- **not-installed-best** - Displays the routes received from the neighbor that are the best BGP4 routes to their destinations, but were not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes).
- **unreachable** - Displays the routes that are unreachable because the device does not have a valid RIP, OSPF, or static route to the next hop.
- **detail** - Displays detailed information for the specified routes. You can refine your information request by also specifying one of the options (**best** , **not-installed-best** , or **unreachable**).

The **rib-out-routes** option lists the route information base (RIB) for outbound routes. You can display all routes or specify a network address.

The **routes-summary** option displays a summary of the following information:

- Number of routes received from the neighbor
- Number of routes accepted by this device from the neighbor
- Number of routes this device filtered out of the UPDATES received from the neighbor and did not accept
- Number of routes advertised to the neighbor
- Number of attribute entries associated with routes received from or advertised to the neighbor.

TABLE 8 show ip bgp neighbor output descriptions

Field	Information displayed
Total Number of BGP4 Neighbors	The number of BGP4 neighbors configured.
IP Address	The IP address of the neighbor.
AS	The AS the neighbor is in.
EBGP or IBGP	Whether the neighbor session is an IBGP session, an EBGP session, or a confederation EBGP session: • EBGP - The neighbor is in another AS. • EBGP_Confed - The neighbor is a member of another sub-AS in the same confederation. • IBGP - The neighbor is in the same AS.
RouterID	The neighbor device ID.
Description	The description you gave the neighbor when you configured it on the device.
Local AS	The value (if any) of the Local AS configured.

TABLE 8 show ip bgp neighbor output descriptions (Continued)

Field	Information displayed
State	The state of the session with the neighbor. The states are from the device perspective, not the neighbor perspective. The state values are based on the BGP4 state machine values described in RFC 1771 and can be one of the following for each device: • IDLE - The BGP4 process is waiting to be started. Usually, enabling BGP4 or establishing a neighbor session starts the BGP4 process. A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • ADMND - The neighbor has been administratively shut down. Refer to Administratively shutting down a session with a BGP4 neighbor on page 55. A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • CONNECT - BGP4 is waiting for the connection process for the TCP neighbor session to be completed. • ACTIVE - BGP4 is waiting for a TCP connection from the neighbor. **NOTE** If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection. • OPEN SENT - BGP4 is waiting for an Open message from the neighbor. • OPEN CONFIRM - BGP4 has received an OPEN message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. If the device receives a KEEPALIVE message from the neighbor, the state changes to Established. If the message is a NOTIFICATION, the state changes to Idle. • ESTABLISHED - BGP4 is ready to exchange UPDATE messages with the neighbor. If there is more BGP4 data in the TCP receiver queue, a plus sign (+) is also displayed. **NOTE** If you display information for the neighbor using the **show ip bgp neighbor***ip-addr* command, the TCP receiver queue value will be greater than 0.
Time	The amount of time this session has been in the current state.
KeepAliveTime	The keep alive time, which specifies how often this device sends keepalive messages to the neighbor.
HoldTime	The hold time, which specifies how many seconds the device will wait for a keepalive or update message from a BGP4 neighbor before deciding that the neighbor is not operational.
PeerGroup	The name of the peer group the neighbor is in, if applicable.

TABLE 8 show ip bgp neighbor output descriptions (Continued)

Field	Information displayed
Multihop-EBGP	Whether this option is enabled for the neighbor.
RouteReflectorClient	Whether this option is enabled for the neighbor.
SendCommunity	Whether this option is enabled for the neighbor.
NextHopSelf	Whether this option is enabled for the neighbor.
DefaultOriginate	Whether this option is enabled for the neighbor.
MaximumPrefixLimit	Maximum number of prefixes the device will accept from this neighbor.
RemovePrivateAs	Whether this option is enabled for the neighbor.
RefreshCapability	Whether this device has received confirmation from the neighbor that the neighbor supports the dynamic refresh capability.
CooperativeFilteringCapability	Whether the neighbor is enabled for cooperative route filtering.
Distribute-list	Lists the distribute list parameters, if configured.
Filter-list	Lists the filter list parameters, if configured.
Prefix-list	Lists the prefix list parameters, if configured.
Route-map	Lists the route map parameters, if configured.
Messages Sent	The number of messages this device has sent to the neighbor. The display shows statistics for the following message types: • Open • Update • KeepAlive • Notification • Refresh-Req
Messages Received	The number of messages this device has received from the neighbor. The message types are the same as for the Message Sent field.
Last Update Time	Lists the last time updates were sent and received for the following: • NLRIs • Withdraws

TABLE 8 show ip bgp neighbor output descriptions (Continued)

Field	Information displayed
Last Connection Reset Reason	The reason the previous session with this neighbor ended. The reason can be one of the following: Reasons described in the BGP4 specifications: • Message Header Error • Connection Not Synchronized • Bad Message Length • Bad Message Type • OPEN Message Error • Unsupported Version Number • Bad Peer AS Number • Bad BGP4 Identifier • Unsupported Optional Parameter • Authentication Failure • Unacceptable Hold Time • Unsupported Capability • UPDATE Message Error • Malformed Attribute List • Unrecognized Well-known Attribute • Missing Well-known Attribute • Attribute Flags Error • Attribute Length Error • Invalid ORIGIN Attribute • Invalid NEXT_HOP Attribute • Optional Attribute Error • Invalid Network Field • Malformed AS_PATH • Hold Timer Expired • Finite State Machine Error • Rcv Notification
Last Connection Reset Reason (cont.)	Reasons specific to the Brocade implementation: • Reset All Peer Sessions • User Reset Peer Session • Port State Down • Peer Removed • Peer Shutdown • Peer AS Number Change • Peer AS Confederation Change • TCP Connection KeepAlive Timeout • TCP Connection Closed by Remote • TCP Data Stream Error Detected

TABLE 8 show ip bgp neighbor output descriptions (Continued)

Field	Information displayed
Notification Sent	If the device receives a NOTIFICATION message from the neighbor, the message contains an error code corresponding to one of the following errors. Some errors have subcodes that clarify the reason for the error. Where applicable, the subcode messages are listed underneath the error code messages. • Message Header Error: - Connection Not Synchronized - Bad Message Length - Bad Message Type - Unspecified • Open Message Error: - Unsupported Version - Bad Peer As - Bad BGP4 Identifier - Unsupported Optional Parameter - Authentication Failure - Unacceptable Hold Time - Unspecified • Update Message Error: - Malformed Attribute List - Unrecognized Attribute - Missing Attribute - Attribute Flag Error - Attribute Length Error - Invalid Origin Attribute - Invalid NextHop Attribute - Optional Attribute Error - Invalid Network Field - Malformed AS Path - Unspecified • Hold Timer Expired • Finite State Machine Error • Cease • Unspecified
Notification Received	Refer to details for the field Notification Sent.

TABLE 8 show ip bgp neighbor output descriptions (Continued)

Field	Information displayed
TCP Connection state	The state of the connection with the neighbor. The connection can have one of the following states: • LISTEN - Waiting for a connection request. • SYN-SENT - Waiting for a matching connection request after having sent a connection request. • SYN-RECEIVED - Waiting for a confirming connection request acknowledgment after having both received and sent a connection request. • ESTABLISHED - Data can be sent and received over the connection. This is the normal operational state of the connection. • FIN-WAIT-1 - Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent. • FIN-WAIT-2 - Waiting for a connection termination request from the remote TCP. • CLOSE-WAIT - Waiting for a connection termination request from the local user. • CLOSING - Waiting for a connection termination request acknowledgment from the remote TCP. • LAST-ACK - Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request). • TIME-WAIT - Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request. • CLOSED - There is no connection state.
Byte Sent	The number of bytes sent.
Byte Received	The number of bytes received.
Local host	The IP address of the device.
Local port	The TCP port the device is using for the BGP4 TCP session with the neighbor.
Remote host	The IP address of the neighbor.
Remote port	The TCP port the neighbor is using for the BGP4 TCP session with the device.
ISentSeq	The initial send sequence number for the session.
SendNext	The next sequence number to be sent.
TotUnAck	The number of sequence numbers sent by the device that have not been acknowledged by the neighbor.

TABLE 8 show ip bgp neighbor output descriptions (Continued)

Field	Information displayed
TotSent	The number of sequence numbers sent to the neighbor.
ReTrans	The number of sequence numbers that the device retransmitted because they were not acknowledged.
UnAckSeq	The current acknowledged sequence number.
IRcvSeq	The initial receive sequence number for the session.
RcvNext	The next sequence number expected from the neighbor.
SendWnd	The size of the send window.
TotalRcv	The number of sequence numbers received from the neighbor.
DupliRcv	The number of duplicate sequence numbers received from the neighbor.
RcvWnd	The size of the receive window.
SendQue	The number of sequence numbers in the send queue.
RcvQue	The number of sequence numbers in the receive queue.
CngstWnd	The number of times the window has changed.

Displaying route information for a neighbor

You can display routes based on the following criteria:

- A summary of the routes for a specific neighbor.
- Routes received from the neighbor that the device selected as the best routes to their destinations.
- Routes received from the neighbor that are the best BGP4 routes to their destinations, but were not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes).
- Routes that are unreachable because the device does not have a valid RIP, OSPF, or static route to the next hop.
- Routes for a specific network advertised by the device to the neighbor.
- The Routing Information Base (RIB) for a specific network advertised to the neighbor. You can display the RIB regardless of whether the device has already sent it to the neighbor.

Displaying advertised routes

To display the routes the device has advertised to a specific neighbor for a specific network, enter a command such as the following at any level of the CLI.

```
device# show ip bgp neighbors 192.168.4.211 advertised-routes
       There are 2 routes advertised to neighbor 192.168.4.211
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST I:IBGP L:LOCAL
       Network        Next Hop        Metric  LocPrf   Weight   Status
```

```
1       10.102.0.0/24   192.168.2.102   12              32768    BL
2       10.200.1.0/24   192.168.2.102   0               32768    BL
```

You also can enter a specific route.

```
device# show ip bgp neighbors 192.168.4.211 advertised 10.1.1.0/24
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST I:IBGP L:LOCAL
       Network        Next Hop      Metric   LocPrf   Weight  Status
1      10.200.1.0/24  192.168.2.102  0               32768   BL
```

Syntax: show ip bgp neighbor *ip-addr* **advertised-routes** [*ip-addr*/*prefix*]

For information about the fields in this display, refer to Displaying summary route information on page 142. The fields in this display also appear in the **show ip bgp** display.

Displaying the best routes

To display the routes received from a specific neighbor that are the "best" routes to their destinations, enter a command such as the following at any level of the CLI.

```
device#show ip bgp neighbors 192.168.4.211 routes best
```

Syntax: show ip bgp neighbors *ip-addr* **routes best**

For information about the fields in this display, refer to Displaying information for a specific route on page 145. The fields in this display also appear in the **show ip bgp** display.

Displaying the routes with destinations that are unreachable

To display BGP4 routes with destinations that are unreachable using any of the BGP4 paths in the BGP4 route table, enter a command such as the following at any level of the CLI:

```
device(config-bgp)# show ip bgp neighbor 192.168.4.211 routes unreachable
```

Syntax: show ip bgp neighbor *ip-addr* **routes unreachable**

For information about the fields in this display, refer to Displaying summary route information on page 142. The fields in this display also appear in the **show ip bgp** display.

Displaying the Adj-RIB-Out for a neighbor

To display the current BGP4 Routing Information Base (Adj-RIB-Out) for a specific neighbor and a specific destination network, enter a command such as the following at any level of the CLI.

```
device# show ip bgp neighbor 192.168.4.211 rib-out-routes 192.168.1.0/24
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST I:IBGP L:LOCAL
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      10.200.1.0/24       0.0.0.0         0          101         32768  BL
```

The Adj-RIB-Out contains the routes that the device either has most recently sent to the neighbor or is about to send to the neighbor.

Syntax: show ip bgp neighbor *ip-addr* **rib-out-routes** [*ip-addr*/*prefix*]

For information about the fields in this display, refer to Displaying summary route information on page 142. The fields in this display also appear in the **show ip bgp** display.

Displaying peer group information

To display peer-group information, enter a command such as the following at the Privileged EXEC level of the CLI.

```
device# show ip bgp peer-group pg1
1   BGP peer-group is pg
    Description: peer group abc
       SendCommunity: yes
       NextHopSelf: yes
       DefaultOriginate: yes
    Members:
       IP Address: 192.168.10.10, AS: 65111
```

Syntax: show ip bgp peer-group [*peer-group-name*]

Only the parameters that have values different from their defaults are listed.

Displaying summary route information

To display summary statistics for all the routes in the device's BGP4 route table, enter a command such as the following at any level of the CLI.

```
device# show ip bgp routes summary
  Total number of BGP routes (NLRIs) Installed     : 20
  Distinct BGP destination networks                 : 20
  Filtered BGP routes for soft reconfig             : 100178
  Routes originated by this router                  : 2
  Routes selected as BEST routes                    : 19
  BEST routes not installed in IP forwarding table : 1
  Unreachable routes (no IGP route for NEXTHOP)     : 1
  IBGP routes selected as best routes               : 0
  EBGP routes selected as best routes               : 17
```

Syntax: show ip bgp routes summary

TABLE 9 show ip bgp routes output descriptions

This field	Displays
Total number of BGP4 routes (NLRIs) Installed	Number of BGP4 routes the device has installed in the BGP4 route table.
Distinct BGP4 destination networks	Number of destination networks the installed routes represent. The BGP4 route table can have multiple routes to the same network.
Filtered BGP4 routes for soft reconfig	Number of route updates received from soft-reconfigured neighbors or peer groups that have been filtered out but retained.
Routes originated by this device	Number of routes in the BGP4 route table that this device originated.
Routes selected as BEST routes	Number of routes in the BGP4 route table that this device has selected as the best routes to the destinations.
BEST routes not installed in IP forwarding table	Number of BGP4 routes that are the best BGP4 routes to their destinations but were not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes).

TABLE 9 show ip bgp routes output descriptions (Continued)

This field	Displays
Unreachable routes (no IGP route for NEXTHOP)	Number of routes in the BGP4 route table whose destinations are unreachable because the next-hop is unreachable.
IBGP routes selected as best routes	Number of "best" routes in the BGP4 route table that are IBGP routes.
EBGP routes selected as best routes	Number of "best" routes in the BGP4 route table that are EBGP routes.

Displaying the BGP4 route table

BGP4 uses filters that you define as well as the algorithm described in How BGP4 selects a path for a route (BGP best path selection algorithm) on page 29 to determine the preferred route to a destination. BGP4 sends only the preferred route to the IP table. To view all the learned BGP4 routes, you can display the BGP4 table.

To view the BGP4 route table, enter the following command.

```
device# show ip bgp routes
Total number of BGP Routes: 97371
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix             Next Hop        MED       LocPrf     Weight Status
1      10.3.0.0/8          192.168.4.106               100        0      BE
         AS_PATH: 65001 4355 701 80
2      10.4.0.0/8          192.168.4.106               100        0      BE
         AS_PATH: 65001 4355 1
3      10.60.212.0/22      192.168.4.106               100        0      BE
         AS_PATH: 65001 4355 701 1 189
4      10.6.0.0/8          192.168.4.106               100        0      BE
         AS_PATH: 65001 4355 3356 7170 1455
5      10.8.1.0/24         192.168.4.106   0           100        0      BE
         AS_PATH: 65001
```

Syntax: show ip bgp routes [[**network**] *ip-addr*] | *num* | [**age** *secs*] | [**as-path-access-list** *num*] | [**best**] | [**cidr-only**] | [**community** *num* | **no-export** | **no-advertise** | **internet** | **local-as**] | [**community-access-list** *num*] | [**community-list** *num* | [**detail** *option*] | [**filter-list** *num,num,..*] | [**next-hop** *ip-addr*] | [**no-best**] | [**not-installed-best**] | [**prefix-list** *string*] | [**regular-expression** *regular-expression*] | [**route-map** *map-name*] | [**summary**] | [**unreachable**]

The *ip-addr* option displays routes for a specific network. The **network** keyword is optional. You can enter the network address without entering **network** in front of it.

The *num* option specifies the table entry with which you want the display to start. For example, if you want to list entries beginning with table entry 100, specify 100.

The **age** *secs* parameter displays only the routes that have been received or updated more recently than the number of seconds you specify.

The **as-path-access-list***num* parameter filters the display using the specified AS-path ACL.

The **best** parameter displays the routes received from the neighbor that the device selected as the best routes to their destinations.

The **cidr-only** option lists only the routes whose network masks do not match their class network length.

The **community** option lets you display routes for a specific community. You can specify **local-as** , **no-export** , **no-advertise** , **internet** , or a private community number. You can specify the community

number as either two five-digit integer values of up to 1 through 65535, separated by a colon (for example, 12345:6789) or a single long integer value.

The **community-access-list** *num* parameter filters the display using the specified community ACL.

The **community-list** option lets you display routes that match a specific community filter.

The **detail** option lets you display more details about the routes. You can refine your request by also specifying one of the other display options after the detail keyword.

The **filter-list** option displays routes that match a specific address filter list.

The **next-hop** *ip-addr* option displays the routes for a given next-hop IP address.

The **no-best** option displays the routes for which none of the routes to a given prefix were selected as the best route. The IP route table does not contain a BGP4 route for any of the routes listed by the command.

The **not-installed-best** option displays the routes received from the neighbor that are the best BGP4 routes to their destinations, but were not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes).

The **prefix-list** *string* parameter filters the display using the specified IP prefix list.

The **regular-expression** *regular-expression* option filters the display based on a regular expression. Refer to Using regular expressions on page 82.

The **route-map** *map-name* parameter filters the display by using the specified route map. The software displays only the routes that match the match clauses in the route map. Software disregards the route map's set clauses.

The **summary** option displays summary information for the routes.

The **unreachable** option displays the routes that are unreachable because the device does not have a valid RIP, OSPF, or static route to the next-hop.

Displaying the best BGP4 routes

To display all the BGP4 routes in the device's BGP4 route table that are the best routes to their destinations, enter a command such as the following at any level of the CLI

```
device# show ip bgp routes best
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix              Next Hop        MED       LocPrf     Weight Status
1      10.3.0.0/8          192.168.4.106               100        0      BE
         AS_PATH: 65001 4355 701 80
2      10.4.0.0/8          192.168.4.106               100        0      BE
         AS_PATH: 65001 4355 1
3      10.60.212.0/22      192.168.4.106               100        0      BE
         AS_PATH: 65001 4355 701 1 189
4      10.6.0.0/8          192.168.4.106               100        0      BE
         AS_PATH: 65001 4355 3356 7170 1455
5      10.2.0.0/16         192.168.4.106               100        0      BE
         AS_PATH: 65001 4355 701
```

Syntax: show ip bgp routes best

Displaying BGP4 routes whose destinations are unreachable

To display BGP4 routes whose destinations are unreachable using any of the BGP4 paths in the BGP4 route table, enter a command such as the following at any level of the CLI.

```
device# show ip bgp routes unreachable
```

```
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
       Prefix      Next Hop        Metric     LocPrf     Weight Status
1      10.8.8.0/24  192.168.5.1     0          101        0
         AS_PATH: 65001 4355 1
```

Syntax: show ip bgp routes unreachable

Displaying information for a specific route

To display BGP4 network information by specifying an IP address within the network, enter a command such as the following at any level of the CLI.

```
device# show ip bgp 10.3.4.0
Number of BGP Routes matching display condition : 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network            Next Hop        Metric LocPrf Weight Path
*>  10.3.4.0/24         192.168.4.106          100    0      65001 4355 1 1221 ?
       Last update to IP routing table: 0h11m38s, 1 path(s) installed:
         Gateway           Port
         192.168.2.1       2/1
       Route is advertised to 1 peers:
        10.20.20.2(65300)
```

Syntax: show ip bgp [**route**] *ip-addr*|*prefix* [**longer-prefixes**] | *ip-addr*

If you use the **route** option, the display for the information is different, as shown in the following example.

```
device# show ip bgp route 10.3.4.0
Number of BGP Routes matching display condition : 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network       Next Hop      Metric LocPrf Weight Path
*>  10.3.4.0/24   192.168.4.106        100    0      65001 4355 1 1221 ?
       Last update to IP routing table: 0h11m38s, 1 path(s) installed:
         Gateway           Port
         192.168.2.1       2/1
       Route is advertised to 1 peers:
        10.20.20.2(65300)
```

TABLE 10 show ip bgp route output descriptions

This field	Displays
Number of BGP4 Routes matching display condition	The number of routes that matched the display parameters you entered. This is the number of routes displayed by the command.
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the left column of the display, to the left of each route. The status codes are described in the command's output. **NOTE** This field appears only if you *do not* enter the **route** option.
Prefix	The network address and prefix.

TABLE 10 show ip bgp route output descriptions (Continued)

This field	Displays
Next Hop	The next-hop device for reaching the network.
Metric	The value of the route's MED attribute. If the route does not have a metric, this field is blank.
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 - 4294967295.
Weight	The value that this device associates with routes from a specific neighbor. For example, if the device receives routes to the same destination from two BGP4 neighbors, the device prefers the route from the neighbor with the larger weight.
Path	The route AS path. **NOTE** This field appears only if you *do not* enter the **route** option.
Origin code	A character that indicates the route origin. The origin code appears to the right of the AS path (Path field). The origin codes are described in the command output. **NOTE** This field appears only if you *do not* enter the **route** option.

TABLE 10 show ip bgp route output descriptions (Continued)

This field	Displays
Status	The route status, which can be one or more of the following: • A - AGGREGATE.The route is an aggregate route for multiple networks. • B - BEST. BGP4 has determined that this is the optimal route to the destination. **NOTE** If the "b" is lowercase, the software was not able to install the route in the IP route table. • b - NOT-INSTALLED-BEST. The routes received from the neighbor are the best BGP4 routes to their destinations, but were not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes). • C - CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D - DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • H - HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I - INTERNAL. The route was learned through BGP4. • L - LOCAL. The route originated on this device. • M - MULTIPATH. BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE** If the "m" is lowercase, the software was not able to install the route in the IP route table. • S - SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors. **NOTE** This field appears only if you enter the **route** option.

Displaying route details

This example shows the information displayed when you use the **detail** option. In this example, the information for one route is shown.

```
device# show ip bgp routes detail
Total number of BGP Routes: 2
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED
1       Prefix: 10.5.0.0/24,  Status: BME,  Age: 0h28m28s
         NEXT_HOP: 10.1.1.2,  Learned from Peer: 10.1.0.2 (5)
          LOCAL_PREF: 101,  MED: 0,  ORIGIN: igp,  Weight: 10
           AS_PATH: 5
            Adj_RIB_out count: 4,  Admin distance 20
```

Syntax: show ip bgp routes detail

TABLE 11 show ip bgp routes detail output descriptions

This field	Displays
Total number of BGP4 Routes	The number of BGP4 routes.
Status codes	A list of the characters that indicate route status. The status code is appears in the left column of the display, to the left of each route. The status codes are described in the command's output.
Prefix	The network prefix and mask length.
Status	The route status, which can be one or more of the following: • A - AGGREGATE.The route is an aggregate route for multiple networks. • B - BEST. BGP4 has determined that this is the optimal route to the destination. **NOTE** If the "b" is lowercase, the software was not able to install the route in the IP route table. • b - NOT-INSTALLED-BEST. The routes received from the neighbor are the best BGP4 routes to their destinations, but were not installed in the IP route table because the device received better routes from other sources (such as OSPF, RIP, or static IP routes). • C - CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D - DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • H - HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I - INTERNAL. The route was learned through BGP4. • L - LOCAL. The route originated on this device. • M - MULTIPATH. BGP4 load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE** If the "m" is lowercase, the software was not able to install the route in the IP route table. • S - SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors.
Age	The last time an update occurred.
Next_Hop	The next-hop device for reaching the network.
Learned from Peer	The IP address of the neighbor that sent this route.

TABLE 11 show ip bgp routes detail output descriptions (Continued)

This field	Displays
Local_Pref	The degree of preference for this route relative to other routes in the local AS. When the BGP4 algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 through 4294967295.
MED	The route metric. If the route does not have a metric, this field is blank.
Origin	The source of the route information. The origin can be one of the following: • EGP - The routes with these attributes came to BGP4 through EGP. • IGP - The routes with these attributes came to BGP4 through IGP. • INCOMPLETE - The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. When BGP4 compares multiple routes to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Weight	The value this device associates with routes from a specific neighbor. For example, if the device receives routes to the same destination from two BGP4 neighbors, the device prefers the route from the neighbor with the larger weight.
Atomic	Whether network information in this route has been aggregated and this aggregation has resulted in information loss. **NOTE** Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Aggregation ID	The device that originated this aggregation.
Aggregation AS	The AS in which the network information was aggregated. This value applies only to aggregated routes and is otherwise 0.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this route has passed.
Learned From	The IP address of the neighbor from which the device learned the route.
Admin Distance	The administrative distance of the route.
Adj_RIB_out	The number of neighbors to which the route has been or will be advertised. This is the number of times the route has been selected as the best route and placed in the Adj-RIB-Out (outbound queue) for a BGP4 neighbor.
Communities	The communities the route is in.

Displaying BGP4 route-attribute entries

The route-attribute entries table lists the sets of BGP4 attributes stored in device memory. Each set of attributes is unique and can be associated with one or more routes. In fact, the device typically has fewer route attribute entries than routes.

To display the IP route table, enter the following command.

```
device# show ip bgp attribute-entries
```

Syntax: show ip bgp attribute-entries

This example shows the information displayed by this command. A zero value indicates that the attribute is not set.

```
device# show ip bgp attribute-entries
        Total number of BGP Attribute Entries: 7753
1       Next Hop  :192.168.11.1      MED :0               Origin:IGP
        Originator:0.0.0.0           Cluster List:None
         Aggregator:AS Number :0     Router-ID:0.0.0.0      Atomic:FALSE
        Local Pref:100               Communities:Internet
        AS Path   :(65002) 65001 4355 2548 3561 5400 6669 5548
2       Next Hop  :192.168.11.1      Metric   :0           Origin:IGP
        Originator:0.0.0.0           Cluster List:None
         Aggregator:AS Number :0     Router-ID:0.0.0.0      Atomic:FALSE
        Local Pref:100               Communities:Internet
        AS Path   :(65002) 65001 4355 2548
```

TABLE 12 show ip bgp attribute-entries output descriptions

This field	Displays
Total number of BGP4 Attribute Entries	The number of routes contained in this BGP4 route table.
Next Hop	The IP address of the next-hop device for routes that have this set of attributes.
Metric	The cost of the routes that have this set of attributes.
Origin	The source of the route information. The origin can be one of the following: • EGP - The routes with these attributes came to BGP4 through EGP. • IGP - The routes with these attributes came to BGP4 through IGP. • INCOMPLETE - The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPF or RIP. When BGP4 compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this set of attributes has passed.
Aggregator	Aggregator information: • AS Number shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise 0. • Router-ID shows the device that originated this aggregator.

TABLE 12 show ip bgp attribute-entries output descriptions (Continued)

This field	Displays
Atomic	Whether the network information in this set of attributes has been aggregated *and* this aggregation has resulted in information loss. • TRUE - Indicates information loss has occurred • FALSE - Indicates no information loss has occurred **NOTE** Information loss under these circumstances is a normal part of BGP4 and does not indicate an error.
Local Pref	The degree of preference for routes that use these attributes relative to other routes in the local AS.
Communities	The communities to which routes with these attributes belong.
AS Path	The autonomous systems through which routes with these attributes have passed. The local AS is shown in parentheses.

Displaying the routes BGP4 has placed in the IP route table

The IP route table indicates the routes it has received from BGP4 by listing "BGP" as the route type.

To display the IP route table, enter the following command.

```
device# show ip route
```

Syntax: show ip route [*ip-addr* | *num* | **bgp** | **ospf** | **rip** | **isis**]

This example shows the information displayed by this command. Notice that most of the routes in this example have type "B", indicating that their source is BGP4.

```
device# show ip route
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
       Destination         Gateway       Port     Cost    Type
1      10.130.130.0/24    10.11.11.1     ve 1     200/0   B
2      10.130.131.0/24    10.11.11.1     ve 1     200/0   B
```

Displaying route flap dampening statistics

To display route dampening statistics or all the dampened routes, enter the following command at any level of the CLI.

```
device# show ip bgp flap-statistics
Total number of flapping routes: 414
    Status Code  >:best d:damped h:history *:valid
    Network           From         Flaps Since    Reuse    Path
h>  10.50.206.0/23    10.90.213.77  1     0 :0 :13 0 :0 :0  65001 4355 1 701
h>  10.255.192.0/20   10.90.213.77  1     0 :0 :13 0 :0 :0  65001 4355 1 7018
h>  10.252.165.0/24   10.90.213.77  1     0 :0 :13 0 :0 :0  65001 4355 1 7018
h>  10.50.208.0/23    10.90.213.77  1     0 :0 :13 0 :0 :0  65001 4355 1 701
h>  10.33.0.0/16      10.90.213.77  1     0 :0 :13 0 :0 :0  65001 4355 1 701
*>  10.17.220.0/24    10.90.213.77  1     0 :1 :4  0 :0 :0  65001 4355 701 62
```

Syntax: show ip bgp flap-statistics [**regular-expression** *regular-expression* | *address mask* [**longer-prefixes**] | **neighbor** *ip-addr* | **filter-list** *num* **...**]

The **regular-expression***regular-expression* parameter is a regular expression. The regular expressions are the same ones supported for BGP4 AS-path filters.

The *address mask* parameters specify a particular route. If you also use the optional **longer-prefixes** parameter, all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, if you specify **10.157.0.0 longer** , all routes with the prefix 10.157 or that have a longer prefix (such as 10.157.22) are displayed.

The **neighbor***ip-addr* parameter displays route flap dampening statistics only for routes learned from the specified neighbor. You can also display route flap statistics for routes learned from a neighbor by entering the **show ip bgp neighbor flap-statistics** command.

The **filter-list***num* parameter specifies one or more filters. Only routes that have been dampened and that match the specified filters are displayed.

TABLE 13 show ip bgp flap-statistics output descriptions

This field	Displays
Total number of flapping routes	The total number of routes in the BGP4 route table that have changed state and have been marked as flapping routes.
Status code	The dampening status of the route, which can be one of the following: • > - This is the best route among those in the BGP4 route table to the route destination. • d - This route is currently dampened, and thus unusable. • h - The route has a history of flapping and is unreachable now. • * - The route has a history of flapping but is currently usable.
Network	The destination network of the route.
From	The neighbor that sent the route to this device.
Flaps	The number of flaps (state changes) the route has experienced.
Since	The amount of time since the first flap of this route.
Reuse	The amount of time remaining until this route will be un-suppressed and thus be usable again.
Path	The AS-path information for the route.

You can display all dampened routes by entering the **show ip bgp dampened-paths** command.

Displaying the active route map configuration

You can view the active route map configuration (contained in the running configuration) without displaying the entire running configuration by entering the following command at any level of the CLI.

```
device# show route-map
route-map permitnet4 permit 10
 match ip address prefix-list plist1
```

```
route-map permitnet1 permit 1
 match ip address prefix-list plist2
route-map setcomm permit 1
 set community 1234:2345 no-export
route-map test111 permit 111
 match address-filters 11
 set community 11:12 no-export
route-map permit1122 permit 12
 match ip address 11
route-map permit1122 permit 13
 match ip address std_22
```

This example shows that the running configuration contains six route maps. Notice that the match and set statements within each route map are listed beneath the command for the route map itself. In this simplified example, each route map contains only one match or set statement.

To display the active configuration for a specific route map, enter a command such as the following, which specifies a route map name.

```
device# show route-map setcomm
route-map setcomm permit 1
 set community 1234:2345 no-export
```

This example shows the active configuration for a route map named "setcomm".

Syntax: show route-map [*map-name*]

Displaying BGP4 graceful restart neighbor information

To display BGP4 restart information for BGP4 neighbors, enter the **show ip bgp neighbors** command.

```
device# show ip bgp neighbors
    Total number of BGP Neighbors: 6
1   IP Address: 10.50.50.10, AS: 20 (EBGP), RouterID: 10.10.10.20, VRF: default
    State: ESTABLISHED, Time: 0h0m18s, KeepAliveTime: 60, HoldTime: 180
       KeepAliveTimer Expire in 34 seconds, HoldTimer Expire in 163 seconds
       Minimum Route Advertisement Interval: 0 seconds
       RefreshCapability: Received
       GracefulRestartCapability: Received
           Restart Time 120 sec, Restart bit 0
           afi/safi 1/1, Forwarding bit 0
       GracefulRestartCapability: Sent
           Restart Time 120 sec, Restart bit 0
           afi/safi 1/1, Forwarding bit 1
    Messages:    Open    Update  KeepAlive Notification Refresh-Req

....
```

Displaying AS4 details

This section describes the use of the following **show** commands, which produce output that includes information about AS4s.

- **show ip bgp neighbor** shows whether the AS4 capability is enabled.
- **show ip bgp attribute-entries** shows AS4 path values.
- **show ip bgp** shows the route entries with two and AS4 path information.
- **show ip extcommunity-list** shows the members of the extended community.
- **show route-map** shows the presence of any AS4 configuration data.
- **show ip as-path-access-lists** shows the presence of any AS4 configuration data.
- **show ip bgp config** shows the presence of any AS4 configuration data.

Route entries with four-byte path information

The **show ip bgp** command without of any optional parameters display AS4 path information.

```
device# show ip bgp
Total number of BGP Routes: 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal, S
stale
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network        Next Hop      Metric LocPrf Weight Path
*>  10.1.1.0/24    192.168.1.5   1      100    0      90000 100 200 65535
65536 65537 65538 65539 75000
```

Syntax: show ip bgp

Current AS numbers

To display current AS numbers, use the **show ip bgp neighbors** command at any level of the CLI.

```
device# show ip bgp neighbors
  neighbors              Details on TCP and BGP neighbor connections
    Total number of BGP Neighbors: 1
1   IP Address: 192.168.1.1, AS: 7701000 (IBGP), RouterID: 192.168.1.1, VRF: default-
vrf
    State: ESTABLISHED, Time: 0h3m33s, KeepAliveTime: 60, HoldTime: 180
       KeepAliveTimer Expire in 49 seconds, HoldTimer Expire in 177 seconds
    Minimal Route Advertisement Interval: 0 seconds
       RefreshCapability: Received
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
       Sent    : 1       0       5         0            0
       Received: 1       1       5         0            0
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                  Tx: ---        ---           Rx: 0h3m33s    ---
    Last Connection Reset Reason:Unknown
    Notification Sent:     Unspecified
    Notification Received: Unspecified
    Neighbor NLRI Negotiation:
      Peer Negotiated IPV4  unicast  capability
      Peer configured for IPV4 unicast  Routes
    Neighbor AS4 Capability Negotiation:
      Peer Negotiated AS4  capability
      Peer configured for AS4  capability
    Neighbor ipv6 MPLS Label Capability Negotiation:
      Peer Negotiated ipv6 MPLS Label  capability
      Peer configured for ipv6 MPLS Label  capabilit
     As-path attribute count: 1
    Outbound Policy Group:
       ID: 1, Use Count: 1
    TCP Connection state: ESTABLISHED, flags:00000044 (0,0)
     Maximum segment size: 1460
    TTL check: 0, value: 0, rcvd: 64
       Byte Sent:   148, Received: 203
       Local host:  192.168.1.2, Local  Port: 179
       Remote host: 192.168.1.1, Remote Port: 8041
       ISentSeq:    1656867  SendNext:    1657016  TotUnAck:          0
       TotSent:         149  ReTrans:          19  UnAckSeq:    1657016
       IRcvSeq:     1984547  RcvNext:     1984751  SendWnd:       64981
       TotalRcv:        204  DupliRcv:        313  RcvWnd:        65000
       SendQue:           0  RcvQue:            0  CngstWnd:       5840
```

Syntax: show ip bgp neighbors

TABLE 14 show ip bgp neighbors output descriptions

Field	Description
Total number of BGP Neighbors	Shows the total number of BGP neighbors.
IP Address	Shows the IPv4 address of the neighbor.
AS	Shows the Autonomous System (AS) in which the neighbor resides.
EBGP or IBGP	Shows whether the neighbor session is an IBGP session, an EBGP session, or a confederation EBGP session: • EBGP - The neighbor is in another AS. • EBGP_Confed - The neighbor is a member of another sub-AS in the same confederation. • IBGP - The neighbor is in the same AS.
RouterID	Shows the device ID of the neighbor.
VRF	Shows the status of the VRF instance.
State	Shows the state of the device session with the neighbor. The states are from the device's perspective of the session, not the neighbor's perspective. The state can be one of the following values: • IDLE - The BGP4 process is waiting to be started. Usually, enabling BGP4 or establishing a neighbor session starts the BGP4 process. A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • ADMND - The neighbor has been administratively shut down. A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • CONNECT - BGP4 is waiting for the connection process for the TCP neighbor session to be completed. • ACTIVE - BGP4 is waiting for a TCP connection from the neighbor. **NOTE** If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection. • OPEN SENT - BGP4 is waiting for an Open message from the neighbor. • OPEN CONFIRM - BGP4 has received an Open message from the neighbor and is now waiting for either a KeepAlive or Notification message. If the device receives a KeepAlive message from the neighbor, the state changes to ESTABLISHED. If the message is a Notification, the state changes to IDLE. • ESTABLISHED - BGP4 is ready to exchange Update messages with the neighbor. If there is more BGP data in the TCP receiver queue, a plus sign (+) is also displayed.
Time	Shows the amount of time this session has been in its current state.
KeepAliveTime	Shows the keepalive time, which specifies how often this device sends KeepAlive messages to the neighbor.

TABLE 14 show ip bgp neighbors output descriptions (Continued)

Field	Description
HoldTime	Shows the hold time, which specifies how many seconds the device will wait for a KeepAlive or Update message from a BGP4 neighbor before deciding that the neighbor is dead.
KeepAliveTimer Expire	Shows the time when the keepalive timer is set to expire.
HoldTimer Expire	Shows the time when the hold timer is set to expire.
Minimal Route Advertisement Interval	Shows the minimum time elapsed between the route advertisements to the same neighbor.
RefreshCapability	Shows whether the device has received confirmation from the neighbor that the neighbor supports the dynamic refresh capability.
Messages Sent and Received	Shows the number of messages this device has sent to and received from the neighbor. The display shows statistics for the following message types: • Open • Update • KeepAlive • Notification • Refresh-Req
Last Update Time	Shows the list of last time updates were sent and received for the following: • NLRIs • Withdraws

TABLE 14 show ip bgp neighbors output descriptions (Continued)

Field	Description
Last Connection Reset Reason	Shows the reason for ending the previous session with this neighbor. The reason can be one of the following: • No abnormal error has occurred. • Reasons described in the BGP specifications: - Message Header Error - Connection Not Synchronized - Bad Message Length - Bad Message Type - OPEN Message Error - Unsupported Version Number - Bad Peer AS Number - Bad BGP Identifier - Unsupported Optional Parameter - Authentication Failure - Unacceptable Hold Time - Unsupported Capability - UPDATE Message Error - Malformed Attribute List - Unrecognized Well-known Attribute - Missing Well-known Attribute - Attribute Flags Error - Attribute Length Error - Invalid ORIGIN Attribute - Invalid NEXT_HOP Attribute
Last Connection Reset Reason (continued)	• Reasons described in the BGP specifications (continued): - Optional Attribute Error - Invalid Network Field - Malformed AS_PATH - Hold Timer Expired - Finite State Machine Error - Rcv Notification - Reset All Peer Sessions - User Reset Peer Session - Port State Down - Peer Removed - Peer Shutdown - Peer AS Number Change - Peer AS Confederation Change - TCP Connection KeepAlive Timeout - TCP Connection Closed by Remote TCP Data Stream Error Detected

TABLE 14 show ip bgp neighbors output descriptions (Continued)

Field	Description
Notification Sent	Shows an error code corresponding to one of the following errors if the device sends a Notification message from the neighbor. Some errors have subcodes that clarify the reason for the error. The subcode messages are listed underneath the error code messages, wherever applicable. • Message Header Error - Connection Not Synchronized - Bad Message Length - Bad Message Type - Unspecified • Open Message Error - Unsupported Version - Bad Peer AS - Bad BGP Identifier - Unsupported Optional Parameter - Authentication Failure - Unacceptable Hold Time - Unspecified • Update Message Error - Malformed Attribute List - Unrecognized Attribute - Missing Attribute - Attribute Flag Error - Attribute Length Error - Invalid Origin Attribute - Invalid NextHop Attribute - Optional Attribute Error - Invalid Network Field - Malformed AS Path - Unspecified • Hold Timer Expired • Finite State Machine Error • Cease • Unspecified
Notification Received	Shows an error code corresponding to one of the listed errors in the Notification Sent field if the device receives a Notification message from the neighbor.
Neighbor NLRI Negotiation	Shows the state of the device's NLRI negotiation with the neighbor. The states can be one of the following: • Peer negotiated IPV4 unicast capability • Peer negotiated IPV6 unicast capability • Peer configured for IPV4 unicast routes • Peer configured for IPV6 unicast routes

TABLE 14 show ip bgp neighbors output descriptions (Continued)

Field	Description
Neighbor IPv6 MPLS Label Capability Negotiation	Shows the state of the device's IPv6 MPLS label capability negotiation with the neighbor. The states can be one of the following: • Peer negotiated IPv6 MPLS Label capability • Peer configured for IPv6 MPLS Label capability
Neighbor AS4 Capability Negotiation	Shows the state of the device's AS4 capability negotiation with the neighbor. The states can be one of the following: • Peer negotiated AS4 capability • Peer configured for AS4 capability
As-path attribute count	Shows the count of the AS-path attribute.
Outbound Policy Group	Shows the ID and the count used in the outbound policy group.
TCP Connection state	Shows the state of the connection with the neighbor. The connection can have one of the following states: • LISTEN - Waiting for a connection request. • SYN-SENT - Waiting for a matching connection request after having sent a connection request. • SYN-RECEIVED - Waiting for a confirming connection request acknowledgment after having both received and sent a connection request. • ESTABLISHED - Data can be sent and received over the connection. This is the normal operational state of the connection. • FIN-WAIT-1 - Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent. • FIN-WAIT-2 - Waiting for a connection termination request from the remote TCP. • CLOSE-WAIT - Waiting for a connection termination request from the local user. • CLOSING - Waiting for a connection termination request acknowledgment from the remote TCP. • LAST-ACK - Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request). • TIME-WAIT - Waiting for the specific time to ensure that the remote TCP received the acknowledgment of its connection termination request. • CLOSED - There is no connection state.
Maximum segment size	Shows the TCP maximum segment size.
TTL check	Shows the TCP TTL check.
Byte Sent	Shows the number of bytes sent.
Byte Received	Shows the number of bytes received.
Local host	Shows the IPv4 address of the device.

TABLE 14 show ip bgp neighbors output descriptions (Continued)

Field	Description
Local port	Shows the TCP port that the device is using for the BGP4 TCP session with the neighbor.
Remote host	Shows the IPv4 address of the neighbor.
Remote port	Shows the TCP port the neighbor is using for the BGP4 TCP session with the device.
ISentSeq	Shows the initial send sequence number for the session.
SendNext	Shows the next sequence number to be sent.
TotUnAck	Shows the count of sequence numbers sent by the device that have not been acknowledged by the neighbor.
TotSent	Shows the count of the sequence numbers sent to the neighbor.
ReTrans	Shows the count of the sequence numbers that the device retransmitted because they were not acknowledged.
UnAckSeq	Shows the current acknowledged sequence number.
IRcvSeq	Shows the initial receive sequence number for the session.
RcvNext	Shows the next sequence number expected from the neighbor.
SendWnd	Shows the size of the send window.
TotalRcv	Shows the count of the sequence numbers received from the neighbor.
DupliRcv	Shows the count of the duplicate sequence numbers received from the neighbor.
RcvWnd	Shows the size of the receive window.
SendQue	Shows the count of the sequence numbers in the send queue.
RcvQue	Shows the count of the sequence numbers in the receive queue.
CngstWnd	Shows the number of times the window has changed.

Attribute entries

Use the **show ip bgp attribute-entries** command to see AS4 path values, as the following example illustrates.

```
device# show ip bgp attribute-entries
        Total number of BGP Attribute Entries: 18 (0)
1       Next Hop  :192.168.1.6        MED :1             Origin:INCOMP
        Originator:0.0.0.0            Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0     Atomic:None
        Local Pref:100               Communities:Internet
```

```
        Extended Community: SOO 300000:3
        AS Path   :90000 80000 (length 11)
        Address: 0x10e4e0c4  Hash:489 (0x03028536), PeerIdx 0
        Links: 0x00000000, 0x00000000, nlri: 0x10f4804a
        Reference Counts: 1:0:1, Magic: 51
2       Next Hop  :192.168.1.5        Metric   :1         Origin:INCOMP
        Originator:0.0.0.0             Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0     Atomic:None
        Local Pref:100               Communities:Internet
        Extended Community: RT 200000:2
        AS Path   :90000 75000 (length 11)
        Address: 0x10e4e062  Hash:545 (0x0301e8f6), PeerIdx 0
        Links: 0x00000000, 0x00000000, nlri: 0x10f47ff0
        Reference Counts: 1:0:1, Magic: 49
```

Syntax: show ip bgp attribute-entries

Running configuration

AS4s appear in the display of a running configuration, as shown.

```
device# show ip bgp config
Current BGP configuration:
router bgp
 local-as 7701000
 confederation identifier 120000
 confederation peers 80000
 neighbor 192.168.1.2 remote-as 80000
```

Access lists that contain AS4s

AS4s that exist in access lists are displayed by the command, as shown.

```
device# show ip as-path-access-lists
ip as-path access list abc: 1 entries
     seq 10 permit _75000_
ip as-path access list def: 1 entries
     seq 5 permit _80000_
```

Formats of AS4s in show command output

To display the asdot and asdot+ notation for AS4s, enter the **as-format asdot** or **as-format asdot+** commands before you enter the **show ip bgp** command.

```
device# as-format asdot
device-mu2(config)# show ip bgp
Total number of BGP Routes: 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal, S stale
Origin codes: i - IGP, e - EGP,? - incomplete
    Network       Next Hop      Metric LocPrf Weight Path
*> 10.1.1.0/24    192.168.1.5   1      100    0      1.24464 100 200 655
5 1.0 1.1 1.2 1.3 1.9464 ?
```

Syntax: as-format asdot

```
device# as-format asdot+
device# show ip bgp
Total number of BGP Routes: 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal, S stale
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network       Next Hop      Metric LocPrf Weight Path
*>  10.1.1.0/24   192.168.1.5   1      100    0      1.24464 0.100 0.200
0.65535 1.0 1.1 1.2 1.3 1.9464?
```

Syntax: as-format asdot+

Displaying route-map continue clauses

This section contains examples of route-map continuation clauses. Both the route map and the routes to which it applies are described.

This example is a simple illustration of route-map continue clauses. If the match clause of either route map instance 5 or 10 matches, the route map traversal continues at instance 100.

```
route-map test permit 5
 match community my_community1
 set comm-list delete my_community1
 continue 100
route-map test permit 10
 match community my_community2
 set comm-list delete my_community2
 continue 100
route-map test permit 100
 match as-path my_aspath
 set community 1234:5678 additive
```

The following example shows the route map "test." The **show ip bgp route** output shows the consequences of the action in instance 1 (set weight = 10); instance 2 (metric becomes 20); and instance 5 (prepend as_path 300).

```
device# show route-map test
route-map test permit 1
 set weight 10
 route-map test permit 2
 set metric 20
 continue 3
route-map test permit 3
 set community 10:20
 continue 4
route-map test permit 4
 set community 30:40
 continue 5
route-map test permit 5
 set as-path prepend 300
 continue 6

device(config-routemap test)# show ip bgp route
Total number of BGP Routes: 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH m:NOT-INSTALLED-MULTIPATH
S:SUPPRESSED F:FILTERED s:STALE
       Prefix          Next Hop     Metric    LocPrf    Weight Status
1      10.8.8.0/24     10.8.8.3     20        100        0     BE
         AS_PATH: 300 200
```

Syntax: show route-map *map-name*

The *map-name* is the name of the route map.

Syntax: show ip bgp route

In the following example, the continue clause of instance 1 has been changed so that program flow jumps to instance 5. The resulting BGP4 route only has the weight updated and as-path prepended. These changes show route-map *route name*

Syntax: route-map

Syntax: [no] continue *instance number*

Syntax: show ip bgp route

In this example, a match clause has been added to instance 8. Because the match clause of instance 8 does not get fired, the search for the next instance continues to the end of the route-map. The set statements set the weight to 10, prepend 300, prepend 100 to the as-path, set the community to none,

and set the local preference to 70. The results of this route-map traversal appear in the output of the **show ip bgp route** command.

```
device# show route-map test
route-map test permit 1
 set weight 10
 continue 5
route-map test permit 2
 set metric 20
 continue 3
route-map test permit 3
 set community 10:20
 continue 4
route-map test permit 4
 set community 30:40
 continue 5
route-map test permit 5
 set as-path prepend 300
 continue 6
route-map test permit 6
 set as-path prepend 100
 continue 7
route-map test permit 7
 set community none
 set local-preference 70
 continue 8
route-map test deny 8
 match metric 60
 set metric 40
 continue 9
device(config-routemap test)# show ip bgp route
Total number of BGP Routes: 1
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH m:NOT-INSTALLED-MULTIPATH
       S:SUPPRESSED F:FILTERED s:STALE
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      10.8.8.0/24         10.8.8.3          0          70         10     BE
         AS_PATH: 100 300 200
```

Syntax: show route-map

Syntax: shop ip bgp route

For this example, an existing route map is displayed by the **show route-map** command, then the addition of instance 8 adds a deny parameter but no match clause. As a result, no incoming routes are accepted (refer to the last line of the show output).

```
device# show route-map test
route-map test permit 1
 set weight 10
 continue 5
route-map test permit 2
 set metric 20
 continue 3
route-map test permit 3
 set community 10:20
 continue 4
route-map test permit 4
 set community 30:40
 continue 5
route-map test permit 5
 set as-path prepend 300
 continue 6
route-map test permit 6
 set as-path prepend 100
 continue 7
route-map test permit 7
 set community none
 set local-preference 70
 continue 8
device(config-routemap test)#route-map test deny 8
device(config-routemap test)#set metric 40
device(config-routemap test)#continue 9
device(config-routemap test)#show ip bgp route
```

```
BGP Routing Table is empty
```

Syntax: show route-map *map-name*

Updating route information and resetting a neighbor session

The following sections describe how to update route information with a neighbor, reset a session with a neighbor, and close a session with a neighbor.

Any change to a policy (ACL, route map, and so on) is automatically applied to outbound routes that are learned from a BGP4 neighbor or peer group after the policy change occurs. However, you must reset the neighbor to update existing outbound routes.

Any change to a policy is automatically applied to inbound routes that are learned after the policy change occurs. However, to apply the changes to existing inbound routes (those inbound routes that were learned before the policy change), you must reset the neighbors to update the routes using one of the following methods:

- Request the complete BGP4 route table from the neighbor or peer group. You can use this method if the neighbor supports the refresh capability (RFCs 2842 and 2858). Most devices today support this capability.
- Clear (reset) the session with the neighbor or peer group. This is the only method you can use if soft reconfiguration is enabled for the neighbor.

You also can clear and reset the BGP4 routes that have been installed in the IP route table.

Using soft reconfiguration

The soft reconfiguration feature applies policy changes without resetting the BGP4 session. Soft reconfiguration does not request the neighbor or group to send the entire BGP4 table, nor does the feature reset the session with the neighbor or group. Instead, soft reconfiguration stores all the route updates received from the neighbor or group. When you request a soft reset of inbound routes, the software performs route selection by comparing the policies against the stored route updates, instead of requesting the neighbor BGP4 route table or resetting the session with the neighbor.

When you enable the soft reconfiguration feature, it sends a refresh message to the neighbor or group if the neighbor or group supports dynamic refresh. Otherwise, the feature resets the neighbor session. This step is required to ensure that the soft reconfiguration feature has a complete set of updates to use, and occurs only once, when you enable the feature. The feature accumulates all the route updates from the neighbor, eliminating the need for additional refreshes or resets when you change policies in the future.

To use soft reconfiguration:

- Enable the feature.
- Make the policy changes.
- Apply the changes by requesting a soft reset of the inbound updates from the neighbor or group.

Enabling soft reconfiguration

To configure a neighbor for soft reconfiguration, enter a command such as the following.

```
device(config-bgp)# neighbor 10.10.200.102 soft-reconfiguration inbound
```

This command enables soft reconfiguration for updates received from 10.10.200.102. The software dynamically resets the session with the neighbor, then retains all route updates from the neighbor following the reset.

Syntax: [no] neighbor *ip-addr* | *peer-group-name* **soft-reconfiguration inbound**

NOTE
The syntax related to soft reconfiguration is shown.

Placing a policy change into effect

To place policy changes into effect, enter a command such as the following.

```
device(config-bgp)# clear ip bgp neighbor 10.10.200.102 soft in
```

This command updates the routes by comparing the route policies against the route updates that the device has stored. The command does not request additional updates from the neighbor or otherwise affect the session with the neighbor.

Syntax: clear ip bgp neighbor *ip-addr* | *peer-group-name* **soft in**

NOTE
If you do not specify **in** , the command applies to both inbound and outbound updates.

NOTE
The syntax related to soft reconfiguration is shown.

Displaying the filtered routes received from the neighbor or peer group

When you enable soft reconfiguration, the device saves all updates received from the specified neighbor or peer group, including updates that contain routes that are filtered out by the BGP4 route policies in effect on the device. To display the routes that have been filtered out, enter the following command at any level of the CLI.

```
device# show ip bgp filtered-routes
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix            Next Hop        MED       LocPrf     Weight Status
1      10.3.0.0/8         192.168.4.106              100        0      EF
         AS_PATH: 65001 4355 701 80
2      10.4.0.0/8         192.168.4.106              100        0      EF
         AS_PATH: 65001 4355 1
3      10.60.212.0/22     192.168.4.106              100        0      EF
         AS_PATH: 65001 4355 701 1 189
```

The routes displayed are the routes that were filtered out by the BGP4 policies on the device. The device did not place the routes in the BGP4 route table, but did keep the updates. If a policy change causes these routes to be permitted, the device does not need to request the route information from the neighbor, but instead uses the information in the updates.

Syntax: show ip bgp filtered-routes [*ip-addr*] | [**as-path-access-list** *num*] | [**detail**] | [**prefix-list** *string*] [**longer-prefixes**]

The *ip-addr* parameter specifies the IP address of the destination network.

The **as-path-access-list** *num* parameter specifies an AS-path ACL. Only the routes permitted by the AS-path ACL are displayed.

The **detail** parameter displays detailed information for the routes. (The example shows summary information.) You can specify any of the other options after **detail** to further refine the display request.

The **prefix-list** *string* parameter specifies an IP prefix list. Only routes permitted by the prefix list are displayed.

If you also use the optional **longer-prefixes** parameter, then all statistics for routes that match the specified route or have a longer prefix than the specified route are displayed. For example, if you specify 10.157.0.0 longer, then all routes with the prefix 10.157 or that have a longer prefix (such as 10.157.22) are displayed.

Displaying all the routes received from the neighbor

To display all the route information received in route updates from a neighbor since you enabled soft reconfiguration, enter a command such as the following at any level of the CLI.

```
device# show ip bgp neighbor 192.168.4.106 routes
       There are 97345 received routes from  neighbor 192.168.4.106
Searching for matching routes, use ^C to quit...
tatus A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTEREDtatus
A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix              Next Hop        MED     LocPrf    Weight Status
1      10.3.0.0/8          192.168.4.106           100       0      BE
         AS_PATH: 65001 4355 701 8
2      10.4.0.0/8          192.168.4.106           100       0      BE
         AS_PATH: 65001 4355 1
3      10.60.212.0/22      192.168.4.106           100       0      BE
         AS_PATH: 65001 4355 701 1 189
4      10.6.0.0/8          192.168.4.106           100       0      BE
```

Syntax: show ip bgp neighbors *ip-addr* **received-routes** [*detail*]

The **detail** parameter displays detailed information for the routes. This example shows summary information.

NOTE
The syntax for displaying received routes is shown. For complete command syntax, refer to Displaying BGP4 neighbor information on page 132.

Dynamically requesting a route refresh from a BGP4 neighbor

You can easily apply changes to filters that control BGP4 routes received from or advertised to a neighbor, without resetting the BGP4 session between the device and the neighbor. For example, if you add, change, or remove a BGP4 IP prefix list that denies specific routes received from a neighbor, you can apply the filter change by requesting a route refresh from the neighbor. If the neighbor also supports dynamic route refreshes, the neighbor resends its Adj-RIB-Out, its table of BGP4 routes. Using the route refresh feature, you do not need to reset the session with the neighbor.

The route refresh feature is based on the following specifications:

- RFC 2842. This RFC specifies the Capability Advertisement, which a BGP4 device uses to dynamically negotiate a capability with a neighbor.
- RFC 2858 for Multi-protocol Extension.
- RFC 2918, which describes the dynamic route refresh capability

The dynamic route refresh capability is enabled by default and cannot be disabled. When the device sends a BGP4 OPEN message to a neighbor, the device includes a Capability Advertisement to inform the neighbor that the device supports dynamic route refresh.

NOTE
The option for dynamically refreshing routes received from a neighbor requires the neighbor to support dynamic route refresh. If the neighbor does not support this feature, the option does not take effect and the software displays an error message. The option for dynamically re-advertising routes to a neighbor does not require the neighbor to support dynamic route refresh.

Dynamically refreshing routes

The following sections describe how to refresh BGP4 routes dynamically to put new or changed filters into effect.

To request a dynamic refresh of all routes from a neighbor, enter a command such as the following.

```
device(config-bgp-router)# clear ip bgp neighbor 192.168.1.170 soft in
```

This command asks the neighbor to send its BGP4 table (Adj-RIB-Out) again. The device applies its filters to the incoming routes and adds, modifies, or removes BGP4 routes as necessary.

Syntax: clear ip bgp neighbor all | *ip-addr* | *peer-group-name* | *as-num* [**soft-outbound** | **soft** [**in** | **out**]]

The **all** , *ip-addr* , *peer-group-name* , and *as-num* parameters specify the neighbor. The *ip-addr* parameter specifies a neighbor by its IP interface with the device. The *peer-group-name* specifies all neighbors in a specific peer group. The *as-num* parameter specifies all neighbors within the specified AS. The **all** parameter specifies all neighbors.

The **soft-outbound** parameter updates all outbound routes by applying the new or changed filters, but sends only the existing routes affected by the new or changed filters to the neighbor.

The **soft in** and **soft out** parameters specify whether you want to refresh the routes received from the neighbor or sent to the neighbor:

- **soft in** does one of the following:
  - If you enabled soft reconfiguration for the neighbor or peer group, **soft in** updates the routes by comparing the route policies against the route updates that the device has stored. Soft reconfiguration does not request additional updates from the neighbor or otherwise affect the session with the neighbor.
  - If you did not enable soft reconfiguration, **soft in** requests the entire BGP4 route table for the neighbor (Adj-RIB-Out), then applies the filters to add, change, or exclude routes.
  - If a neighbor does not support dynamic refresh, **soft in** resets the neighbor session.
- **soft out** updates all outbound routes, then sends the entire BGP4 router table for the device (Adj-RIB-Out) to the neighbor, after changing or excluding the routes affected by the filters.

If you do not specify **in** or **out** , the device performs both options.

NOTE
The **soft-outbound** parameter updates all outbound routes by applying the new or changed filters, but sends only the existing routes affected by the new or changed filters to the neighbor. The **soft out** parameter updates all outbound routes, then sends the entire BGP4 route table for the device (Adj-RIB-Out) to the neighbor, after changing or excluding the routes affected by the filters. Use **soft-outbound** if only the outbound policy is changed.

To dynamically resend all the device BGP4 routes to a neighbor, enter a command such as the following.

```
device(config-bgp)# clear ip bgp neighbor 192.168.1.170 soft out
```

This command applies filters for outgoing routes to the device BGP4 route table (Adj-RIB-Out), changes or excludes routes accordingly, then sends the resulting Adj-RIB-Out to the neighbor.

NOTE
The Brocade device does not automatically update outbound routes using a new or changed outbound policy or filter when a session with the neighbor goes up or down. Instead, the device applies a new or changed policy or filter when a route is placed in the outbound queue (Adj-RIB-Out).To place a new or changed outbound policy or filter into effect, you must enter a **clear ip bgp neighbor** command regardless of whether the neighbor session is up or down. You can enter the command without optional parameters or with the **soft out** or **soft-outbound** option. Either way, you must specify a parameter for the neighbor (*ip-addr*, *as-num* , *peer-group-name*, or **all**).

Displaying dynamic refresh information

You can use the **show ip bgp neighbors** command to display information for dynamic refresh requests. For each neighbor, the display lists the number of dynamic refresh requests the device has sent to or received from the neighbor and indicates whether the device received confirmation from the neighbor that the neighbor supports dynamic route refresh.

The RefreshCapability field indicates whether this device has received confirmation from the neighbor that the neighbor supports the dynamic refresh capability. The statistics in the Message Sent and Message Received rows under Refresh-Req indicate how many dynamic refreshes have been sent to and received from the neighbor. The statistic is cumulative across sessions.

```
device(config-bgp)# show ip bgp neighbor 10.4.0.2
1   IP Address: 10.4.0.2, AS: 5 (EBGP), RouterID: 100.0.0.1
      Description: neighbor 10.4.0.2
    State: ESTABLISHED, Time: 0h1m0s, KeepAliveTime: 0, HoldTime: 0
      PeerGroup: pg1
      Mutihop-EBGP: yes, ttl: 1
      RouteReflectorClient: yes
      SendCommunity: yes
      NextHopSelf: yes
      DefaultOriginate: yes (default sent)
      MaximumPrefixLimit: 90000
      RemovePrivateAs: : yes
      RefreshCapability: Received
    Route Filter Policies:
      Distribute-list: (out) 20
      Filter-list: (in) 30
      Prefix-list: (in) pf1
      Route-map: (in) setnp1  (out) setnp2
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
      Sent    : 1       1       1         0            0
      Received: 1       8       1         0            0
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                 Tx: 0h0m59s    ---           Rx: 0h0m59s    ---
    Last Connection Reset Reason:Unknown
      Notification Sent:     Unspecified
      Notification Received: Unspecified
    TCP Connection state: ESTABLISHED
      Byte Sent:   115, Received: 492
      Local host:  10.4.0.1, Local  Port: 179
      Remote host: 10.4.0.2, Remote Port: 8053
      ISentSeq:   52837276  SendNext:   52837392  TotUnAck:          0
      TotSent:        116  ReTrans:          0  UnAckSeq:   52837392
      IRcvSeq:  2155052043  RcvNext:  2155052536  SendWnd:       16384
      TotalRcv:       493  DupliRcv:         0  RcvWnd:        16384
      SendQue:          0  RcvQue:           0  CngstWnd:       1460
```

Closing or resetting a neighbor session

You can close a neighbor session or resend route updates to a neighbor.

If you make changes to filters or route maps and the neighbor does not support dynamic route refresh, use the following methods to ensure that neighbors contain only the routes you want them to contain:

- If you close a neighbor session, the device and the neighbor clear all the routes they learned from each other. When the device and neighbor establish a new BGP4 session, they exchange route tables again. Use this method if you want the device to relearn routes from the neighbor and resend its own route table to the neighbor.
- If you use the soft-outbound option, the device compiles a list of all the routes it would normally send to the neighbor at the beginning of a session. However, before sending the updates, the device also applies the filters and route maps you have configured to the list of routes. If the filters or route maps result in changes to the list of routes, the device sends updates to advertise, change, or even withdraw routes on the neighbor as needed. This ensures that the neighbor receives only the routes you want it to contain. Even if the neighbor already contains a route learned from the device that you later decided to filter out, using the soft-outbound option removes that route from the neighbor.

You can specify a single neighbor or a peer group.

To close a neighbor session and thus flush all the routes exchanged by the device and the neighbor, enter the following command.

```
device# clear ip bgp neighbor all
```

Syntax: clear ip bgp neighbor all | *ip-addr* | *peer-group-name* | *as-num* [**soft-outbound** | **soft** [**in** | **out**]]

The **all** , *ip-addr* , *peer-group-name* , and *as-num* parameters specify the neighbor. The *ip-addr* parameter specifies a neighbor by its IP interface with the device. The *peer-group-name* specifies all neighbors in a specific peer group. The *as-num* parameter specifies all neighbors within an AS and has a range of 1 through 4294967295. The **all** keyword specifies all neighbors.

To resend routes to a neighbor without closing the neighbor session, enter a command such as the following.

```
device# clear ip bgp neighbor 10.0.0.1 soft out
```

Clearing and resetting BGP4 routes in the IP route table

To clear BGP4 routes from the IP route table and reset the routes, enter a command such as the following.

```
device# clear ip bgp routes
```

Syntax: clear ip bgp routes [*ip-addr*|*prefix-length*]

Clearing traffic counters

You can clear the counters (reset them to 0) for BGP4 messages.

To clear the BGP4 message counter for all neighbors, enter the following command.

```
device# clear ip bgp traffic
```

Syntax: clear ip bgp traffic

To clear the BGP4 message counter for a specific neighbor, enter a command such as the following.

```
device# clear ip bgp neighbor 10.0.0.1 traffic
```

To clear the BGP4 message counter for all neighbors within a peer group, enter a command such as the following.

```
device# clear ip bgp neighbor PeerGroup1 traffic
```

Syntax: clear ip bgp neighbor all | *ip-addr* | *peer-group-name* | *as-num* **traffic**

The **all** , *ip-addr* , *peer-group-name* , and *as-num* parameters specify the neighbor. The *ip-addr* parameter specifies a neighbor by its IP interface with the device. The *peer-group-name* specifies all neighbors in a specific peer group. The *as-num* parameter specifies all neighbors within the specified AS. The **all** parameter specifies all neighbors.

Clearing diagnostic buffers

The device stores the following BGP4 diagnostic information in buffers:

- The first 400 bytes of the last packet received that contained an error
- The last NOTIFICATION message either sent or received by the device

To display these buffers, use options with the **show ip bgp neighbors** command.

This information can be useful if you are working with Brocade Technical Support to resolve a problem. The buffers do not identify the system time when the data was written to the buffer. If you want to ensure that diagnostic data in a buffer is recent, you can clear the buffers. You can clear the buffers for a specific neighbor or for all neighbors.

If you clear the buffer containing the first 400 bytes of the last packet that contained errors, all the bytes are changed to zeros. The Last Connection Reset Reason field of the BGP4 neighbor table also is cleared.

If you clear the buffer containing the last NOTIFICATION message sent or received, the buffer contains no data.

You can clear the buffers for all neighbors, for an individual neighbor, or for all the neighbors within a specific peer group.

To clear these buffers for neighbor 10.0.0.1, enter the following commands.

```
device# clear ip bgp neighbor 10.0.0.1 last-packet-with-error
device# clear ip bgp neighbor 10.0.0.1 notification-errors
```

Syntax: clear ip bgp neighbor all | *ip-addr* | *peer-group-name* | *as-num* **last-packet-with-error** | **notification-errors**

The The **all** , *ip-addr* , *peer-group-name* , and *as-num* parameters specify the neighbor. The *ip-addr* parameter specifies a neighbor by its IP interface with the device. The *peer-group-name* specifies all neighbors in a specific peer group. The *as-num* parameter specifies all neighbors within the specified AS. The **all** parameter specifies all neighbors.

OSPFv2

- OSPFv2 (NI) PROPOSED 171
- OSPF overview 172
- OSPF point-to-point links 174
- Designated routers in multi-access networks 175
- Designated router election in multi-access networks 175
- OSPF RFC 1583 and 2328 compliance 177
- Reduction of equivalent AS external LSAs 177
- Support for OSPF RFC 2328 Appendix E 179
- OSPF graceful restart 180
- OSPF VRF-Lite for customer edge routers 182
- Configuring OSPF 183
- OSPF non-stop routing 208
- Synchronization of critical OSPF elements 208
- BFD with OSPF NSR 210
- Standby module operations 210
- Enabling and disabling NSR 211
- Adding additional parameters 211
- Disabling configuration 212
- OSPF distribute list 213
- Displaying OSPF information 228
- Clearing OSPF information 253

OSPFv2 (NI) PROPOSED

Documents software features supporting OSPFv2.

The following features are documented in the *Multi-Service IronWare Routing Configuration Guide*.

Features supported	Brocade NetIron XMR Series	Brocade NetIron MLX Series	Brocade NetIron CES 2000 SeriesBASE package	Brocade NetIron CES 2000 Series ME_PREM package	Brocade NetIron CES 2000 Series L3_PREM package	Brocade NetIron CER 2000 Series Base package	Brocade NetIron CER 2000 Series Advanced Services package
OSPFv2	Yes	Yes	No	Yes	Yes	Yes	Yes
RFC 1583 compliant	Yes	Yes	No	Yes	Yes	Yes	Yes

Features supported	Brocade NetIron XMR Series	Brocade NetIron MLX Series	Brocade NetIron CES 2000 SeriesBASE package	Brocade NetIron CES 2000 Series ME_PREM package	Brocade NetIron CES 2000 Series L3_PREM package	Brocade NetIron CER 2000 Series Base package	Brocade NetIron CER 2000 Series Advanced Services package
RFC 2178 compliant	Yes	Yes	No	Yes	Yes	Yes	Yes
RFC 2328 Appendix E	Yes	Yes	No	Yes	Yes	Yes	Yes
Dynamic OSPF memory	Yes	Yes	No	Yes	Yes	Yes	Yes
Graceful restart	Yes	Yes	No	No	No	No	No
Graceful restart helper-mode	Yes	Yes	No	Yes	Yes	Yes	Yes
Block outbound LSA flooding	Yes	Yes	No	Yes	Yes	Yes	Yes
OSPF route redistribution filters	Yes	Yes	No	Yes	Yes	Yes	Yes
Filter OSPF routes into IP route table	Yes	Yes	No	Yes	Yes	Yes	Yes
OSPF group LSA pacing	Yes	Yes	No	Yes	Yes	Yes	Yes
OSPF traps	Yes	Yes	No	Yes	Yes	Yes	Yes
OSPF debugging	Yes	Yes	No	Yes	Yes	Yes	Yes
Encryption code for passwords, authentication keys and community strings	Yes	Yes	No	Yes	Yes	Yes	Yes
VRF-lite for CE routers	Yes	Yes	No	Yes	Yes	Yes	Yes
OSPF NSR	Yes	Yes	No	No	No	No	No
OSPF distribution lists	Yes	Yes	No	Yes	Yes	Yes	Yes

OSPF overview

OSPF is a link-state routing protocol. The protocol uses link-state advertisements (LSA) to update neighboring routers regarding its interfaces and information on those interfaces. The router floods these LSAs to all neighboring routers to update them regarding the interfaces. Each router maintains

an identical database that describes its area topology to help a router determine the shortest path between it and any neighboring router.

The Brocade device supports the following types of LSAs, which are described in RFC 2328 and 3101:

- Router link
- Network link
- Summary link
- Autonomous system (AS) summary link
- AS external link
- Not-So-Stubby Area (NSSA) external link
- Grace LSAs

OSPF is built upon a hierarchy of network components. The highest level of the hierarchy is the Autonomous System (AS) . An autonomous system is defined as a number of networks, all of which share the same routing and administration characteristics.

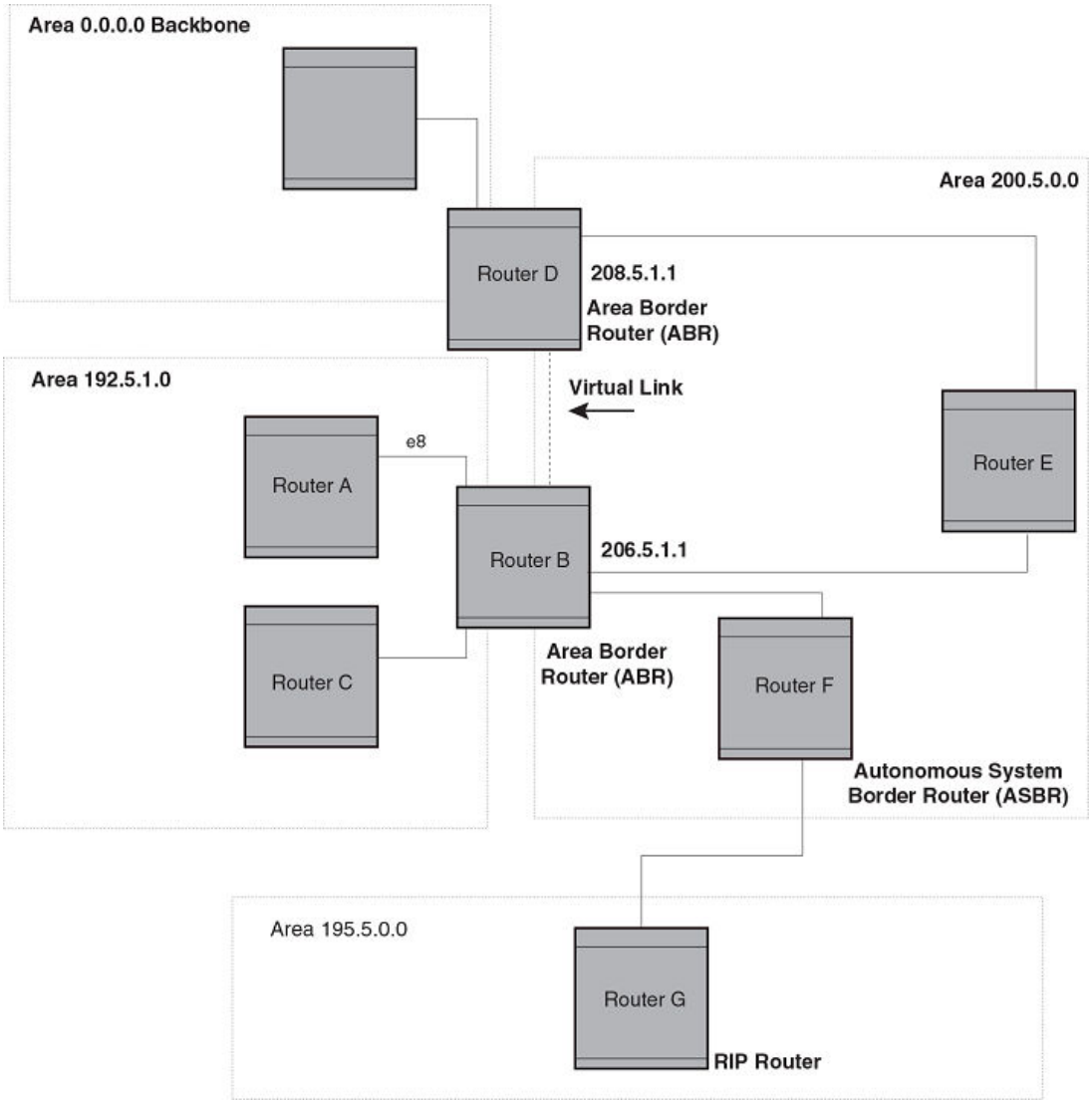
An AS can be divided into multiple areas. Each area represents a collection of contiguous networks and hosts. Areas limit the area to which link-state advertisements are broadcast, thereby limiting the amount of flooding that occurs within the network. An area is represented in OSPF by either an IP address or a number.

You can further limit the broadcast area of flooding by defining an area range. The area range allows you to assign an aggregate value to a range of IP addresses. This aggregate value becomes the address that is advertised instead all of the individual addresses it represents being advertised. You can assign up to 32 ranges in an OSPF area.

An OSPF router can be a member of multiple areas. Routers with membership in multiple areas are known as Area Border Routers (ABRs) . Each ABR maintains a separate topological database for each area the router is in. Each topological database contains all of the LSA databases for each router within a given area. The routers within the same area have identical topological databases. The ABR is responsible for forwarding routing information or changes between its border areas.

An Autonomous System Boundary Router (ASBR) is a router that is running multiple protocols and serves as a gateway to routers outside an area and those operating with different protocols. The ASBR is able to import and translate different protocol routes into OSPF through a process known as redistribution .

FIGURE 8 OSPF operating in a network



OSPF point-to-point links

In an OSPF point-to-point network, where a direct Layer 3 connection exists between a single pair of OSPF routers, there is no need for Designated and Backup Designated Routers, as is the case in OSPF multi-access networks. Without the need for Designated and Backup Designated routers, a point-to-point network establishes adjacency and converges faster. The neighboring routers become adjacent whenever they can communicate directly. In contrast, in broadcast and non-broadcast multi-access (NBMA) networks, the Designated Router and Backup Designated Router become adjacent to all other routers attached to the network.

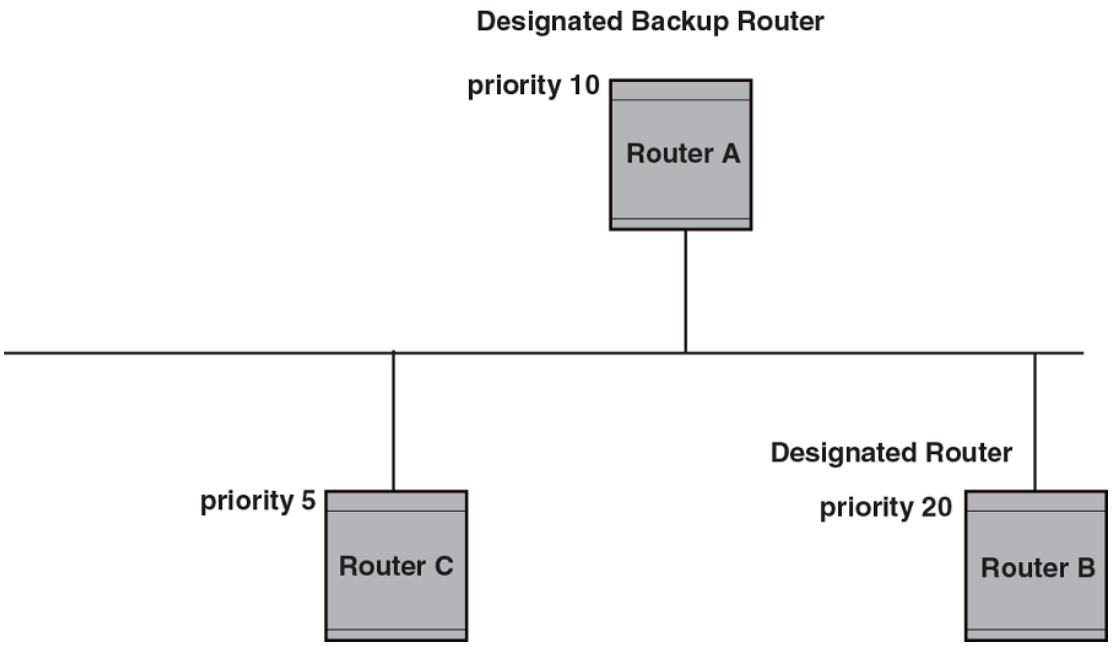
Designated routers in multi-access networks

In a network that has multiple routers attached, OSPF elects one router to serve as the designated router (DR) and another router on the segment to act as the backup designated router (BDR). This arrangement minimizes the amount of repetitive information that is forwarded on the network by forwarding all messages to the designated router and backup designated routers responsible for forwarding the updates throughout the network.

Designated router election in multi-access networks

In a network with no designated router and no backup designated router, the neighboring router with the highest priority is elected as the DR, and the router with the next largest priority is elected as the BDR.

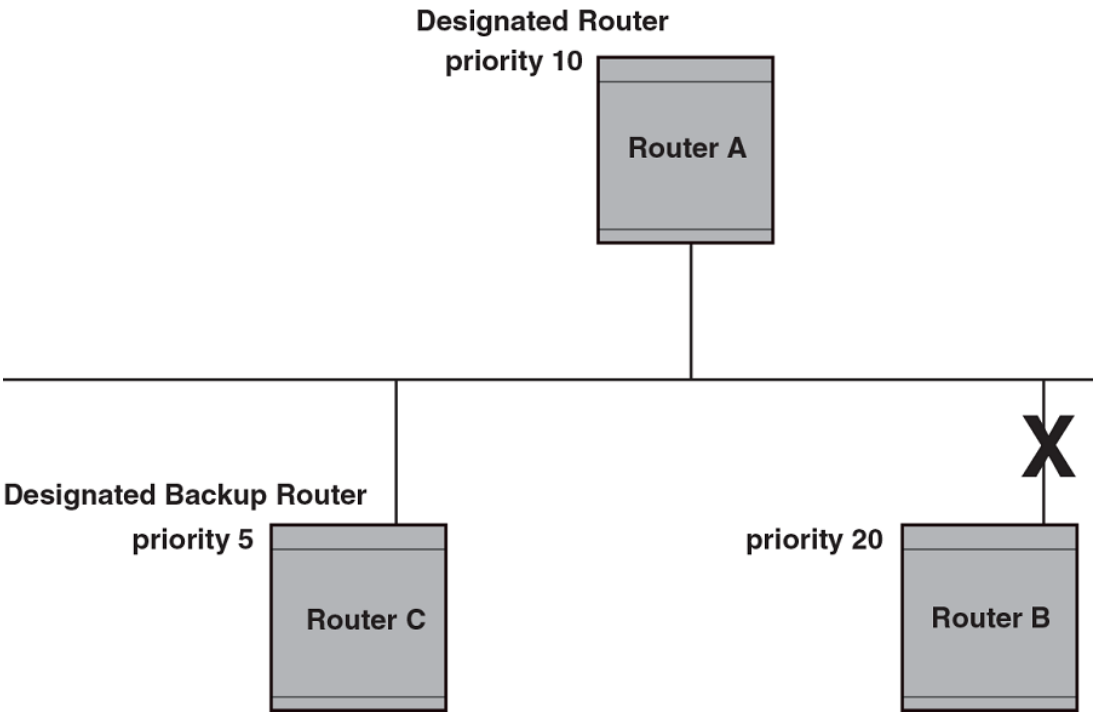
FIGURE 9 Designated and backup router election



If the DR goes off-line, the BDR automatically becomes the DR. The router with the next highest priority becomes the new BDR.

NOTE
Priority is a configurable option at the interface level. You can use this parameter to help bias one router as the DR.

FIGURE 10 Backup designated router becomes designated router



If two neighbors share the same priority, the router with the highest router ID is designated as the DR. The router with the next highest router ID is designated as the BDR.

NOTE
By default, the Brocade device's router ID is the IP address configured on the lowest numbered loopback interface. If the device does not have a loopback interface, the default router ID is the lowest numbered IP address configured on the device.

When multiple routers on the same network are declaring themselves as DRs, then both priority and router ID are used to select the designated router and backup designated routers.

When only one router on the network claims the DR role despite neighboring routers with higher priorities or router IDs, this router remains the DR. This is also true for BDRs.

The DR and BDR election process is performed when one of the following events occurs:

- an interface is in a waiting state and the wait time expires
- an interface is in a waiting state and a hello packet is received that addresses the BDR
- a change in the neighbor state occurs, such as:
  - a neighbor state transitions from ATTEMPT state to a higher state
  - communication to a neighbor is lost
  - a neighbor declares itself to be the DR or BDR for the first time

OSPF RFC 1583 and 2328 compliance

Brocade devices are configured, by default, to be compliant with the RFC 1583 OSPF V2 specification. Brocade devices can also be configured to operate with the latest OSPF standard, RFC 2328.

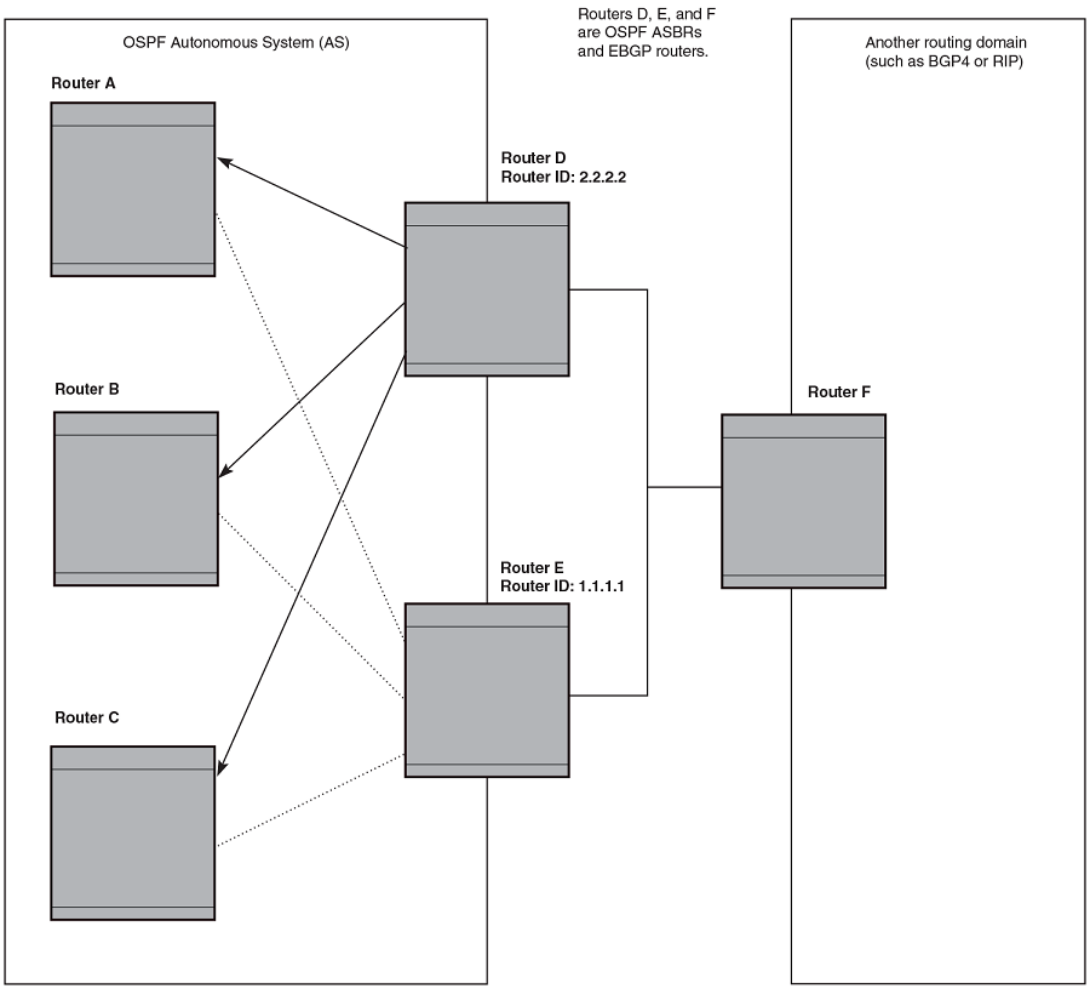
Reduction of equivalent AS external LSAs

An OSPF ASBR uses AS External link advertisements (AS External LSAs) to originate advertisements of a route learned from another routing domain, such as a BGP4 or RIP domain. The ASBR advertises the route to the external domain by flooding AS External LSAs to all the other OSPF routers (except those inside stub networks) within the local OSPF Autonomous System (AS).

In some cases, multiple ASBRs in an AS can originate equivalent LSAs. The LSAs are equivalent when they have the same cost, the same next hop, and the same destination. The device optimizes OSPF by eliminating duplicate AS External LSAs in this case. The device with the lower router ID flushes the duplicate External LSAs from its database and thus does not flood the duplicate External LSAs into the OSPF AS. AS External LSA reduction therefore reduces the size of the link state database on the device. The AS External LSA reduction is described in RFC 2328

In this example, Routers D and E are OSPF ASBRs, and thus communicate route information between the OSPF AS, which contains Routers A, B, and C, and another routing domain, which contains Router F. The other routing domain is running another routing protocol, such as BGP4 or RIP. Routers D, E, and F, therefore, are each running both OSPF and either BGP4 or RIP.

FIGURE 11 AS external LSA reduction



Notice that both Router D and Router E have a route to the other routing domain through Router F.

OSPF eliminates the duplicate AS External LSAs. When two or more devices are configured as ASBRs have equal-cost routes to the same next-hop router in an external routing domain, the ASBR with the highest router ID floods the AS External LSAs for the external domain into the OSPF AS, while the other ASBRs flush the equivalent AS External LSAs from their databases. As a result, the overall volume of route advertisement traffic within the AS is reduced and the devices that flush the duplicate AS External LSAs have more memory for other OSPF data. Because Router D has a higher router ID than Router E, Router D floods the AS External LSAs for Router F to Routers A, B, and C. Router E flushes the equivalent AS External LSAs from its database.

Algorithm for AS external LSA reduction

The AS external LSA reduction example shows the normal AS External LSA reduction feature. The behavior changes under the following conditions:

- There is one ASBR advertising (originating) a route to the external destination, but one of the following happens:

- A second ASBR comes on-line
- A second ASBR that is already on-line begins advertising an equivalent route to the same destination.

In either case above, the router with the higher router ID floods the AS External LSAs and the other router flushes its equivalent AS External LSAs. For example, if Router D is offline, Router E is the only source for a route to the external routing domain. When Router D comes on-line, it takes over flooding of the AS External LSAs to Router F, while Router E flushes its equivalent AS External LSAs to Router F.

- One of the ASBRs starts advertising a route that is no longer equivalent to the route the other ASBR is advertising. In this case, the ASBRs each flood AS External LSAs. Since the LSAs either no longer have the same cost or no longer have the same next-hop router, the LSAs are no longer equivalent, and the LSA reduction feature no longer applies.
- The ASBR with the higher router ID becomes unavailable or is reconfigured so that it is no longer an ASBR. In this case, the other ASBR floods the AS External LSAs. For example, if Router D goes off-line, then Router E starts flooding the AS with AS External LSAs for the route to Router F.

Support for OSPF RFC 2328 Appendix E

Brocade devices support Appendix E in OSPF RFC 2328. Appendix E describes a method to ensure that an OSPF router generates unique link state IDs for type-5 (External) link state advertisements (LSAs) in cases where two networks have the same network address but different network masks.

NOTE
Support for Appendix E of RFC 2328 is enabled automatically and cannot be disabled. No user configuration is required.

Normally, an OSPF router uses the network address alone for the link state ID of the link state advertisement (LSA) for the network. For example, if the router needs to generate an LSA for network 10.1.2.3 255.0.0.0, the router generates ID 10.1.2.3 for the LSA.

However, suppose that an OSPF router needs to generate LSAs for all the following networks:

- 10.0.0.0 255.0.0.0
- 10.0.0.0 255.255.0.0
- 10.0.0.0 255.255.255.0

All three networks have the same network address, 10.0.0.0. Without support for RFC 2328 Appendix E, an OSPF router uses the same link state ID, 10.0.0.0, for the LSAs for all three networks. For example, if the router generates an LSA with ID 10.0.0.0 for network 10.0.0.0 255.0.0.0, this LSA conflicts with the LSA generated for network 10.0.0.0 255.255.0.0 or 10.0.0.0 255.255.255.0. The result is multiple LSAs that have the same ID but that contain different route information.

When appendix E is supported, the router generates the link state ID for a network as the following steps.

1. Does an LSA with the network address as its ID already exist?
   - No - Use the network address as the ID.
   - Yes - Go to "Support for OSPF RFC 2328 Appendix E".
2. Compare the networks that have the same network address, to determine which network is more specific. The more specific network is the one that has more contiguous one bits in its network mask. For example, network 10.0.0.0 255.255.0.0 is more specific than network 10.0.0.0 255.0.0.0,

because the first network has 16 ones bits (255.255.0.0) whereas the second network has only 8 ones bits (255.0.0.0).

- For the less specific network, use the networks address as the ID.
- For the more specific network, use the network's broadcast address as the ID. The broadcast address is the network address, with all ones bits in the host portion of the address. For example, the broadcast address for network 10.0.0.0 255.255.0.0 is 10.0.255.255.

If this comparison results in a change to the ID of an LSA that has already been generated, the router generates a new LSA to replace the previous one. For example, if the router has already generated an LSA for network with ID 10.0.0.0 for network 10.0.0.0 255.255.255.0, the router must generate a new LSA for the network, if the router needs to generate an LSA for network 10.0.0.0 255.255.0.0 or 10.0.0.0 255.0.0.0.

OSPF graceful restart

The OSPF Graceful Restart feature provides support for high-availability routing. With this feature enabled, disruptions in forwarding are minimized and route flapping diminished to provide continuous service during times when a router experiences a restart.

With OSPF graceful restart enabled, a restarting router sends special LSAs to its neighbors called grace LSAs. These LSAs are sent to neighbors either before a planned OSPF restart or immediately after an unplanned restart. The grace LSA specifies a grace period for the neighbors of the restarting router to continue using the existing routes to and through the router after a restart. The restarting router comes up, it continues to use its existing OSPF routes as if nothing has occurred. In the background, the router re-acquires its neighbors prior to the restart and recalculates its OSPF routes and replaces them with new routes as necessary. Once the grace period has passed, the adjacent routers return to normal operation.

OSPF Graceful Restart can be enabled in the following configurations:

- Configuring OSPF Graceful Restart for the Global Instance – In this configuration all OSPF neighbors other than those used by VRFs are made subject to the Graceful Restart capability. The restart timer set globally does not apply to Graceful Restart on a configured VRF.
- Configuring OSPF Graceful Restart per VRF – In this configuration all OSPF neighbors for the specified VRF are made subject to the Graceful Restart capability. The restart timer set for a specific VRF only applies to that VRF.

NOTE
If a 32-slot Brocade NetIron XMR or Brocade MLX series system running a version 03.6.00 or later application image is configured for OSPF graceful restart and intended to be used in switchover or hitless upgrade, the OSPF dead-interval needs to be changed to 60 seconds on OSPF interfaces to ensure that the graceful restart process succeeds without a timeout.

Hitless upgrade support for OSPF graceful restart

OSPF graceful restart experiences minimal packet loss during hitless upgrade on a non-default VRF. On a default VRF, there is no packet loss during hitless upgrade.

OSPF stub router advertisement

OSPFv2 stub router advertisement is an open standard based feature and it is specified in RFC 3137. This feature provides a user with the ability to gracefully introduce and remove an OSPFv2 router from the network by controlling when the data traffic can start and stop flowing through the router in case where there are other OSPFv2 routers present on the network providing alternative paths for the traffic. This feature does not work if there is no alternative for the traffic through other OSPFv2 routers. The router can control the data traffic flowing through it by changing the cost of the paths passing through the configured router. By setting the path cost high the traffic will be redirected to other OSPFv2 routers providing a lower cost path. This change in path cost is accomplished by setting the metric of the links advertised in the Router LSA to a maximum value. When the OSPFv2 router is ready to forward the traffic the links are advertised with the real metric value instead of the maximum value.

The feature is useful for avoiding a loss of traffic during short periods when adjacency failures are detected and traffic is rerouted. Using this feature, traffic can be rerouted before an adjacency failure occurs due to common services interruptions such as a router being shutdown for maintenance.

The feature is also useful during router startup because it gives the router enough time to build up its routing table before forwarding traffic. This can be useful where BGP is enabled on the router because it takes time for the BGP routing table to converge.

You can also configure and set a metric value for the following LSA types:

- Summary (type 3 and type 4)
- External (type 5 and type 7)
- Opaque (type 10, TE link)

OSPF Shortest Path First throttling

Rapid triggering of SPF calculations with exponential back-off to offer the advantages of rapid convergence without sacrificing stability. As the delay increases, multiple topology changes can occur within a single SPF. This dampens network activity due to frequent topology changes.

This scheduling method starts with an initial value after which a configured delay time is followed. If a topology change event occurs the SPF is schedule after the time specified by the initial value, the router starts a timer for the time period specified by a configured hold time value. If no topology events occur during this hold time, the router returns to using the initial delay time.

If a topology event occurs during the hold time period, the next hold time period is recalculated to a value that is double the initial value. If no topology events occur during this extended hold time, the router resets to its initial value. If an event occurs during this extended hold time, the next hold time is doubled again. The doubling occurs as long as topology events occur during the calculated hold times until a configured maximum delay time value is reached or no event occurs (which resets the router to the initial hold time). The maximum value is then held until the hold time expires without a topology change event occurring. At any time that a hold time expires without a topology change event occurring, the router reverts to the initial hold value and begins the process all over again.

For example if you set the initial delay timer to 100 milliseconds, the hold timer to 300 and the maximum hold timer to 2000 milliseconds, the following would occur:

If a topology change occurs the initial delay of 100 milliseconds will be observed. If a topology change occurs during the hold time of 300 milliseconds the hold time is doubled to 600 milliseconds. If a topology change event occurs during the 600 millisecond period, the hold time is doubled again to 1200 milliseconds. If a topology change event occurs during the 1200 millisecond period, the hold time is doubled to 2400 milliseconds. Because the maximum hold time is specified as 2000, the value will be held at 2000. This 2000 millisecond period will then repeat as long as topology events occur within the maximum 2000 millisecond hold time. When a maximum hold time expires without a topology event occurring, the router reverts to the initial delay time and the cycle repeats as described.

The purpose of this feature is to use longer SPF scheduling values during network topology instability.

IETF RFC and internet draft support

The implementation of OSPF Graceful Restart supports the following IETF RFC:

- RFC 3623: Graceful OSPF Restart

NOTE
A secondary management module must be installed for the device to function as a graceful restart device. If the device functions as a graceful restart helper device only, there is no requirement for a secondary management module.

Dynamic OSPF activation and configuration

OSPF is automatically activated when you enable it. The protocol does not require a software reload.

You can configure and save the following OSPF changes without resetting the system:

- All OSPF interface-related parameters (for example: area, hello timer, router dead time cost, priority, re-transmission time, transit delay)
- All area parameters
- All area range parameters
- All virtual-link parameters
- All global parameters
- creation and deletion of an area, interface or virtual link
- Changes to address ranges
- Changes to global values for redistribution
- Addition of new virtual links

OSPF VRF-Lite for customer edge routers

When a type 3, 5, or 7 LSA is sent from a provider edge (PE) router to a customer edge (CE) router, the DN (down) bit in the LSA options field must be set. This prevents any type 3, 5, or 7 LSA messages sent from the CE router to the PE router from being distributed any farther. The PE router ignores messages with the DN bit set and does not add these routes to the VRF routing table.

When you enable VRF-Lite on the CE router, the DN setting is ignored, allowing the CE router to add these routes to the VRF routing table.

To enable VRF-Lite, enter commands such as the following:

```
device(config)# router ospf vrf 1
device(config-ospf-router-vrf-1)# vrf-lite-capability
```

Syntax: [no] vrf-lite-capability

Use the **no** form of the command to disable VRF-Lite. This applies to the VRF instance only. It does not apply to the default VRF.

NOTE
For vpn4 external routes to be installed on CE routers, the domain-tags on PE routers must be different that the domain-tags on CE routers.

NOTE
This command applies to CE routers only. This command does not apply to PE routers.

Configuring OSPF

1. Enable OSPF on the router.
2. Assign the areas to which the router will be attached.
3. Assign individual interfaces to the OSPF areas.
4. Configure route map for route redistribution, if desired.
5. Enable redistribution, if desired.
6. Modify default global and port parameters as required.
7. Modify OSPF standard compliance, if desired.

Configuration rules

The configuration rules are as follows:

- Brocade ICX 6650 devices support a maximum of 676 OSPF interfaces.
- If a router is to operate as an ASBR, you must enable the ASBR capability at the system level.
- Redistribution must be enabled on routers configured to operate as ASBRs.
- All router ports must be assigned to one of the defined areas on an OSPF router. When a port is assigned to an area, all corresponding subnets on that port are automatically included in the assignment.

OSPF parameters

You can modify or set the following global and interface OSPF parameters.

Global parameters

The global OSPF parameters are as follows:

- Modify OSPF standard compliance setting.
- Assign an area.
- Define an area range.
- Define the area virtual link.
- Set global default metric for OSPF.
- Change the reference bandwidth for the default cost of OSPF interfaces.
- Disable or re-enable load sharing.
- Enable or disable default-information-originate.

- Modify Shortest Path First (SPF) timers
- Define external route summarization
- Define redistribution metric type.
- Define redistribution route maps.
- Enable redistribution.
- Change the LSA pacing interval.
- Modify OSPF Traps generated.
- Modify database overflow interval.
- Stub Router advertisement
- Set all the OSPFv2 interfaces to the passive state.

Interface parameters

The interface OSPF parameters are as follows:

- Assign interfaces to an area.
- Define the authentication key for the interface.
- Change the authentication-change interval
- Modify the cost for a link.
- Modify the dead interval.
- Modify MD5 authentication key parameters.
- Modify the priority of the interface.
- Modify the retransmit interval for the interface.
- Modify the transit delay of the interface.

NOTE
You set global level parameters at the OSPF CONFIG Level of the CLI. To reach that level, enter **router ospf** at the global CONFIG Level. Interface parameters for OSPF are set at the interface CONFIG Level using the CLI command **ip ospf.**

When using the Web Management Interface, you set OSPF global parameters using the OSPF configuration panel. All other parameters are accessed through links accessed from the OSPF configuration sheet.

Enable OSPF on the device

When you enable OSPF on the device, the protocol is automatically activated. To enable OSPF on the device, use the following method.

```
device(config)# router ospf
device(config-ospf-router)#
```

This command launches you into the OSPF router level where you can assign areas and modify OSPF global parameters.

Note regarding disabling OSPF

If you disable OSPF, the device removes all the configuration information for the disabled protocol from the running configuration. Moreover, when you save the configuration to the startup configuration file after disabling one of these protocols, all the configuration information for the disabled protocol is removed from the startup configuration file.

The CLI displays a warning message such as the following.

```
device(config-ospf-router)# no router ospf
router ospf mode now disabled. All ospf config data will be lost when writing to
flash!
```

The Web Management Interface does not display a warning message.

If you have disabled the protocol but have not yet saved the configuration to the startup configuration file and reloaded the software, you can restore the configuration information by re-entering the **router ospf** command or by selecting the Web management option to enable the protocol. If you have already saved the configuration to the startup configuration file and reloaded the software, the information is gone.

If you are testing an OSPF configuration and are likely to disable and re-enable the protocol, you might want to make a backup copy of the startup configuration file containing the protocol's configuration information. This way, if you remove the configuration information by saving the configuration after disabling the protocol, you can restore the configuration by copying the backup copy of the startup configuration file onto the flash memory.

Resetting OSPF

The **clear ip ospf all** command globally resets (disables then re-enables) OSPF without deleting the OSPF configuration information. This command is equivalent to entering the commands **no router ospf** followed by **router ospf** . Whereas the **no router ospf** command disables OSPF and removes all the configuration information for the disabled protocol from the running-config, the **router ospf** command re-enables OSPF and restores the OSPF configuration information.

The **clear ip ospf all** command is useful If you are testing an OSPF configuration and are likely to disable and re-enable the protocol. This way, you do not have to save the configuration after disabling the protocol, and you do not have to restore the configuration by copying the backup copy of the startup-config file onto the flash memory.

To reset OSPF without deleting the OSPF configuration, enter the following command at the Global CONFIG level or at the Router OSPF level of the CLI.

```
device# clear ip ospf all
```

Syntax: clear ip ospf all

To reset OSPF for VRFs, enter command such as the following

```
Brocade # clear ip ospf vrf red all
```

Syntax: clear ip ospf vrf [*vrf-name*] **all**

Assign OSPF areas

Once OSPF is enabled on the system, you can assign areas. Assign an IP address or number as the area ID for each area. The area ID is representative of all IP addresses (subnets) on a router port. Each port on a router can support one area.

An area can be normal, a stub, or a Not-So-Stubby Area (NSSA) :

- Normal - OSPF routers within a normal area can send and receive External Link State Advertisements (LSAs).
- Stub - OSPF routers within a stub area cannot send or receive External LSAs. In addition, OSPF routers in a stub area must use a default route to the area's Area Border Router (ABR) or Autonomous System Boundary Router (ASBR) to send traffic out of the area.
- NSSA - The ASBR of an NSSA can import external route information into the area.

- ASBRs redistribute (import) external routes into the NSSA as type 7 LSAs. Type-7 External LSAs are a special type of LSA generated only by ASBRs within an NSSA, and are flooded to all the routers within only that NSSA.
- ABRs translate type 7 LSAs into type 5 External LSAs, which can then be flooded throughout the AS. You can configure address ranges on the ABR of an NSSA so that the ABR converts multiple type-7 External LSAs received from the NSSA into a single type-5 External LSA.

When an NSSA contains more than one ABR, OSPF elects one of the ABRs to perform the LSA translation for NSSA. OSPF elects the ABR with the highest router ID. If the elected ABR becomes unavailable, OSPF automatically elects the ABR with the next highest router ID to take over translation of LSAs for the NSSA. The election process for NSSA ABRs is automatic.

To set up the OSPF areas use the following method.

```
device(config-ospf-router)# area 192.5.1.0
device(config-ospf-router)# area 200.5.0.0
device(config-ospf-router)# area 195.5.0.0
device(config-ospf-router)# area 0.0.0.0
device(config-ospf-router)# write memory
```

Syntax: [no] area { *num* | *ip-addr* }

The *num* and *ip-addr* parameters specify the area number, which can be a number or in IP address format. If you specify a number, the number can be from 0 - 2,147,483,647.

In versions of the Multi-Service IronWare prior to version 03.2.01, up to 18 OSPF areas are supported. Version 03.2.01 of the Multi-Service IronWare software and later support up to 200 OSPF areas.

Assign a totally stubby area

By default, the device sends summary LSAs (LSA type 3) into stub areas. You can further reduce the number of link state advertisements (LSA) sent into a stub area by configuring the device to stop sending summary LSAs (type 3 LSAs) into the area. You can disable the summary LSAs when you are configuring the stub area or later after you have configured the area.

This feature disables origination of summary LSAs, but the device still accepts summary LSAs from OSPF neighbors and floods them to other neighbors. The device can form adjacencies with other routers regardless of whether summarization is enabled or disabled for areas on each router.

When you enter a command or apply a Web management option to disable the summary LSAs, the change takes effect immediately. If you apply the option to a previously configured area, the device flushes all of the summary LSAs it has generated (as an ABR) from the area.

NOTE
This feature applies only when the device is configured as an Area Border Router (ABR) for the area. To completely prevent summary LSAs from being sent to the area, disable the summary LSAs on each OSPF router that is an ABR for the area.

To disable summary LSAs for a stub area, enter commands such as the following.

```
device(config-ospf-router)# area 40 stub 99 no-summary
```

Syntax: [no] area { *num* | *ip-addr* **stub** *cost* [**no-summary**] }

The *num* and *ip-addr* parameters specify the area number, which can be a number or in IP address format. If you specify a number, the number can be from 0 - 2,147,483,647.

The **stub** *cost* parameter specifies an additional cost for using a route to or from this area and can be from 1 - 16777215. There is no default. Normal areas do not use the cost parameter.

The **no-summary** parameter applies only to stub areas and disables summary LSAs from being sent into the area.

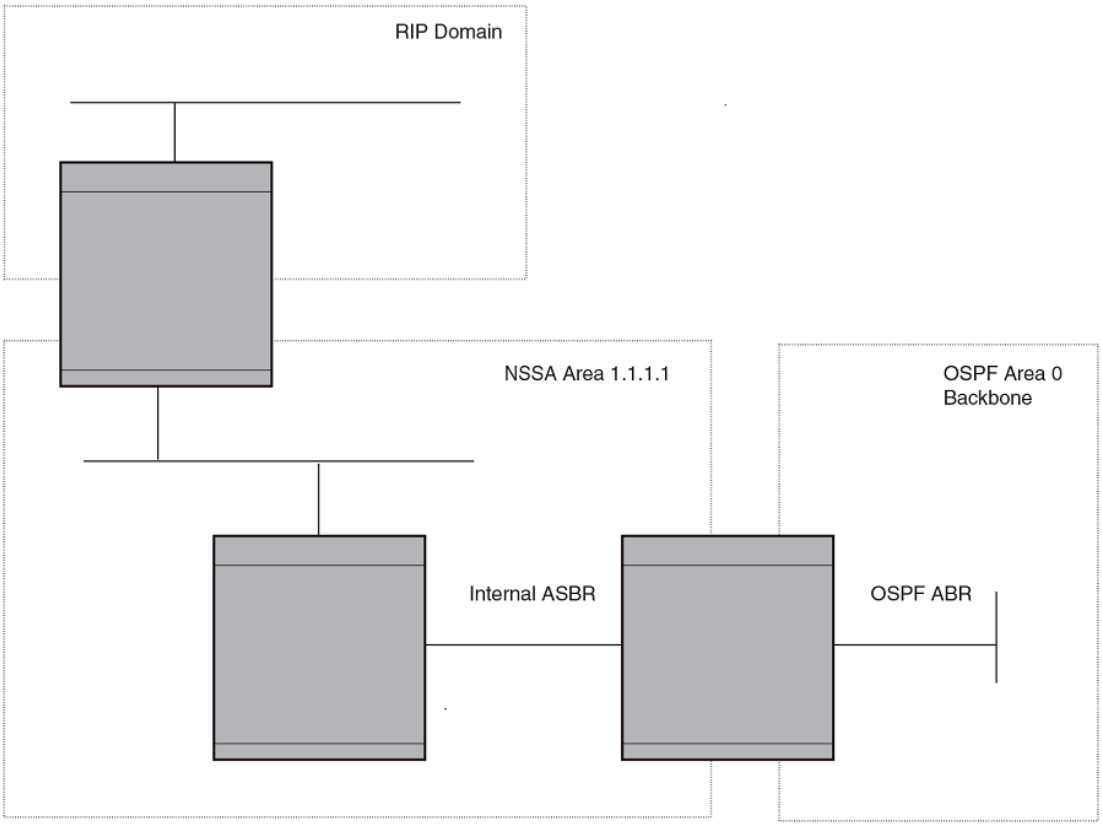
Assign a Not-So-Stubby Area (NSSA)

The OSPF Not So Stubby Area (NSSA) feature enables you to configure OSPF areas that provide the benefits of stub areas, but that also are capable of importing external route information. OSPF does not flood external routes from other areas into an NSSA, but does translate and flood route information from the NSSA into other areas such as the backbone.

NSSAs are especially useful when you want to summarize Type-5 External LSAs (external routes) before forwarding them into an OSPF area. The OSPF specification (RFC 2328) prohibits summarization of Type-5 LSAs and requires OSPF to flood Type-5 LSAs throughout a routing domain. When you configure an NSSA, you can specify an address range for aggregating the external routes that the NSSA's ABR exports into other areas.

The implementation of NSSA is based on RFC 3101.

FIGURE 12 OSPF network containing an NSSA



This example shows two routing domains, a RIP domain and an OSPF domain. The ASBR inside the NSSA imports external routes from RIP into the NSSA as Type-7 LSAs, which the ASBR floods throughout the NSSA.

The ABR translates the Type-7 LSAs into Type-5 LSAs. If an area range is configured for the NSSA, the ABR also summarizes the LSAs into an aggregate LSA before flooding the Type-5 LSAs into the backbone.

Since the NSSA is partially "stubby" the ABR does not flood external LSAs from the backbone into the NSSA. To provide access to the rest of the Autonomous System (AS), the ABR generates a default Type-7 LSA into the NSSA.

Configuring an NSSA

To configure OSPF area 1.1.1.1 as an NSSA, enter the following commands.

```
device(config)# router ospf
device(config-ospf-router)# area 1.1.1.1 nssa 1
device(config-ospf-router)# write memory
```

Syntax: [no] area *area-id* **nssa** [[*nssa-ext-metric*] [**default-information-originate** [**metric** *metric-value* | **metric-type** *type-value*]] [**no-redistribution**] [**translator-always**] [**translator-interval** *stability-interval*]]

The *area-id* parameter specifies the area and has the format xx or xx.xxxx. For example, 49 and 49.2211 are valid area IDs.

The *nssa-ext-metric* parameter specifies the NSSA's advertised external route metric.

The **default-information-originate metric** *metric-value* parameter indicates the cost of the default LSA that originated into the NSSA area. The range is from 1 to 16777215.

The**default-information-originate metric-type** *type-value* parameter indicates the default external LSA type that originated into the NSSA area. The default type is type-2.

The **no-summary** option directs the router to not import type-3 summary LSAs into the NSSA area. The default operation is to import summary LSAs into an NSSA area.

The **no-redistribution** option prevents a NSSA Area Border Router (ABR) from generating external (type-7) LSA into an NSSA area. By default redistribution is enabled in a NSSA.

NOTE
Use this option when an ASBR generates type-5 LSA into normal areas and does not generate type-7 LSA into NSSA area.

The **translator-always** option configures the translator-role.By default, translator-always option is not set. The translator role by default is candidate.

The **translator-interval** *stability-interval* parameter configures the time interval for which an elected NSSA translator continues to perform its duties even after its NSSA translator role has been deposed by another router. By default, the stability-interval is 40 seconds and its range is from 10 to 60 seconds.

NOTE
The device does not inject the default route into an NSSA by default.

To configure additional parameters for OSPF interfaces in the NSSA, use the **ip ospf area** command at the interface level of the CLI.

Disabling the router to perform translations for NSSA LSAs

The **no nssa-translator** command allows you to disable the router to perform translations for NSSA LSAs. When this command is used, type 7 NSSA external LSAs are not translated into type 5 external LSAs. This command is useful when the router is an area border router with many NSSA areas, and does not need to export the NSSA external routes into the backbone.

The following command enables this feature.

```
device(config)# router ospf
device(config-ospf-router)# no nssa-translator
```

Syntax: [no] nssa-translator

Configuring an address range for the NSSA

If you want the ABR that connects the NSSA to other areas to summarize the routes in the NSSA before translating them into Type-5 LSAs and flooding them into the other areas, configure an address range. The ABR creates an aggregate value based on the address range. The aggregate value becomes the address that the ABR advertises instead of advertising the individual addresses represented by the aggregate. You can configure up to 32 ranges in an OSPF area.

To configure an address range in NSSA 1.1.1.1, enter the following commands. This example assumes that you have already configured NSSA 1.1.1.1.

```
device(config)# router ospf
device(config-ospf-router)# area 1.1.1.1 range 209.157.22.1 255.255.0.0
device(config-ospf-router)# write memory
```

Syntax: [no] area { *num* | *ip-addr* **range** *ip-addr ip-mask* [**advertise** | **not-advertise**] }

The *num* and *ip-addr* parameters specify the area number, which can be in IP address format. If you specify a number, the number can be from 0 - 2,147,483,647.

The **range** *ip-addr* parameter specifies the IP address portion of the range. The software compares the address with the significant bits in the mask. All network addresses that match this comparison are summarized in a single route advertised by the router.

The *ip-mask* parameter specifies the portions of the IP address that a route must contain to be summarized in the summary route. In the example above, all networks that begin with 209.157 are summarized into a single route.

The **advertise** and **not-advertise** parameters specify whether you want the device to send type 3 LSAs for the specified range in this area. The default is **advertise** .

Assigning an area range (optional)

You can assign a range for an area, but it is not required. Ranges allow a specific IP address and mask to represent a range of IP addresses within an area, so that only that reference range address is advertised to the network, instead of all the addresses within that range. Each area can have up to 32 range addresses.

To define an area range for subnets on 10.45.5.1 and 10.45.6.2, enter the following command.

```
device(config)# router ospf
device(config-ospf-router)# area 10.45.5.1 range 10.45.0.0 255.255.0.0
device(config-ospf-router)# area 10.45.6.2 range 10.45.0.0 255.255.0.0
```

Syntax: [no] area { *num* | *ip-addr* } **range** *ip-addr ip-mask*

The *num* and *ip-addr* parameters specify the area number, which can be in IP address format.

The **range** *ip-addr* parameter specifies the IP address portion of the range. The software compares the address with the significant bits in the mask. All network addresses that match this comparison are summarized in a single route advertised by the router.

The *ip-mask* parameter specifies the portions of the IP address that a route must contain to be summarized in the summary route. In the example above, all networks that begin with 10.45 are summarized into a single route.

Assigning an area cost (optional parameter)

You can assign a cost for an area, but it is not required. To consolidate and summarize routes at an area boundary, use the **area range cost** command in router configuration mode.

If the **cost** parameter is specified, it will be used (overriding the computed cost) to generate the summary LSA. If the **cost** parameter is not specified, then the existing range metric computation max or min cost of routes falling under this range will be used to generate summary LSA.

NOTE
The area should be already configured before using this command.

Creates an area range entry with ip address 10.1.1.1 and network mask 255.255.255.0 with the area-id 10.

```
device(config)# router ospf
device(config-ospf-router)# area 10 range 10.1.1.1 255.255.255.0
```

Modifies the address range status to **DoNotAdvertise** . Neither the individual intra-area routes falling under range nor the ranged prefix is advertised as summary LSA.

```
device(config)# router ospf
device(config-ospf-router)# area 10 range 10.1.1.1 255.255.255.0 not-advertise
```

Modifies the address range status to advertise and a Type 3 summary link-state advertisement (LSA) can be generated for this address range.

```
device(config)# router ospf
device(config-ospf-router)# area 10 range 10.1.1.1 255.255.255.0 advertise
```

Modifies the address range status to advertise and assign cost for this area range to 10.

```
device(config)# router ospf
device(config-ospf-router)# area 10 range 10.1.1.1 255.255.255.0 advertise cost 10
```

Modifies the address range status to not-advertise and cost from 10 to 5.

```
device(config)# router ospf
device(config-ospf-router)# area 10 range 10.1.1.1 255.255.255.0 not-advertise cost 5
```

Removes the cost from the area range. The area range will be advertised with computed cost which is the max/min(based on RFC 1583 compatibility) of all individual intra-area routes falling under this range.

```
device(config)# router ospf
device(config-ospf-router)# no area 10 range 10.1.1.1 255.255.255.0 cost 5
```

Removes the area range.

```
device(config)# router ospf
device(config-ospf-router)# no area 10 range 10.1.1.1 255.255.255.0
```

NOTE
This command does not work in incremental fashion. So both the optional parameters have to be configured each time. Otherwise it will take the default value.

Syntax: no area { *num* | *ip-addr* **range** *ip-addr ip-mask* [**advertise** | **not-advertise**] **cost** *cost-value* }

The *num* and *ip-addr* parameters specify the area number, which can be in IP address format.

The **range** *ip-addr* parameter specifies the IP address portion of the range. The software compares the address with the significant bits in the mask. All network addresses that match this comparison are summarized in a single route advertised by the router.

The *ip-mask* parameter specifies the portions of the IP address that a route must contain to be summarized in the summary route.

The **advertise** parameter sets the address range status to advertise and generates a Type 3 summary link-state advertisement (LSA). If at least a single route falls under the range, a ranged LSA will be advertised.

The not-advertise parameter sets the address range status to DoNotAdvertise. Neither the individual intra-area routes falling under range nor the ranged prefix is advertised as summary LSA.

The **cost** *cost-value* parameter specifies the cost-value to be used while generating type-3 summary LSA. If the cost value is configured, then configured cost is used while generating the summary LSA. If the cost value is not configured, then computed range cost will be used. The cost-value ranges from 1 - 16777215.

To disable this function, use the **no** form of this command.

Assigning interfaces to an area

Once you define OSPF areas, you can assign interfaces to the areas. All router ports must be assigned to one of the defined areas on an OSPF router. When a port is assigned to an area, all corresponding subnets on that port are automatically included in the assignment.

To assign interface 1/8 of Router A to area 10.5.0.0 and then save the changes, enter the following commands.

```
RouterA(config)# interface e 1/8
RouterA(config-if-e10000-1/8)# ip ospf area 10.5.0.0
RouterA(config-if-e10000-1/8)# write memory
```

Setting all OSPFv2 interfaces to the passive state

You can set all the Open Shortest Path First Version 2 (OSPFv2) interfaces to the default passive state using the **default-passive-interface** command. When you configure the interfaces as passive, the interfaces drop all the OSPFv2 control packets.

To set all the OSPFv2 interfaces to passive, enter the following command.

```
device# configure terminal
device(config)# router ospf vrf A
device(config-ospf-router-vrf-A)# default-passive-interface
```

Syntax: [no] default-passive-interface

Modify interface defaults

OSPF has interface parameters that you can configure. For simplicity, each of these parameters has a default value. No change to these default values is required except as needed for specific network configurations.

Port default values can be modified using the following CLI commands at the interface configuration level of the CLI:

- **ip ospf area**
- **ip ospf auth-change-wait-time**
- **ip ospf authentication-key**
- **ip ospf cost**
- **ip ospf database-filter all out**
- **ip ospf dead-interval**
- **ip ospf hello-interval**
- **ip ospf md5-authentication key-activation-wait-time**
- **ip ospf mtu-ignore**
- **ip ospf passive**
- **ip ospf active**
- **ip ospf priority**
- **ip ospf retransmit-interval**
- **ip ospf transmit-delay**

OSPF interface parameters

The following parameters apply to OSPF interfaces:

- area—Assigns an interface to a specific area. You can assign either an IP address or number to represent an OSPF Area ID. If you assign a number, it can be any value from 0 - 2,147,483,647.
- auth-change-wait-time—OSPF gracefully implements authentication changes to allow all routers to implement the change and thus prevent disruption to neighbor adjacencies. During the authentication-change interval, both the old and new authentication information is supported. The default authentication-change interval is 300 seconds (5 minutes). You change the interval to a value from 0 - 14400 seconds.
- authentication-key *string*—

  By default, the authentication key is encrypted. If you want the authentication key to be in clear text, insert a **0** between **authentication-key** and *string*. For example:

  ```
  device(config-if-e10000-1/8)# ip ospf authentication-key 0 morningadmin
  ```

  The software adds a prefix to the authentication key string in the configuration. For example, the following portion of the code has the encrypted code "2".

  ```
  ip ospf authentication-key 2 $on-o
  ```

  The prefix can be one of the following:

  - 0 = the key string is not encrypted and is in clear text.
  - 1 = the key string uses proprietary simple cryptographic 2-way algorithm.
  - 2 =the key string uses proprietary base64 cryptographic 2-way algorithm (only for Brocade NetIron XMR and Brocade NetIron MLX Series devices).
- cost—Indicates the overhead required to send a packet across an interface. You can modify the cost to differentiate between 100 Mbps, 1 Gbps, and 10 Gbps. The default cost is calculated by dividing 100 million by the bandwidth. For 10 Mbps links, the cost is 10. The cost for 100 Mbps, 1

Gbps, and 10 Gbps links is 1, because the speed of 100 Mbps and 10 Gbps was not in use at the time the OSPF cost formula was devised.

- database-filter—Blocks all outbound LSAs on the OSPF interface.
- dead-interval— Indicates the number of seconds that a neighbor router waits for a hello packet from the current router before declaring the router down. The value can be from 40 - 65535 seconds. The default is 40 seconds.
- hello-interval—Represents the length of time between the transmission of hello packets. The value can be from 1 - 65535 seconds. The default is 10 seconds.
- MD5-authentication activation wait time—The number of seconds the device waits until placing a new MD5 key into effect. The wait time provides a way to gracefully transition from one MD5 key to another without disturbing the network. The wait time can be from 0 - 14400 seconds. The default is 300 seconds (5 minutes).
- MD5-authentication key *string*—The MD5 **authentication-key** is a number from 1 - 255 and identifies the MD5 key that is being used. This parameter is required to differentiate among multiple keys defined on a router.

  By default, the authentication key is encrypted. If you want the authentication key to be in clear text, insert a **0** between **authentication-key** and *string*. For example,

  ```
  device(config-if-e10000-1/8)# ip ospf 1 md-5-authentication key-id 5 key 2
  morningadmin
  ```

  The software adds a prefix to the authentication key string in the configuration. For example, the following portion of the code has the encrypted code "2".

  ```
  ip ospf 1 md-5-authentication key-id 5 key 2 $on-o
  ```

  The prefix can be one of the following:

  - 0 = the key string is not encrypted and is in clear text.
  - 1 = the key string uses proprietary simple cryptographic 2-way algorithm.
  - 2 =the key string uses proprietary base64 cryptographic 2-way algorithm (only for Brocade NetIron XMR Series and Brocade NetIron MLX Series devices).
- mtu-ignore—A database description packet is rejected if the interface MTU specified in the DBD packet is greater than the MTU of the interface shared between the neighbors. To disable the mismatch condition set "mtu-ignore". By default, the mismatch detection is enabled.
- passive—When you configure an OSPF interface to be passive, that interface does not send or receive OSPF route updates. By default, all OSPF interfaces are active and thus can send and receive OSPF route information. Since a passive interface does not send or receive route information, the interface is in effect a stub network. OSPF interfaces are active by default.

NOTE
This option affects all IP subnets configured on the interface. If you want to disable OSPF updates only on some of the IP subnets on the interface, use the **ospf-ignore** or **ospf-passive** parameter with the **ip address** command.

- active—When you configure an OSPFv2 interface to be active, that interface sends or receives all the control packets and forms the adjacency. By default, the **ip ospf active** command is disabled. Whenever you configure the OSPF interfaces to be passive using the **default-passive-interface** command, all the OSPF interfaces stop sending and receiving control packets. To send and receive packets over specific interfaces, you can use the **ip ospf active** command.
- priority—Allows you to modify the priority of an OSPF router. The priority is used when selecting the designated router (DR) and backup designated routers (BDRs). The value can be from 0 - 255. The default is 1. If you set the priority to 0, the device does not participate in DR and BDR election.
- retransmit-interval—The time between retransmissions of link-state advertisements (LSAs) to adjacent routers for this interface. The value can be from 0 - 3600 seconds. The default is 5 seconds.
- transit-delay—The time it takes to transmit Link State Update packets on this interface. The value can be from 0 - 3600 seconds. The default is 1 second.

Rules for OSPF dead interval and hello interval timers

Beginning with version 03.2.00 of the Multi-Service IronWare software, the following rules apply regarding these timers:

- If both the **hello-interval** and **dead-interval** parameters are configured, they will each be set to the values that you have configured.
- If the **hello-interval** parameter is configured, but not the **dead-interval** parameter, the **dead-interval** parameter will be set to a value that is 4 times the value set for the **hello-interval.**
- If the **dead-interval** parameter is configured, but not the **hello-interval** parameter, the **hello-interval.** parameter will be set to a value that is 1/4 the value set for the **dead-interval.** The minimum value for the **hello-interval** is 1.

Change the timer for OSPF authentication changes

When you make an OSPF authentication change, the software uses the authentication-change timer to gracefully implement the change. The software implements the change in the following ways:

- Outgoing OSPF packets - After you make the change, the software continues to use the old authentication to send packets, during the remainder of the current authentication-change interval. After this, the software uses the new authentication for sending packets.
- Inbound OSPF packets - The software accepts packets containing the new authentication and continues to accept packets containing the older authentication for two authentication-change intervals. After the second interval ends, the software accepts packets only if they contain the new authentication key.

The default authentication-change interval is 300 seconds (5 minutes). You change the interval to a value from 0 - 14400 seconds.

OSPF provides graceful authentication change for all the following types of authentication changes in OSPF:

- Changing authentication methods from one of the following to another of the following:
  - Simple text password
  - MD5 authentication
  - No authentication
- Configuring a new simple text password or MD5 authentication key
- Changing an existing simple text password or MD5 authentication key

To change the authentication-change interval, enter a command such as the following at the interface configuration level of the CLI.

```
device(config-if-e10000-2/5)# ip ospf auth-change-wait-time 400
```

Syntax: [no] ip ospf auth-change-wait-time *secs*

The *secs* parameter specifies the interval and can be from 0 - 14400 seconds. The default is 300 seconds (5 minutes).

NOTE
For backward compatibility, the **ip ospf md5-authentication key-activation-wait-time** command is still supported.

Block flooding of outbound LSAs on specific OSPF interfaces

By default, the device floods all outbound LSAs on all the OSPF interfaces within an area. You can configure a filter to block outbound LSAs on an OSPF interface. This feature is particularly useful when you want to block LSAs from some, but not all, of the interfaces attached to the area.

This command blocks all outbound LSAs. Beginning with version 03.6.00, the command has been enhanced to provide options for selective blocking of LSAs.

After you apply filters to block the outbound LSAs, the filtering occurs during the database synchronization and flooding. When a filtering configuration is changed on a interface, all adjacencies on the interface are set to the Extstart state to restart the database exchange process. In cases where an LSA has already been flooded on an interface prior to application of the LSA filter, the LSA will not be flushed out from the remote neighbors. In this situation the user must clear the link state database and the adjacencies on all remote neighbors to flush out the leaked LSAs or wait for the LSAs to be aged out.

If you remove the filters, the blocked LSAs are automatically re-flooded. You do not need to reset OSPF to re-flood the LSAs.

NOTE
You cannot block LSAs on virtual links, and LSA filtering is not supported on sham links.

To apply a filter to an OSPF interface to block flooding of outbound LSAs on the interface, enter the following command at the Interface configuration level for that interface.

```
device(config-if-e10000-1/1)# ip ospf database-filter all out
```

The command in this example blocks all outbound LSAs on the OSPF interface configured on port 1/1.

Syntax: [no] ip ospf database-filter { **all** | **all-external** [**allow-default** | **allow-default-and-type4**] | **all-summary-external** [**allow-default** | **allow-default-and-type4**] } **out**

The **all** parameter directs the router to block all outbound LSAs on the OSPF interface.

The **all-external** option directs the router to allow the following LSAs: Router, Network, Opq-Area-TE, Opq-Link-Graceful and Type-3 Summary while it blocks all Type-4 and Type-5 LSAs unless directed by one of the following keywords:

allow-default - allows only Type-5 default LSAs.

allow-default-and-type4 - allows Type-5 default LSAs and all Type 4 LSAs.

The **all-summary-external** option directs the router to allow the following LSAs: Router, Network, Opq-Area-TE and Opq-Link-Graceful while it blocks all Type-3, Type-4 and Type-5 LSAs unless directed by one of the following keywords:

allow-default - allows only Type-3 or Type-5 default LSAs.

allow-default-and-type4 - allows Type-3 or Type-5 default LSAs and all Type 4 LSAs.

All Type-7 LSAs are always filtered if the **ip ospf database-filter** command is enabled.

By default, OSPF LSA filtering is disabled on all interfaces.

To remove the filter, enter a command such as the following.

```
device(config-if-e10000-1/1)# no ip ospf database-filter all out
```

Assign virtual links

All ABRs (area border routers) must have either a direct or indirect link to the OSPF backbone area (0.0.0.0 or 0). If an ABR does not have a physical link to the area backbone, the ABR can configure a virtual link to another router within the same area, which has a physical connection to the area backbone.

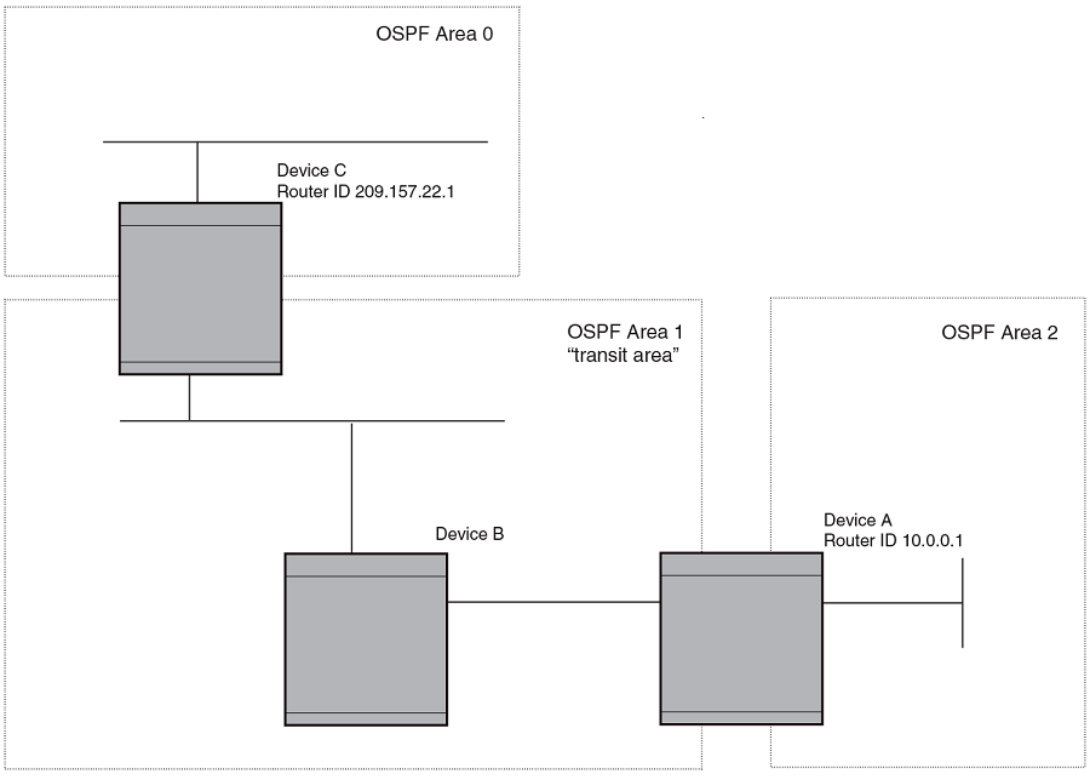
The path for a virtual link is through an area shared by the neighbor ABR (router with a physical backbone connection), and the ABR requiring a logical connection to the backbone.

Two parameters fields must be defined for all virtual links--transit area ID and neighbor router:

- The transit area ID represents the shared area of the two ABRs and serves as the connection point between the two routers. This number should match the area ID value.
- The neighbor router field is the router ID (IP address) of the router that is physically connected to the backbone, when assigned from the router interface requiring a logical connection. When assigning the parameters from the router with the physical connection, the router ID is the IP address of the router requiring a logical connection to the backbone.

NOTE
By default, the Brocade device's router ID is the IP address configured on the lowest numbered loopback interface. If the device does not have a loopback interface, the default router ID is the lowest numbered IP address configured on the device. When you establish an area virtual link, you must configure it on both of the routers (both ends of the virtual link).

FIGURE 13 Defining OSPF virtual links within a network



The example shows an OSPF area border router, Device A, that is cut off from the backbone area (area 0). To provide backbone access to Device A, you can add a virtual link between Device A and Device C using area 1 as a transit area. To configure the virtual link, you define the link on the router that is at each end of the link. No configuration for the virtual link is required on the routers in the transit area.

To define the virtual link on Device A, enter the following commands.

```
device A(config)# router ospf
device A(config-ospf-router)# area 2
device A(config-ospf-router)# area 1
device A(config-ospf-router)# area 1 virtual-link 209.157.22.1
device A(config-ospf-router)# write memory
```

Enter the following commands to configure the virtual link on Device C.

```
device C(config)# router ospf
device C(config-ospf-router)# area 0
device C(config-ospf-router)# area 1
device C(config-ospf-router)# area 1 virtual-link 10.0.0.1
```

Syntax: [no] area { *ip-addr* | *num* } [**virtual-link** *router-id* [**authentication-key** *string* | **dead-interval** *num* | **hello-interval** *num* | **retransmit-interval** *num* | **transmit-delay** *num* | **md5-authentication key-activation-wait-time** *num* | **md5-authentication key-id** *num* **key** [**0** | **1**] *string*]]

The **area** *ip-addr* and *num* parameters specify the transit area.

The **virtual-link** *router-id* parameter specifies the router ID of the OSPF router at the remote end of the virtual link. To display the router ID on a device, enter the **show ip** command.

Modify virtual link parameters

OSPF has some parameters that you can modify for virtual links. Notice that these are the same parameters as the ones you can modify for physical interfaces.

You can modify default values for virtual links using the following CLI command at the OSPF router level of the CLI, as shown in the following syntax:

Syntax: [no] area { *ip-addr* | *num* } [**virtual-link** *router-id* **dead-interval** *num* | **hello-interval** *num* | **retransmit-interval** *num* | **transmit-delay** *num* | **authentication-key** *string* | **md5-authentication key** *key-string* | **md5-authentication key-activation-wait-time** *num*]

Virtual link parameter descriptions

You can modify the following virtual link interface parameters:

area *ip-addr* | *num*	The IP address or number of the transit area.
virtual-link *router-id*	The router ID of the OSPF router at the remote end of the virtual link.
dead-interval *num*	The number of seconds that a neighbor router waits for a hello packet from the current router before declaring the router down. The value can be from 1 - 65535 seconds. The default is 40 seconds.
hello-interval *num*	The length of time between the transmission of hello packets. The range is 1 - 65535 seconds. The default is 10 seconds.
retransmit-interval *num*	The interval between the re-transmission of link state advertisements to router adjacencies for this interface. The range is 0 - 3600 seconds. The default is 5 seconds.

transmit-delay *num*	The period of time it takes to transmit Link State Update packets on the interface. The range is 0 - 3600 seconds. The default is 1 second.

authentication-key *string*	This parameter allows you to assign different authentication encryption methods on a port-by-port basis. OSPF supports three methods of authentication for each interface: none, simple encryption, and base 64 encryption. Only one encryption method can be active on an interface at a time.
	
	The simple encryption and base 64 encryption methods requires you to configure an alphanumeric password on an interface. The password can be up to eight characters long. All OSPF packets transmitted on the interface contain this password. All OSPF packets received on the interface are checked for this password. If the password is not present, then the packet is dropped.
	
	By default, the authentication key is encrypted. If you want the authentication key to be in clear text, insert a **0** between **key** and *string*. For example,
	
	`device C(config-ospf-router)# area 1 virtual-link 10.0.0.1 authentication-key 0 afternoon`
	
	The software adds a prefix to the authentication key string in the configuration. For example, the following portion of the code has the encrypted code "2".
	
	`area 1 virtual-link 12.12.12.25 authentication-key 2 $on-o`
	
	The prefix can be one of the following:
	• 0 = the key string is not encrypted and is in clear text
	• 1 = the key string uses proprietary simple cryptographic 2-way algorithm
	• 2 = the key string uses proprietary base64 cryptographic 2-way algorithm (only for Brocade NetIron XMR and Brocade NetIron MLX Series devices)

md5-authentication key *string*	The MD5 **key** is a number from 1 - 255 and identifies the MD5 key that is being used. This parameter is required to differentiate among multiple keys defined on a router.
	
	When MD5 is enabled, the key-string is an alphanumeric password of up to 16 characters that is later encrypted and included in each OSPF packet transmitted. You must enter a password in this field when the system is configured to operate with either simple or MD5 authentication.
	
	By default, the MD5 authentication key is encrypted. If you want the authentication key to be in clear text, insert a **0** between **key** and *string*. For example,
	
	`device(config-ospf-router)# area 1 virtual-link 10.0.0.1 md5-authentication key-id 5 key evening`
	
	The software adds a prefix to the authentication key string in the configuration. For example, the following portion of the code has the encrypted code "2".
	
	`device(config-ospf-router)# area 1 virtual-link 12.12.12.25 md5-authentication key-id 5 key 2 $on-o`
	
	The prefix can be one of the following:
	• 0 = the key string is not encrypted and is in clear text
	• 1 = the key string uses proprietary simple cryptographic 2-way algorithm
	• 2 = the key string uses proprietary base64 cryptographic 2-way algorithm (only for Brocade NetIron XMR and Brocade NetIron MLX Series and devices)

md5-authentication wait time	This parameter determines when a newly configured MD5 authentication key is valid. This parameter provides a graceful transition from one MD5 key to another without disturbing the network. All new packets transmitted after the key activation wait time interval use the newly configured MD5 Key. OSPF packets that contain the old MD5 key are accepted for up to five minutes after the new MD5 key is in operation. The range for the key activation wait time is from 0 - 14400 seconds. The default value is 300 seconds.

Changing the reference bandwidth for the cost on OSPF interfaces

Each interface on which OSPF is enabled has a cost associated with it. The device advertises its interfaces and their costs to OSPF neighbors. For example, if an interface has an OSPF cost of ten, the device advertises the interface with a cost of ten to other OSPF routers.

By default, an interface's OSPF cost is based on the port speed of the interface. The cost is calculated by dividing the reference bandwidth by the port speed. The default reference bandwidth is 100 Mbps, which results in the following default costs:

- 10 Mbps port - 10
- All other port speeds - 1

You can change the reference bandwidth, to change the costs calculated by the software.

The software uses the following formula to calculate the cost:

Cost = reference-bandwidth/interface-speed

If the resulting cost is less than 1, the software rounds the cost up to 1. The default reference bandwidth results in the following costs:

- 10 Mbps port's cost = 100/10 = 10
- 100 Mbps port's cost = 100/100 = 1
- 1000 Mbps port's cost = 100/1000 = 0.10, which is rounded up to 1
- 10 Gbps port's cost = 100/10000 = 0.01, which is rounded up to 1

The bandwidth for interfaces that consist of more than one physical port is calculated as follows:

- LAG group - The combined bandwidth of all the ports.
- Virtual interface - The combined bandwidth of all the ports in the port-based VLAN that contains the virtual interface.

The default reference bandwidth is 100 Mbps. You can change the reference bandwidth to a value from 1 - 4294967.

If a change to the reference bandwidth results in a cost change to an interface, the device sends a link-state update to update the costs of interfaces advertised by the device.

NOTE
If you specify the cost for an individual interface, the cost you specify overrides the cost calculated by the software.

Interface types to which the reference bandwidth does not apply

Some interface types are not affected by the reference bandwidth and always have the same cost regardless of the reference bandwidth in use:

- The cost of a loopback interface is always 1.
- The cost of a virtual link is calculated using the Shortest Path First (SPF) algorithm and is not affected by the auto-cost feature.
- The bandwidth for tunnel interfaces is 9 Kbps and is also subject to the auto-cost reference bandwidth setting.

Changing the reference bandwidth

To change the reference bandwidth, enter a command such as the following at the OSPF configuration level of the CLI.

```
device(config)# router ospf
device(config-ospf-router)# auto-cost reference-bandwidth 500
```

The reference bandwidth specified in this example results in the following costs:

- 10 Mbps port's cost = 500/10 = 50
- 100 Mbps port's cost = 500/100 = 5
- 1000 Mbps port's cost = 500/1000 = 0.5, which is rounded up to 1

The costs for 10 Mbps and 100 Mbps ports change as a result of the changed reference bandwidth. Costs for higher-speed interfaces remain the same.

Syntax: [no] auto-cost { **reference-bandwidth** *num* | **use-active-ports** }

The *num* parameter specifies the reference bandwidth and can be a value from 1 - 4294967. The default is 100.

To restore the reference bandwidth to its default value and thus restore the default costs of interfaces to their default values, enter the following command.

```
device(config-ospf-router)# no auto-cost reference-bandwidth
```

Determining cost calculation for active ports only on LAG and VE interfaces

The default operation is for cost calculation of OSPF interfaces to be based upon all configured ports. There is also an option for the **auto-cost reference-bandwidth** command for the calculation of OSPF costs on active ports of LAG and VE interfaces. This option allows you to calculate cost based on the ports that are currently active. The following example enables cost calculation for currently active ports.

```
device(config-ospf-router)# auto-cost use-active-ports
```

The **use-active-ports** option enables cost calculation for currently active ports only. This option does not have any effect on non-VE or non-LAG interfaces. The default operation is for costs to be based on configured ports.

Define redistribution filters

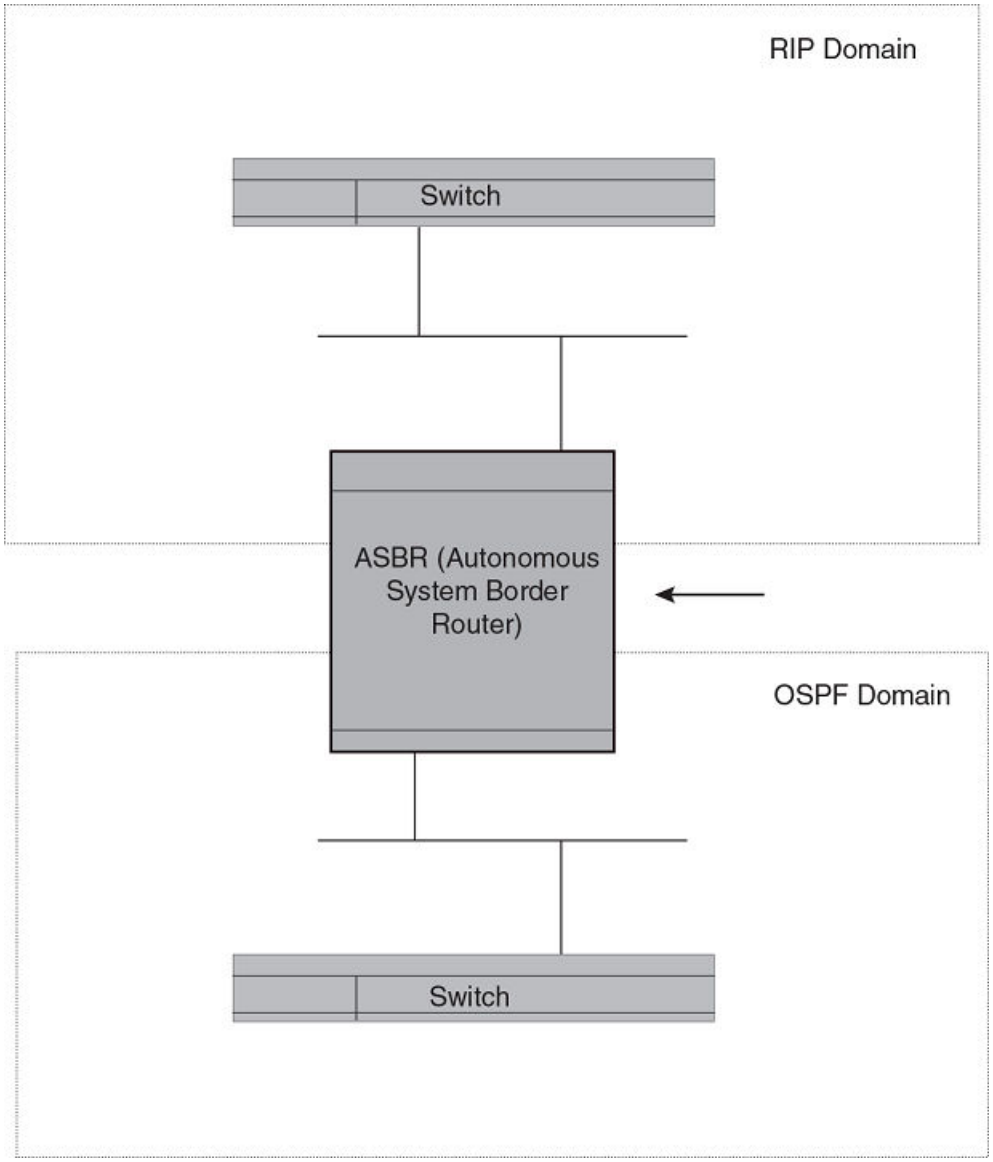
Route redistribution imports and translates different protocol routes into a specified protocol type. On the device, redistribution is supported for static routes, ISIS, OSPF, RIP, and BGP4. OSPF redistribution supports the import of static, ISIS, RIP, and BGP4 routes into OSPF routes.

NOTE
The device advertises the default route into OSPF even if redistribution is not enabled, and even if the default route is learned through an IBGP neighbor. IBGP routes (including the default route) are not redistributed into OSPF by OSPF redistribution (for example, by the OSPF **redistribute** command).

In this example, an administrator wants to configure the device acting as the ASBR (Autonomous System Boundary Router) between the RIP domain and the OSPF domain to redistribute routes between the two domains.

NOTE
The ASBR must be running both RIP and OSPF protocols to support this activity.

FIGURE 14 Redistributing OSPF and static routes to RIP routes



You also have the option of specifying import of just ISIS, RIP, OSPF, BGP4, or static routes, as well as specifying that only routes for a specific network or with a specific cost (metric) be imported, as shown in the command syntax below.

Syntax: [no] redistribute { **bgp** | **connected** | **rip** | **static** [**route-map** *map-name*] }

NOTE
Prior to software release 04.1.00, the **redistribution** command is used instead of **redistribute** .

For example, to enable redistribution of RIP and static IP routes into OSPF, enter the following commands.

```
device(config)# router ospf
device(config-ospf-router)# redistribute rip
device(config-ospf-router)# redistribute static
device(config-ospf-router)# write memory
```

Modify default metric for redistribution

The default metric is a global parameter that specifies the cost applied to all OSPF routes by default. The default value is 10. You can assign a cost from 1 - 65535.

NOTE
You also can define the cost on individual interfaces. The interface cost overrides the default cost.

To assign a default metric of 4 to all routes imported into OSPF, enter the following commands.

```
device(config)# router ospf
device(config-ospf-router)# default-metric 4
```

Syntax: default-metric *value*

The *value* can be from 1 - 15. The default is 10.

Enable route redistribution

NOTE
Do not enable redistribution until you have configured the redistribution route map. Otherwise, you might accidentally overload the network with routes you did not intend to redistribute.

To enable redistribution of RIP and static IP routes into OSPF, enter the following commands.

```
device(config)# router ospf
device(config-ospf-router)# redistribute rip
device(config-ospf-router)# redistribute static
device(config-ospf-router)# write memory
```

Example using a route map

To configure a route map and use it for redistribution of routes into OSPF, enter commands such as the following.

```
device(config)# ip route 1.1.0.0 255.255.0.0 10.95.7.30
device(config)# ip route 1.2.0.0 255.255.0.0 10.95.7.30
device(config)# ip route 1.3.0.0 255.255.0.0 10.95.7.30
device(config)# ip route 4.1.0.0 255.255.0.0 10.95.6.30
device(config)# ip route 4.2.0.0 255.255.0.0 10.95.6.30
device(config)# ip route 4.3.0.0 255.255.0.0 10.95.6.30
device(config)# ip route 4.4.0.0 255.255.0.0 10.95.6.30 5
device(config)# route-map abc permit 1
device(config-routemap abc)# match metric 5
device(config-routemap abc)# set metric 8
device(config-routemap abc)# router ospf
device(config-ospf-router)# redistribute static route-map abc
```

The commands in this example configure some static IP routes, then configure a route map and use the route map for redistributing static IP routes into OSPF.

The **ip route** commands configure the static IP routes. The **route-map** command begins configuration of a route map called "abc". The number indicates the route map entry (called the "instance") you are configuring. A route map can contain multiple entries. The software compares routes to the route map entries in ascending numerical order and stops the comparison once a match is found.

The **match** command in the route map matches on routes that have 5 for their metric value (cost). The **set** command changes the metric in routes that match the route map to 8.

The **redistribute static** command enables redistribution of static IP routes into OSPF, and uses route map "abc" to control the routes that are redistributed. In this example, the route map allows a static IP route to be redistributed into OSPF only if the route has a metric of 5, and changes the metric to 8 before placing the route into the OSPF route table.

The following command shows the result of the redistribution. Since only one of the static IP routes configured above matches the route map, only one route is redistributed. Notice that the route's metric is 5 before redistribution but is 8 after redistribution.

```
device# show ip ospf database external
Index Aging  LS ID           Router          Netmask  Metric    Flag
1     2      4.4.0.0         10.10.10.60     ffff0000 80000008  0000
```

Syntax: [no] redistribute { **bgp** | **connected** | **rip** | **isis** [**level-1** | **level-1-2** | **level-2**] | **static** [**route-map** *map-name*] }

The **bgp**, **connected**, **rip**, **isis**, and **static** parameters specify the route source.

The **route-map** *map-name* parameter specifies the route map name. The following match parameters are valid for OSPF redistribution:

- **match ip address** | **next-hop** *acl-num*
- **match metric** *num*
- **match tag** *tag-value*

NOTE
A match tag can take up to 16 tags. During the execution of a route-map a match on any tag value in the list is considered a successful match.

The following set parameters are valid for OSPF redistribution:

- **set ip next hop** *ip-addr*
- **set metric [+ | -]** *num* | **none**
- **set metric-type type-1 type-1** | **type-2**
- **set tag** *tag-value*

NOTE
You must configure the route map before you configure a redistribution that uses the route map.

NOTE
When you use a route map for route redistribution, the software disregards the permit or deny action of the route map.

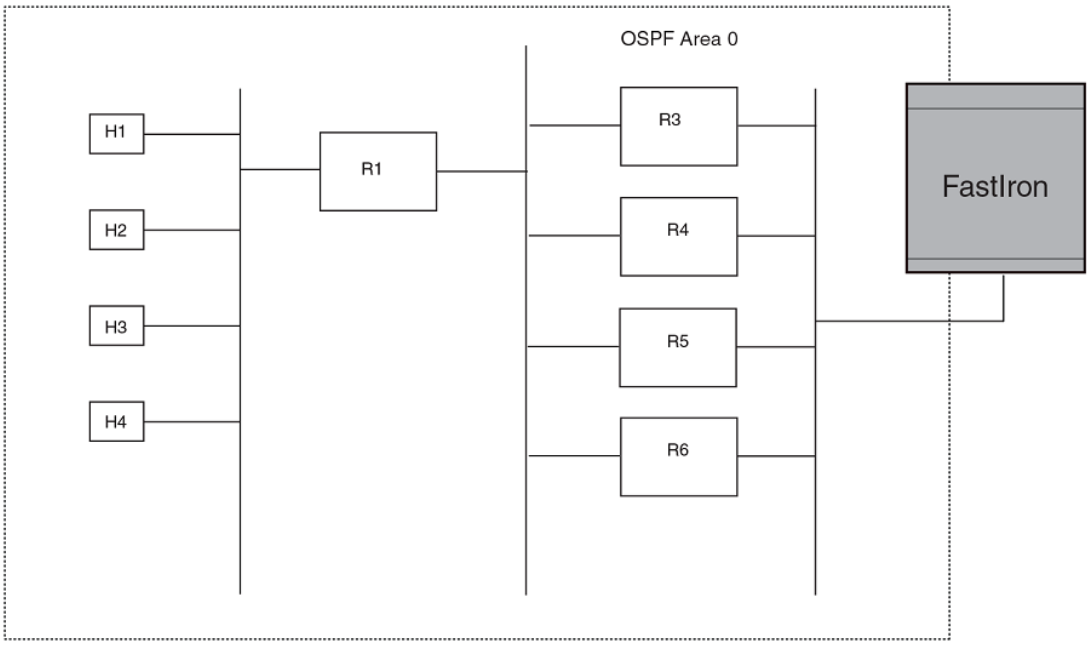
NOTE
For an external route that is redistributed into OSPF through a route map, the metric value of the route remains the same unless the metric is set by a **set metric** command inside the route map. The **default-metric** *num* command has no effect on the route. This behavior is different from a route that is

redistributed without using a route map. For a route redistributed without using a route map, the metric is set by the **default-metric** command.

Disable or re-enable load sharing

Brocade devices can load share among up to eight equal-cost IP routes to a destination. By default, IP load sharing is enabled. The default is 4 equal-cost paths but you can specify from 2 - 8 paths. The router software can use the route information it learns through OSPF to determine the paths and costs.

FIGURE 15 Example OSPF network with four equal-cost paths



The device has four paths to R1:

- Router ->R3
- Router ->R4
- Router ->R5
- Router ->R6

Normally, the device will choose the path to the R1 with the lower metric. For example, if the metric for R3 is 1400 and the metric for R4 is 600, the device will always choose R4.

However, suppose the metric is the same for all four routers in this example. If the costs are the same, the router now has four equal-cost paths to R1. To allow the router to load share among the equal cost routes, enable IP load sharing. The software supports four equal-cost OSPF paths by default when you enable load sharing. You can specify from 2 - 8 paths.

NOTE
The device is not source routing in these examples. The device is concerned only with the paths to the next-hop routers, not the entire paths to the destination hosts.

OSPF load sharing is enabled by default when IP load sharing is enabled.

Configure external route summarization

When the device is an OSPF Autonomous System Boundary Router (ASBR), you can configure it to advertise one external route as an aggregate for all redistributed routes that are covered by a specified address range.

When you configure an address range, the range takes effect immediately. All the imported routes are summarized according to the configured address range. Imported routes that have already been advertised and that fall within the range are flushed out of the AS and a single route corresponding to the range is advertised.

If a route that falls within a configured address range is imported by the device, no action is taken if the device has already advertised the aggregate route; otherwise the device advertises the aggregate route. If an imported route that falls with in a configured address range is removed by the device, no action is taken if there are other imported routes that fall with in the same address range; otherwise the aggregate route is flushed.

You can configure up to 32 address ranges. The device sets the forwarding address of the aggregate route to zero and sets the tag to zero.

If you delete an address range, the advertised aggregate route is flushed and all imported routes that fall within the range are advertised individually.

If an external LSDB overflow condition occurs, all aggregate routes are flushed out of the AS, along with other external routes. When the device exits the external LSDB overflow condition, all the imported routes are summarized according to the configured address ranges.

NOTE
If you use redistribution filters in addition to address ranges, the device applies the redistribution filters to routes first, then applies them to the address ranges.

NOTE
If you disable redistribution, all the aggregate routes are flushed, along with other imported routes.

NOTE
This option affects only imported, type 5 external routes. A single type 5 LSA is generated and flooded throughout the AS for multiple external routes. Type 7-route redistribution is not affected by this feature. All type 7 routes will be imported (if redistribution is enabled). To summarize type 7 LSAs or exported routes, use NSSA address range summarization.

To configure a summary address for OSPF routes, enter commands such as the following.

```
device(config-ospf-router)# summary-address 10.1.0.0 255.255.0.0
```

The command in this example configures summary address 10.1.0.0, which includes addresses 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. For all of these networks, only the address 10.1.0.0 is advertised in external LSAs.

Syntax: summary-address *ip-addr ip-mask*

The *ip-addr* parameter specifies the network address.

The *ip-mask* parameter specifies the network mask.

To display the configured summary addresses, enter the following command at any level of the CLI.

```
device# show ip ospf config
OSPF Redistribution Address Ranges currently defined:
Range-Address    Subnetmask
10.0.0.0          255.0.0.0
10.0.1.0          255.255.255.0
10.0.2.0          255.255.255.0
```

Syntax: show ip ospf config

Configure default route origination

When the device is an OSPF Autonomous System Boundary Router (ASBR), you can configure it to automatically generate a default external route into an OSPF routing domain. This feature is called "default route origination" or "default information origination".

By default, the device does not advertise the default route into the OSPF domain. If you want the device to advertise the OSPF default route, you must explicitly enable default route origination.

When you enable OSPF default route origination, the device advertises a type 5 default route that is flooded throughout the AS (except stub areas and NSSAs). In addition, internal NSSA ASBRs advertise their default routes as translatable type 7 default routes.

The device advertises the default route into OSPF even if OSPF route redistribution is not enabled, and even if the default route is learned through an IBGP neighbor.

NOTE
The device never advertises the OSPF default route, regardless of other configuration parameters, unless you explicitly enable default route origination using the following method.

If the device is an ASBR, you can use the "always" option when you enable the default route origination. The always option causes the ASBR to create and advertise a default route if it does not already have one configured.

If default route origination is enabled and you disable it, the default route originated by the device is flushed. Default routes generated by other OSPF routers are not affected. If you re-enable the feature, the feature takes effect immediately and thus does not require you to reload the software.

NOTE
The ABR (device) will not inject the default route into an NSSA by default and the command described in this section will not cause the device to inject the default route into the NSSA. To inject the default route into an NSSA, use the **area nssa default-information-originate** command.

To enable default route origination, enter the following command.

```
device(config-ospf-router)# default-information-originate
-ospf-router)# default-information-originate
```

To disable the feature, enter the following command.

```
device(config
-ospf-router)# no default-information-originate
```

Syntax: [no] default-information-originate [**always**] [**metric** *value*] [**metric-type** *type*]

The **always** parameter advertises the default route regardless of whether or not the router has a default route. This option is disabled by default.

NOTE
The **always** parameter is not necessary pre-FastIron 8.0 releases.

The **metric** *value* parameter specifies a metric for the default route. If this option is not used, the default metric is used for the route.

The **metric-type** *type* parameter specifies the external link type associated with the default route advertised into the OSPF routing domain. The *type* can be one of the following:

- type1 - Type 1 external route
- type2 - Type 2 external route

If you do not use this option, the default redistribution metric type is used for the route type.

The **route-map** parameter overrides other options. If **set** commands for **metric** and **metric-type** are specified in the route-map, the command-line values of metric and metric-type if specified, are “ignored” for clarification.

The **route-map** *rmap* parameter specifies the route map reference.

The corresponding route-map should be created before configuring the **route-map** option along with the **default-information-originate** command . If the corresponding route-map was not been created beforehand, then the an error message will be displayed stating that the route-map must be created.

NOTE
The route-map option cannot be used with a non-default address in the match conditions. The default-route LSA shall not be generated if a default route is not present in the routing table and a **match ip address** condition for an existing non-default route is configured in the route-map. The **match ip-address** command in the route-map is a no-op operation for the default information originate command.

Supported match and set conditions

The supported **match** and **set** conditions of a normal route-map configuration are as follows:

TABLE 15 Match Conditions

Match Conditions	
ip nexthop prefix-list	*prefixList*
ip nexthop	*accessList*
interface	*interfaceName*
metric	*metricValue*
tag	*routeTagValue*
protocol-type	*protocol route type and (or) sub-type value*
route-type	r*oute type (IS-IS sub-type values)*

TABLE 16 Set Conditions

Set Conditions:	
metric	*metricValue*
metric-type	*type1/type2*
tag	*routeTagValue*

OSPF non-stop routing

The graceful restart feature supported by open shortest path first (OSPF) maintains area topology and dataflow. Though the network requires neighboring routers to support graceful restart and perform hitless failover, the graceful restart feature may not be supported by all routers in the network. To eliminate this dependency, the non-stop routing (NSR) feature is supported on Brocade devices. NSR does not require support from neighboring routers to perform hitless failover. NSR does not support virtual link, so traffic loss is expected while performing hitless failover.

NSR does not require support from neighboring

routers to perform hitless failover.

If the active management module fails, the standby management module takes over and maintains the current OSPF routes, link-state advertisements (LSAs), and neighbor adjacencies, so that there is no loss of existing traffic to the OSPF destination.

Synchronization of critical OSPF elements

All types of LSAs and the neighbor information are synchronized to the standby module using the NSR synchronization library and IPC mechanism to transmit and receive packets.

Link state database synchronization

When the active management module fails, the standby management module takes over from the active management module with the identical OSPF link state database it had before the failure to ensure non-stop routing. The next shortest path first (SPF) run after switchover yields the same result in routes as the active module had before the failure and OSPF protocol requires that all routers in the network to have identical databases.

LSA delayed acknowledging

When an OSPF router receives LSAs from its neighbor, it acknowledges the LSAs. After the acknowledgement is received, the neighbor removes this router from its retransmission list and stops resending the LSAs.

In the case of NSR, the router fails after receiving the LSA from its neighbor and has acknowledged that neighbor upon receipt of an LSA, and the LSA synchronization to the standby module is completed. In this case, the standby module when taking over from the active module does not have that LSA in its database and the already acknowledged neighbor does not retransmit that LSA. For

this reason, the NSR-capable router waits for LSA synchronization of the standby module to complete (Sync-Ack) and then acknowledges the neighbor that sent the LSA.

LSA syncing and packing

When the LSA processing is completed on the active management module and the decision is made to install the LSA in its link state database (LSDB), OSPF synchronizes that LSA to the standby module. OSPF checks the current state of the database entry whether or not it is marked for deletion. After checking the database state, OSPF packs the LSA status and other necessary information needed for direct installation in the standby OSPF LSDB along with the LSA portion. When the LSA reaches the standby module, OSPF checks the database entry state in the buffer and takes appropriate action, such as adding, overwriting, updating, or deleting the LSA from the LSDB.

Neighbor router synchronization

When the neighbor router is added in the active management module, it is synchronized and added to the standby module. When the neighbor is deleted in the active module, it is synchronized to the standby and deleted in the standby. When the neighbor router state becomes 2WAY or FULL, the neighbor router is synchronized to the standby module.The following attributes of the neighbor router is synchronized to the standby module:

- Neighbor router id
- Neighbor router ip address
- Destination router or backup destination router information
- Neighbor state 2WAY or FULL
- MD5 information
- Neighbor priority

Limitations

- If a neighbor router is inactive for 30 seconds, and if the standby module takes over in another 10 seconds, the neighbor router cannot be dropped. The inactivity timer starts again and takes another 40 seconds to drop the neighbor router.
- In standby module, the valid neighbor states are LOADING, DOWN, 2WAY, and FULL. If the active management processor (MP) fails when the neighbor state is LOADING, the standby module cannot continue from LOADING, but the standby can continue from 2WAY and tries to establish adjacency between the neighboring routers.
- The minimum OSPF dead-interval timer value is 40 seconds (default dead-interval value). When the dead-interval value is configured less than this minimum value, OSPF NSR cannot be supported.

Interface synchronization

Interface information is synchronized for interfaces such as PTPT, broadcast, and non-broadcast. Interface wait time is not synchronized to the standby module. If an interface waits for 30 seconds to determine the identity of designated router (DR) or backup designated router (BDR), and if the standby module takes over, the wait timer starts again and takes another 40 seconds for the interface state to change from waiting to BDR, DR, or DROther.

BFD with OSPF NSR

Bidirectional forwarding detection (BFD) supports MP switchover and all BFD sessions for OSPF with graceful OSPF NSR, which are in the up state after the switchover. The BFD sessions for OSPF that do not use OSPF NSR are cleared before the switchover and then re-established on the new active MP after the MP switchover.

In case the active MP learns an OSPF neighbor and then restarts before a new BFD session is established, the standby module will not have a BFD session for the new OSPF neighbor. To overcome this and to support OSPF NSR with BFD, the following functions are supported when the active MP restarts:

- During MP switchover, BFD checks whether OSPF NSR is enabled. If OSPF NSR is enabled, the existing BFD sessions for OSPF is maintained during the switchover.
- OSPF sets up or clears the BFD sessions after OSPF neighbor transition.
- After the switchover, BFD sessions correspond with the active OSPF neighbor.

Standby module operations

The standby management module with OSPF configuration performs the following functions.

Neighbor database

Neighbor information is updated in the standby module based on updates from the active module. Certain neighbor state and interface transitions are synchronized to the standby module. By default, the neighbor timers on the standby module are disabled.

LSA database

The standby module processes LSA synchronization events from the active module and unpacks the LSA synchronization information to directly install it in its LSDB as the LSA has already been processed on the active module. The information required to install all types of LSAs (and special LSAs such as Grace LSAs) is packed by OSPF on the active module in the synchronization buffer, so that you can directly install LSAs on the standby module without extra processing.

The standby module is not allowed to originate any LSAs of its own. This is to maintain all information consistently from the active module. The active module synchronizes self-originated LSAs to the standby module.

LSA aging is not applicable on the standby module. During synchronization from the active, the current LSA age is recorded and the new database timestamp is created on the standby to later derive the LSA age as needed.

When the active module sends the LSAs to the standby module, based on the message, the standby module deletes or updates its link state database with the latest information.

LSA acknowledging or flooding are not done on the standby module. When the LSA synchronization update arrives from the active module, it will be directly installed into the LSDB.

Enabling and disabling NSR

To enable NSR for OSPF, enter the following commands:

```
device(config)# router ospf
device(config-ospf-router)# nonstop-routing
```

To disable NSR for OSPF, enter the following commands:

```
device(config)# router ospf
device(config-ospf-router)# no nonstop-routing
```

Syntax: [no] nonstop-routing

If you enter the **graceful-restart** command when NSR is already enabled, the command is rejected with the following message: “Error - Please disable NSR before enabling Graceful Restart.”

Similarly, if you enter the **nonstop-routing** command when graceful restart is already enabled, the command is rejected and the following message is displayed: “Error - Please disable Graceful Restart before enabling NSR.”

To disable **graceful-restart** command, the following commands:

```
Brocade (config)# router ospf
Brocade (config-ospf-router)# no graceful-restart
```

Limitations of NSR

Following are the limitations of NSR:

- Configurations that occur before the switchover are lost due to the CLI synchronization.
- Sham links are not supported.
- OSPF adjacency over GRE tunnels is not supported.
- Changes in the neighbor state or interface state before or during a switchover do not take effect.
- Traffic counters are not synchronized because the neighbor and LSA database counters are recalculated on the standby module during synchronization.
- LSA acknowledging is delayed because it has to wait until standby acknowledging occurs.
- Depending on the sequence of redistribution or new LSAs (from neighbors), the LSAs accepted within the limits of the database may change after switchover.
- In NSR hitless failover, after switchover, additional flooding-related protocol traffic is generated to the directly connected neighbors.
- OSPF startup timers, database overflow, and max-metric, are not applied during NSR switchover.
- Brocade NetIron devices may generate OSPF log messages or OSPF neighbor timers are reset but these issues do not cause any OSPF or traffic disruption.

Adding additional parameters

Previously, to add new parameters, the old configuration had to be undone and the newer configuration had to be recreated. In release 04.1.00 however, to add new parameters, the existing configuration need not be undone or removed. Any successive configuration changes with new parameters is

appended to the existing configuration. If the same parameter is entered again with a different value, then the corresponding parameter value is updated.

```
device(config-ospf-router)#default-information-originate route-map defaultToOspf

device(config-ospf-router)#default-information-originate always

device(config-ospf-router)#default-information-originate metric 200
```

In the above example, **default-information-originate** is enabled with the **route-map** parameter for the first CLI and the **always** and **metric** parameters are appended to the existing configuration. The running configuration of the above three split commands would be as follows:

```
device(config-ospf-router)#default-information-originate metric 200 route-map
defaultToOspf
```

Disabling configuration

To disable the **route-map** parameter from the configuration, enter the following command:

```
device(config-ospf-router)# no default-information-originate route-map defaultToOspf
```

The above CLI would retain the configuration with **default-information-originate** alone and the **route-map** option would get reset or removed.

The following commands with any or all of the options will remove the options from the **default-information-originate** command if any of the options are configured:

```
device(config-ospf-router)# no default-information-originate always

device(config-ospf-router)# no default-information-originate always route-map test

device(config-ospf-router)# no default-information-originate always route-map test
metric 200

device(config-ospf-router)# no default-information-originate always route-map test
metric 200 metric-type type1
```

In the following example, the parameters of the **default-information-originate** command are reset if they are configured and if none of the parameters are configured then, these commands will have no effect.

To disable the origination of default route, issue the command with **no** option and without any other options. This would remove the configuration of the **default information origination** even if any of the above mentioned options are configured.

Syntax: [no] default-information-originate [**always**] [**metric** *metricvalue*] [**metric-type** *metric-type*] [**route-map** *rmap-name*]

The **always** parameter advertises the default route regardless of whether the router has a default route. This option is disabled by default.

The **metric** *value* parameter specifies a metric for the default route. If this option is not used, the default metric is used for the route.

The **metric-type** *type* parameter specifies the external link type associated with the default route advertised into the OSPF routing domain. The *type* can be one of the following:

- type1 - Type 1 external route
- type2 - Type 2 external route

If you do not use this option, the default redistribution metric type is used for the route type.

NOTE
If you specify a metric and metric type, the values you specify are used even if you do not use the **always** option.

The **route-map** parameter overrides other options. If **set** commands for **metric** and *metric-type* are specified in the route-map, the command-line values of metric and metric-type if specified, are ignored for clarification.

The **route-map** *rmap* parameter specifies the route map reference.

The corresponding route-map should be created before configuring the **route-map** option along with the **default-information-originate**. If the corresponding route-map was not been created beforehand, then the an error message will be displayed stating that the route-map must be created.

OSPF distribute list

This feature configures a distribution list to explicitly deny specific routes from being eligible for installation in the IP route table. By default, all OSPF routes in the OSPF route table are eligible for installation in the IP route table. This feature does not block receipt of LSAs for the denied routes. The device still receives the routes and installs them in the OSPF database. The feature only prevents the software from installing the denied OSPF routes into the IP route table.

The OSPF distribution list can be managed using ACLs or Route Maps to identify routes to be denied as described in the following sections:

- Configuring an OSPF Distribution List using ACLs
- Configuring an OSPF Distribution List using Route Maps

Configuring an OSPF distribution list using ACLs

To configure an OSPF distribution list using ACLs:

- Configure an ACL that identifies the routes you want to deny. Using a standard ACL lets you deny routes based on the destination network, but does not filter based on the network mask. To also filter based on the destination network's network mask, use an extended ACL.
- Configure an OSPF distribution list that uses the ACL as input.

Examples

In the following example, the first three commands configure a standard ACL that denies routes to any 10.x.x.x destination network and allows all other routes for eligibility to be installed in the IP route table. The last three commands change the CLI to the OSPF configuration level and configure an OSPF distribution list that uses the ACL as input. The distribution list prevents routes to any 10.x.x.x destination network from entering the IP route table. The distribution list does not prevent the routes from entering the OSPF database.

```
device(config)# ip access-list standard no_ip
device(config-std-nacl)# deny 10.0.0.0 0.255.255.255
device(config-std-nacl)# permit any
```

```
device(config)# router ospf
device(config-ospf-router) # area 0
device(config-ospf-router) # distribute-list no_ip in
```

In the following example, the first three commands configure an extended ACL that denies routes to any 10.31.39.x destination network and allows all other routes for eligibility to be installed in the IP route table. The last three commands change the CLI to the OSPF configuration level and configure an OSPF distribution list that uses the ACL as input. The distribution list prevents routes to any 10.31.39.x destination network from entering the IP route table. The distribution list does not prevent the routes from entering the OSPF database.

```
device(config)# ip access-list extended DenyNet39
device(config-ext-nacl)# deny ip 10.31.39.0 0.0.0.255 any
device(config-ext-nacl)# permit ip any any
device(config)# router ospf
device(config-ospf-router) # area 0
device(config-ospf-router) # distribute-list DenyNet39 in
```

In the following example, the first command configures a numbered ACL that denies routes to any 10.31.39.x destination network and allows all other routes for eligibility to be installed in the IP route table. The last three commands change the CLI to the OSPF configuration level and configure an OSPF distribution list that uses the ACL as input. The distribution list prevents routes to any 10.31.39.x destination network from entering the IP route table. The distribution list does not prevent the routes from entering the OSPF database.

```
device(config)# ip access-list 100 deny ip 10.31.39.0 0.0.0.255 any
device(config)# ip access-list 100 permit ip any any
device(config)# router ospf
device(config-ospf-router) # area 0
device(config-ospf-router) # distribute-list 100 in
```

Syntax: [no] distribute-list { *acl-name* | *acl-number* } **in**

The **distribute-list** command is applied globally to all interfaces on the router where it is executed.

Configuring an OSPF distribution list using route maps

You can manage an OSPF distribution list using route maps that apply match operations as defined by an ACL or an IP prefix list. You can also use other options available within the route maps and ACLs to further control the contents of the routes that OSPF provides to the IP route table. This section describes an example of an OSPF distribution list using a route map to specify an OSPF administrative distance for routes identified by an IP prefix list.

To configure an OSPF distribution list using route maps:

- Configure a route map that identifies the routes you want to manage
- Optionally configure an OSPF administrative distance to apply to the OSPF routes
- Configure an OSPF distribution list that uses the route map as input

In the following example, the first two commands identify two routes using the **ip prefix-list test1** command. Next, a route map is created that uses the **prefix-list test1** command to identify the two routes and the **set distance** command to set the OSPF administrative distance of those routes to 200. A distribution list is then configured under the OSPF configuration that uses the route map named “setdistance” as input.

```
device(config)# ip prefix-list test1 seq 5 permit 10.100.1.0/24
device(config)# ip prefix-list test1 seq 10 permit 10.100.2.0/24
device(config)# route-map setdistance permit 1
device(config-routemap setdistance)# match ip address prefix-list test1
device(config-routemap setdistance)# set distance 200
device(config-routemap setdistance)# exit
device(config)# route-map setdistance permit 2
device(config-routemap setdistance)# exit
device(config)# router ospf
```

```
device(config-ospf-router)# area 0
device(config-ospf-router)# area 1
device(config-ospf-router)# distribute-list route-map setdistance in
device(config-ospf-router)# exit
```

Once this configuration is implemented, the routes identified by the **ip prefix-list** command and matched in the route map will have their OSPF administrative distance to 200. This is displayed in the output from the **show ip route** command, as shown in the following.

```
device# show ip route
Total number of IP routes: 4
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
      Destination        Gateway     Port       Cost          Type
1     10.1.1.0/24        DIRECT      eth 1/1    0/0                    D
2     10.100.1.0/24      10.1.1.1    eth 1/1    200/2       O
3     10.100.2.0/24      10.1.1.1    eth 1/1    200/10      02
4     10.100.6.0/24      10.1.1.1    eth 1/1    110/2       O
```

Routes 1 and 2 demonstrate the actions of the example configuration as both display an OSPF administrative distance value of 200. Note that the value is applied to both OSPF learned routes that match the route-map instance containing the set distance clause. The other OSPF route (route 3), which does not match the relevant instance, continues to have the default OSPF administrative distance of 110.

The following is an example of the **distribute-list** command applied with route-map “setdistance” set as the input.

```
device(config-ospf-router)# distribute-list route-map setdistance in
```

Syntax: [no] distribute-list route-map *routemap-name* **in**

The *routemap-name* variable specifies the name of the route map being used to define the OSPF distribute list.

The *distribute-list* command is applied to all OSPF LSAs on the router where it is executed.

NOTE
A route map used with the **distribute-list** command can use either the **ip prefix-list** command (as shown in the example) or an ACL to define the routes.

The **set distance** command is used in association with a route map configuration.

Modify SPF timers

The device uses the following timers when calculating the shortest path for OSPF routes:

- **SPF delay** - When the device receives a topology change, the software waits before it starts a Shortest Path First (SPF) calculation. By default, the software waits 0 (zero) seconds. You can configure the SPF delay to a value from 0 - 65535 seconds. If you set the SPF delay to 0 seconds, the software immediately begins the SPF calculation after receiving a topology change.
- SPF hold time - The device waits for a specific amount of time between consecutive SPF calculations. By default, the device waits zero seconds. You can configure the SPF hold time to a value from 0 - 65535 seconds. If you set the SPF hold time to 0 seconds, the software does not wait between consecutive SPF calculations.

You can set the delay and hold time to lower values to cause the device to change to alternate paths more quickly in the event of a route failure. Note that lower values require more CPU processing time.

You can change one or both of the timers.

To change the SPF delay and hold time, enter commands such as the following.

```
device(config-ospf-router)# timers spf 10 20
```

The command in this example changes the SPF delay to 10 seconds and changes the SPF hold time to 20 seconds.

To set the timers back to their default values, enter a command such as the following.

```
device(config-ospf-router)# no timers spf 10 20
```

Syntax: [no] timers spf *delay hold-time*

The *delay* parameter specifies the SPF delay.

The *hold-time* parameter specifies the SPF hold time.

NOTE
OSPF incrementally updates the OSPF routing table when new Type-3 or Type-4 Summary, Type-5 External, or Type-7 External NSSA LSAs are received.

Modify redistribution metric type

The redistribution metric type is used by default for all routes imported into OSPF unless you specify different metrics for individual routes using redistribution filters. Type 2 specifies a big metric (three bytes). Type 1 specifies a small metric (two bytes). The default value is type 2.

To modify the default value to type 1, enter the following command.

```
device(config-ospf-router)# metric-type type1
```

Syntax: [no] metric-type type1 | **type2**

The default is **type2**.

Modify administrative distance

The device can learn about networks from various protocols, including Border Gateway Protocol version 4 (BGP4), RIP, IS-IS, and OSPF. Consequently, the routes to a network may differ depending on the protocol from which the routes were learned. The default administrative distance for OSPF routes is 110.

The router selects one route over another based on the source of the route information. To do so, the router can use the administrative distances assigned to the sources. You can bias the decision the device makes by changing the default administrative distance for OSPF routes.

Configuring administrative distance based on route type

You can configure a unique administrative distance for each type of OSPF route. For example, you can use this feature to prefer a static route over an OSPF inter-area route but you also want to prefer OSPF intra-area routes to static routes.

The distance you specify influences the choice of routes when the device has multiple routes for the same network from different protocols. The device prefers the route with the lower administrative distance.

You can specify unique default administrative distances for the following route types:

- Intra-area routes
- Inter-area routes
- External routes

The default for all these OSPF route types is 110.

NOTE
This feature does not influence the choice of routes within OSPF. For example, an OSPF intra-area route is always preferred over an OSPF inter-area route, even if the intra-area route's distance is greater than the inter-area route's distance.

To change the default administrative distances for inter-area routes, intra-area routes, and external routes, enter the following command.

```
device(config-ospf-router)# distance external 100
device(config-ospf-router)# distance inter-area 90
device(config-ospf-router)# distance intra-area 80
```

Syntax: [no] distance { **external** | **inter-area** | **intra-area** } *distance*

The **distance external**, **inter-area** , and **intra-area** parameters specify the route type for which you are changing the default administrative distance.

The *distance* parameter specifies the new distance for the specified route type. Unless you change the distance for one of the route types using commands such as those shown above, the default is 110.

To reset the administrative distance to its system default (110), enter a command such as the following.

```
device(config-ospf-router)# no distance external 100
```

Configure OSPF group Link State Advertisement (LSA) pacing

The device paces LSA refreshes by delaying the refreshes for a specified time interval instead of performing a refresh each time an individual LSA refresh timer expires. The accumulated LSAs constitute a group, which the device refreshes and sends out together in one or more packets.

The pacing interval, which is the interval at which the device refreshes an accumulated group of LSAs, is configurable to a range from 10 - 1800 seconds (30 minutes). The default is 240 seconds (four minutes). Thus, every four minutes, the device refreshes the group of accumulated LSAs and sends the group together in the same packets.

Usage guidelines

The pacing interval is inversely proportional to the number of LSAs the device is refreshing and aging. For example, if you have approximately 10,000 LSAs, decreasing the pacing interval enhances performance. If you have a very small database (40 - 100 LSAs), increasing the pacing interval to 10 - 20 minutes might enhance performance slightly.

Changing the LSA pacing interval

To change the LSA pacing interval, use the following CLI method.

To change the LSA pacing interval to two minutes (120 seconds), enter the following command.

```
device(config-ospf-router)# timers lsa-group-pacing 120
```

Syntax: [no] timers lsa-group-pacing *secs*

The *secs* parameter specifies the number of seconds and can be from 10 - 1800 (30 minutes). The default is 240 seconds (four minutes).

To restore the pacing interval to its default value, enter the following command.

```
device(config-ospf-router)# no timers lsa-group-pacing
```

OSPF ABR type 3 LSA filtering

OSPF ABR Type 3 LSA filtering increases the ability of an ABR that is running the OSPF protocol to filter type 3 link-state advertisements (LSAs) that are sent between different OSPF areas. Only packets with specified prefixes will be sent from one area to another area and all packets with other prefixes will be prohibited.

Type 3 LSAs refer to summary links and are sent by ABRs to advertise destinations outside the area. OSPF ABR Type 3 LSA filtering gives the administrator improved control of route distribution between OSPF areas.

Usage and configuration guidelines

- The “area prefix-list” command is only applicable to the ABRs. If the router is not an ABR the configuration is accepted however it will start working only after router is made ABR.
- With this feature enabled in the “in” direction, all type 3 LSAs originated by the ABR to this area, based on information from all other areas, are filtered by the prefix list. Type 3 LSAs that were originated as a result of the area range command in another area are treated like any other type 3 LSA that was originated individually. Any prefix that does not match an entry in the prefix list is implicitly denied
- With this feature enabled in the “out” direction, all type 3 LSAs advertised by the ABR, based on information from this area to all other areas, are filtered by the prefix list. If the area range command has been configured for this area, Type 3 LSAs that corresponds to the area range command are treated like any other type 3 LSA.
- Prefixes that are not permitted by the prefix list are implicitly denied

TABLE 17 Behavior for prefix list configurations

IP prefix list	OSPF area prefix list	Event	Filtering done
XXX	Not defined	None	No (permit all)
Not defined	Defined	None	Yes (deny all)
Not defined	Defined	IP prefix list defined	Recalculation
Defined (no rules configured)	Defined	None	Implicit deny (deny all)
Defined (rules configured)	Defined	IP prefix list deleted	Recalculation and deny all
Defined (rules configured)	Defined	IP prefix list rule added or modified or deleted	Recalculation
Defined (rules configured)	Defined	Area prefix list deleted	Recalculation and permit all

Configuring an OSPF area prefix list

To filter prefixes advertised in type 3 link-state advertisements (LSAs) between (OSPF) areas of an Area Border Router (ABR), use the **area prefix-list** command in router configuration mode. To change or cancel the filter, use the no form of this command.

Configuring OSPF ABR type 3 LSA filtering

To filter inter-area routes into a specified area, use the following commands beginning in router configuration mode.

To configure the router to run an OSPF process, enter commands such as the following.

```
device(config)# router ospf
device(config-ospf-router)#
```

To filter prefixes advertised in type 3 link-state advertisements (LSAs) between (OSPF) areas of an Area Border Router (ABR), use the area prefix-list command in router configuration mode. To change or cancel the filter, use the **no** form of this command.

```
device(config-ospf-router)# area 1 prefix-list area in
```

To configure the switch to filter inter-area routes out of the specified area, enter a command such as the following.

```
device(config-ospf-router)# area 10.10.10.1 prefix-list Routesfor20 out
```

Syntax: [no] area { *area-id* | *area_ip* } **prefix-list** *prefix-list-name* { **in** | **out** }

The **area**, *area-id*, and *area_ip* parameters specify the area id in different formats.

The **in** keyword specifies that prefix list is applied to prefixes advertised to the specified area from other areas.

The **out** keyword specifies that prefix list is applied to prefixes advertised out of the specified area to other areas.

Defining and applying IP prefix lists

An IP prefix list specifies a list of networks. When you apply an IP prefix list to an area, the Brocade device sends or receives only a route whose destination is in the IP prefix list. The software interprets the prefix lists in order, beginning with the lowest sequence number.

To configure an IP prefix list and apply it to an area, enter commands such as the following.

```
device(config)# ip prefix-list Routesfor20 permit 20.20.0.0/24
device(config)# router ospf
device(config-ospf-router)# area 10.10.10.1 prefix-list Routesfor20 out
```

These commands configure an IP prefix list named Routesfor20, which permits routes to network 20.20.0.0/24. The **area** command configures the device to use IP prefix list Routesfor20 to determine which routes to send to area 10.10.10.1. The device sends routes that go to 20.20.x.x to area 10.10.10.1 because the IP prefix list explicitly permits these routes to be sent to the area.

Syntax: ip prefix-list *name* [**seq** *seq-value*] [**description** *string*] { **deny** | **permit** } *network-addr/mask-bits* [**ge** *ge-value*] [**le** *le-value*]

The *name* variable specifies the prefix list name. You use this name when applying the prefix list to a neighbor.

The **seq** *seq-value* parameter is optional and specifies the IP prefix list's sequence number. If you do not specify a sequence number, the software numbers them in increments of 5, beginning with prefix list entry 5. The software interprets the prefix list entries in numerical order, beginning with the lowest sequence number.

The **description** *string* parameter is a text string describing the prefix list.

The **deny** or **permit** parameters specify the action the software takes if a neighbor's route is in this prefix list.

The prefix-list matches only on this network unless you use the **ge** *ge-value* or **le** *le-value* parameters. (See below.)

The *network-addr/mask-bits* variable specifies the network number and the number of bits in the network mask.

You can specify a range of prefix length for prefixes that are more specific than *network-addr/mask-bits*.

- If you specify only **ge** *ge-value*, then the mask-length range is from **ge** *ge-value* to 32.
- If you specify only **le** *le-value*, then the mask-length range is from length to *le-value*.
- The *ge-value* or *le-value* you specify must meet the following condition.
  - length < *ge-value* <= *le-value* <= 32

If you do not specify **ge** *ge-value* or **le** *le-value*, the prefix list matches only on the exact network prefix you specify with the network-addr/mask-bits parameter.

Displaying the configured OSPF area prefix list

To display the prefix-lists attached to the areas, enter the following command.

To change the LSA pacing interval to two minutes (120 seconds), enter the following command.

```
device(config)# show ip ospf config
Router OSPF: Enabled
Graceful Restart: Disabled, timer 120
Graceful Restart Helper: Enabled
Redistribution: Disabled
Default OSPF Metric: 10
OSPF Auto-cost Reference Bandwidth: Disabled
OSPF Redistribution Metric: Type2
OSPF External LSA Limit: 14447047
OSPF Database Overflow Interval: 0
RFC 1583 Compatibility: Enabled
Router id: 10.5.5.1
Interface State Change Trap:                    Enabled
Virtual Interface State Change Trap:            Enabled
Neighbor State Change Trap:                     Enabled
Virtual Neighbor State Change Trap:             Enabled
Interface Configuration Error Trap:             Enabled
Virtual Interface Configuration Error Trap:     Enabled
Interface Authentication Failure Trap:          Enabled
Virtual Interface Authentication Failure Trap: Enabled
Interface Receive Bad Packet Trap:              Enabled
Virtual Interface Receive Bad Packet Trap:      Enabled
Interface Retransmit Packet Trap:               Disabled
Virtual Interface Retransmit Packet Trap:       Disabled
Originate LSA Trap:                             Disabled
Originate MaxAge LSA Trap:                      Disabled
Link State Database Overflow Trap:              Disabled
Link State Database Approaching Overflow Trap: Disabled
OSPF Area currently defined:
Area-ID    Area-Type  Cost      Prefix List In      Prefix List Out
0             normal  0
1             normal  0         Area_1_Pfx_list in  Area_1_Pfx_List_Out
```

Syntax: show ip ospf config

Displaying the configured IP prefix list

To only display the configured ip prefix-list, enter a command such as the following.

```
device# show ip prefix-lists
ip prefix-list abc: 2 entries
seq 5 deny 2.3.4.0/24
seq 10 permit 4.5.0.0/16.0
```

Syntax: show ip prefix-lists *prefix-list-name*

The *prefix-list-name* specifies the name of the prefix list. You use this name when applying the prefix list to an area.

Modify OSPF traps generated

OSPF traps as defined by RFC 1850 are supported on device.

You can disable all or specific OSPF trap generation by entering the following CLI command.

```
device(config)# no snmp-server trap ospf
```

To later re-enable the trap feature, enter **snmp-server trap ospf** .

To disable a specific OSPF trap, enter the command as **no snmp-server trap ospf** *ospf-trap*.

These commands are at the OSPF router Level of the CLI.

Here is a summary of OSPF traps supported on device, their corresponding CLI commands, and their associated MIB objects from RFC 1850. The first list are traps enabled by default:

- **interface-state-change-trap** - [MIB object: OspfIfstateChange]
- **virtual-interface-state-change-trap** - [MIB object: OspfVirtIfStateChange
- **neighbor-state-change-trap** - [MIB object:ospfNbrStateChange]
- **virtual-neighbor-state-change-trap** - [MIB object: ospfVirtNbrStateChange]
- **interface-config-error-trap** - [MIB object: ospfIfConfigError]
- **virtual-interface-config-error-trap** - [MIB object: ospfVirtIfConfigError]
- **interface-authentication-failure-trap** - [MIB object: ospfIfAuthFailure]
- **virtual-interface-authentication-failure-trap** - [MIB object: ospfVirtIfAuthFailure]
- **interface-receive-bad-packet-trap** - [MIB object: ospfIfrxBadPacket]
- **virtual-interface-receive-bad-packet-trap** - [MIB object: ospfVirtIfRxBadPacket]

The following traps are disabled by default.

- **interface-retransmit-packet-trap** - [MIB object: ospfTxRetransmit]
- **virtual-interface-retransmit-packet-trap** - [MIB object: ospfVirtIfTxRetransmit]
- **originate-lsa-trap** - [MIB object: ospfOriginateLsa]
- **originate-maxage-lsa-trap** - [MIB object: ospfMaxAgeLsa]
- **link-state-database-overflow-trap** - [MIB object: ospfLsdbOverflow]
- **link-state-database-approaching-overflow-trap** - [MIB object: ospfLsdbApproachingOverflow

To stop an OSPF trap from being collected, use the CLI command: **no trap** *ospf-trap* at the Router OSPF level of the CLI. To disable reporting of the neighbor-state-change-trap, enter the following command.

```
device(config-ospf-router)# no trap neighbor-state-change-trap
```

To reinstate the trap, enter the following command.

```
device(config-ospf-router)# trap neighbor-state-change-trap
```

Syntax: [no] trap ospf-trap

Modify OSPF standard compliance setting

The device is configured, by default, to be compliant with the RFC 1583 OSPF V2 specification.

NOTE
In the current implementation, the NetIron devices are not compliant with RFC3509.

To configure a router to operate with the latest OSPF standard, RFC 2328, enter the following commands.

```
device(config)# router ospf
device(config-ospf-router)# no rfc1583-compatibility
```

Syntax: [no] rfc1583-compatibility

Modify exit overflow interval

If a database overflow condition occurs on a router, the router eliminates the condition by removing entries that originated on the router. The exit overflow interval allows you to set how often a device checks to see if the overflow condition has been eliminated. The default value is 0. The range is 0 - 86400 seconds (24 hours). If the configured value of the database overflow interval is zero, then the router never leaves the database overflow condition.

To modify the exit overflow interval to 60 seconds, enter the following command.

```
device(config-ospf-router)# database-overflow-interval 60
```

Syntax: [no] database-overflow-interval *value*

The *value* can be from 0 - 86400 seconds. The default is 0 seconds.

Specify types of OSPF Syslog messages to log

You can specify which kinds of OSPF-related Syslog messages are logged. By default, the only OSPF messages that are logged are those indicating possible system errors. If you want other kinds of OSPF messages to be logged, you can configure the device to log them.

For example, to specify that all OSPF-related Syslog messages be logged, enter the following commands.

```
device(config)# router ospf
device(config-ospf-router)# log all
```

Syntax: [no] log { **all** | **adjacency** [**dr-only**] | **bad_packet** [**checksum**] | **database** | **memory** | **retransmit** }

The **log** command has the following options:

The **all** option causes all OSPF-related Syslog messages to be logged. If you later disable this option with the **no log all** command, the OSPF logging options return to their default settings.

The **adjacency** option logs essential OSPF neighbor state changes, especially on error cases. This option is disabled by default. The **dr-only** sub-option only logs essential OSPF neighbor state changes where the interface state is designated router (DR).

NOTE
For interfaces where the designated router state is not applicable, such as point-to-point and virtual links, OSPF neighbor state changes will always be logged irrespective of the setting of the **dr-only** sub-option.

NOTE
A limitation with the **dr-only** sub-option is that when a DR/BDR election is underway, OSPF neighbor state changes pertaining to non-DR/BDR routers are not logged. Logging resumes once a DR is elected on that network.

The **bad_packet checksum** option logs all OSPF packets that have checksum errors. This option is enabled by default.

The **bad_packet** option logs all other bad OSPF packets. This option is disabled by default.

The **database** option logs OSPF LSA-related information. This option is disabled by default.

The **memory** option logs abnormal OSPF memory usage. This option is enabled by default.

The **retransmit** option logs OSPF retransmission activities. This option is disabled by default.

Configuring an OSPF network type

To configure an OSPF network, enter commands such as the following.

```
device(config)# interface eth 1/5
device(config-if-1/5)# ip ospf network point-to-point
```

This command configures an OSPF point-to-point link on Interface 5 in slot 1.

Syntax: [no] ip ospf network { **point-to-point** | **broadcast** | **non-broadcast** }

The **point-to-point** option configures the network type as a point to point connection.

NOTE
Brocade devices support numbered point-to-point networks, meaning the OSPF router must have an IP interface address which uniquely identifies the router over the network. Brocade devices do not support unnumbered point-to-point networks.

The **broadcast** option configures the network type as a broadcast connection. This is the default option for Ethernet, VE and Loopback interfaces.

The **non-broadcast** option configures the network type as a non-broadcast connection. This allows you to configure the interface to send OSPF traffic to its neighbor as unicast packets rather than multicast packets. This can be useful in situations where multicast traffic is not feasible (for example when a firewall does not allow multicast packets).

On a non-broadcast interface, the routers at either end of this interface must configure non-broadcast interface type and the neighbor IP address. There is no restriction on the number of routers sharing a non-broadcast interface (for example, through a hub/switch).

To configure an OSPF interface as a non-broadcast interface, you enable the feature on a physical interface or a VE, following the **ip ospf area** statement, and then specify the IP address of the neighbor in the OSPF configuration. The non-broadcast interface configuration must be done on the OSPF routers at either end of the link.

For example, the following commands configure VE 20 as a non-broadcast interface.

```
device(config)# int ve 20
device(config-vif-20)# ip address 10.1.20.4/24
device(config-vif-20)# ip ospf area 0
device(config-vif-20)# ip ospf network non-broadcast
```

The following commands specify 10.1.20.1 as an OSPF neighbor address. The address specified must be in the same sub-net as the non-broadcast interface.

```
device(config)# router ospf
device(config-ospf-router)# neighbor 10.1.20.1
```

For example, to configure the feature in a network with three routers connected by a hub or switch, each router must have the linking interface configured as a non-broadcast interface, and the two other routers must be specified as neighbors.

Configuring OSPF Graceful Restart

OSPF Graceful Restart can be enabled in the following configurations:

- **Configuring OSPF Graceful Restart for the Global Instance** - In this configuration all OSPF neighbors other than those used by VRFs are made subject to the Graceful Restart capability. The restart timer set globally does not apply to Graceful Restart on a configured VRF.
- **Configuring OSPF Graceful Restart per VRF** - In this configuration all OSPF neighbors for the specified VRF are made subject to the Graceful Restart capability. The restart timer set for a specific VRF only applies to that VRF.

Configuring OSPF Graceful Restart for the global instance

OSPF Graceful restart can be configured for the global instance or for a specified Virtual Routing and Forwarding (VRF) instance. Configuring OSPF Graceful restart for the global instance does not configure it for any VRFs. The following sections describe how to enable the OSPF graceful restart feature for the global instance on a device.

Use the following command to enable the graceful restart feature for the global instance on a device.

```
device(config)# router ospf
device(config-ospf-router)# graceful-restart
```

Syntax: [no] graceful-restart

Configuring OSPF Graceful Restart time for the global instance

Use the following command to specify the maximum amount of time advertised to a neighbor router to maintain routes from and forward traffic to a restarting router.

```
device(config)# router ospf
device(config-ospf-router)# graceful-restart restart-time 120
```

Syntax: [no] graceful-restart restart-time *seconds*

The *seconds* variable sets the maximum restart wait time advertised to neighbors.

Possible values are 10 - 1800 seconds.

The default value is 120 seconds.

Disabling OSPF Graceful Restart helper mode for the global instance

By default, a router supports other restarting routers as a helper. You can prevent your router from participating in OSPF Graceful Restart by using the following command.

```
device(config)# router ospf
device(config-ospf-router)# graceful-restart helper-disable
```

Syntax: [no] graceful-restart helper-disable

This command disables OSPF Graceful Restart helper mode.

The default behavior is to help the restarting neighbors.

Configuring OSPF Graceful Restart per VRF

The following sections describe how to enable the OSPF Graceful Restart feature on a specified VRF.

Use the following command to enable the graceful restart feature on a specified VRF.

```
device(config)# router ospf vrf blue
device(config-ospf-router)# graceful-restart
```

Syntax: [no] graceful-restart

Configuring OSPF Graceful Restart time per VRF

Use the following command to specify the maximum amount of time advertised to an OSPF neighbor router to maintain routes from and forward traffic to a restarting router.

```
device(config)# router ospf vrf blue
device(config-ospf-router)# graceful-restart restart-time 120
```

Syntax: [no] graceful-restart restart-time *seconds*

The *seconds* variable sets the maximum restart wait time advertised to OSPF neighbors of the VRF.

Possible values are 10 - 1200 seconds.

The default value is 60 seconds.

Disabling OSPF Graceful Restart helper mode per VRF

You can prevent your router from participating in OSPF Graceful Restart with VRF neighbors by using the following command.

```
device(config)# router ospf vrf blue
device(config-ospf-router)# graceful-restart helper-disable
```

Syntax: [no] graceful-restart helper-disable

This command disables OSPF Graceful Restart helper mode.

The default behavior is to help the restarting neighbors.

Configuring OSPF router advertisement

You can configure OSPF router advertisement in the **router ospf** mode or**router ospf vrf** mode as shown in the following examples.

```
device(config)# router ospf
device(config-ospf-router)# max-metric router-lsa all-vrfs on-startup 30 link all
device(config)# router ospf vrf blue
device(config-ospf-router)# max-metric router-lsa on-startup 30 link all
```

Syntax: [no] max-metric router-lsa [**all-vrfs**] [**on-startup** { *time* | **wait-for-bgp** }] [**summary-lsa** *metric-value*] [**external-lsa** *metric-value*] [**te-lsa** *metric-value*] [**all-lsas**] [**link** { **ptp** | **stub** | **transit** | **all** }]

The **all-vrfs** parameter specifies that the command will be applied to all VRF instances of OSPFv2.

NOTE
This command is supported only for VRFs that are already configured when the **max-metric router-lsa all-vrfs** command is issued.

Any new OSPF instance configured after the **max-metric** configuration is completed requires that the **max-metric** command be configured again to take in the new OSPF instance.

The **on-startup** parameter specifies that the OSPF router advertisement be performed at the next system startup. This is an optional parameter.

When using the **on-startup** option you can set a *time* in seconds for which the specified links in Router LSA will be advertised with the metric set to a maximum value of 0xFFFF. Optional values for *time* are 5 to 86400 seconds. There is no default value for *time*.

The **wait-for-bgp** option for the **on-startup** parameter directs OSPF to wait for either 600 seconds or until BGP has finished route table convergence (whichever event happens first), before advertising the links with the normal metric.

Using the **link** parameter you can specify the type of links for which the maximum metric is to be advertised. The default value is for maximum metric to be advertised for transit links only. This is an optional parameter.

Additional options are supported that allow you to select the following LSA types and set the required metric:

The **summary-lsa** option specifies that the metric for all summary type 3 and type 4 LSAs will be modified to the specified *metric-value* or the default value. The range of possible values for the *metric-value* variable are 1 to 16777214 (Hex: 0x00001 to 0x00FFFFFE). The default value is 16711680 (Hex: 0x00FF0000).

The **external-lsa** option specifies that the metric for all external type 5 and type 7 LSAs will be modified to the specified *metric-value* or a default value. The range of possible values for the *metric-value* variable are 1 to 16777214 (Hex: 0x00001 to 0x00FFFFFE). The default value is 16711680 (Hex: 0x00FF0000).

The **te-lsa** option specifies that the TE metric field in the TE metric sub tlv for all type 10 Opaque LSAs LINK TLV originated by the router will be modified to the specified *metric-value* or a default value. The range of possible values for the *metric-value* variable are 1 to 4294967295 (Hex: 0x00001 to 0xFFFFFFFF). The default value is 4294967295 (Hex: 0xFFFFFFFF). This parameter only applies to the default instance of OSPF.

Examples

The following examples of the command max-metric router-lsa command demonstrate how it can be used:

The following command indicates that OSPF is being shutdown and that all links in the router LSA should be advertised with the value 0xFFFF and the metric value for all external and summary LSAs is set to 0xFF0000 until OSPF is restarted. This configuration will not be saved.

```
device(config)# router ospf
device(config-ospf-router)# max-metric router-lsa external-lsa summary-lsa link all
```

The following command indicates that OSPF is being shutdown and that all links in the router LSA should be advertised with the value 0xFFFF and the metric value for all external and summary LSAs should be set to 0xFF0000 until OSPF is restarted. Also, if OSPF TE is enabled then all LINK TLVs advertised by the router in Opaque LSAs should be updated with the TE Metric set to 0xFFFFFFFF and the available bandwidth set to 0. This configuration will not be saved.

```
device(config)# router ospf
device(config-ospf-router)# max-metric router-lsa all-lsas link all
```

The following command indicates that OSPF is being shutdown and that all links in the router LSA should be advertised with the value 0xFFFF and the metric value for all summary LSAs should be set to 0xFFFFFE until OSPF is restarted. This configuration will not be saved.

```
device(config)# router ospf
device(config-ospf-router)# max-metric router-lsa summary-lsa 16777214 link all
```

The following command turns off the advertisement of special metric values in all Router, Summary, and External LSAs.

```
device(config)# router ospf
device(config-ospf-router)# no max-metric router-lsa
```

Configuring OSPF shortest path first throttling

To set OSPF shortest path first throttling to the values in the previous example, use the following command.

```
device(config-ospf-router)# timer throttle spf 200 300 2000
```

Syntax: [no] timer throttle spf *initial-delay hold-time max-hold-time*

The *initial-delay* variable sets the initial value for the SPF delay in milliseconds. Possible values are between 0 and 65535 milliseconds.

The *hold-time* variable sets the minimum hold time between SPF calculations after the initial delay. This value will be doubled after hold-time expires until the max-hold-time is reached. Possible values are between 0 and 65535 milliseconds.

The *max-hold-time* variable sets the maximum hold time between SPF calculations Possible values are between 0 and 65535 milliseconds.

NOTE
The hold time values that you specify are rounded up to the next highest 100 ms value. For example, any value between 0 and 99 will be configured as 100 ms.

Command replacement

This command overlaps in functionality with the timer throttle spf command which will be phased out. To use this command to replicate the exact functionality of the **timer throttle spf** command configure it as shown in the following.

```
device(config-ospf-router)# timer throttle spf 1000 5000 5000
```

Displaying OSPF Router Advertisement

Using the **show ip ospf** command you can display the current OSPF Router Advertisement configuration.

```
device# show ip ospf
OSPF Version                          Version 2
Router Id                             10.10.10.10
ASBR Status                           No
ABR Status                            No          (0)
Redistribute Ext Routes from
External LSA Counter                  5
External LSA Checksum Sum          0002460e
Originate New LSA Counter          5
Rx New LSA Counter                    8
External LSA Limit                    14447047
Database Overflow Interval         0
Database Overflow State :             NOT OVERFLOWED
RFC 1583 Compatibility :              Enabled
Originating router-LSAs with maximum metric
   Condition: Always Current State: Active
   Link Type: PTP STUB TRANSIT
   Additional LSAs originated with maximum metric:
     LSA Type              Metric Value
     AS-External             16711680
     Type 3 Summary          16711680
     Type 4 Summary          16711680
     Opaque-TE               4294967295
```

Displaying OSPF information

You can use CLI commands and Web management options to display the following OSPF information:

- Trap, area, and interface information
- CPU utilization statistics
- Area information
- Neighbor information
- Interface information
- Route information
- External link state information
- Database Information
- Link state information
- Virtual Neighbor information
- Virtual Link information
- ABR and ASBR information
- Trap state information
- OSPF Point-to-Point Links
- OSPF Graceful Restart information
- OSPF Router Advertisement information

Displaying general OSPF configuration information

To display general OSPF configuration information, enter the following command at any CLI level.

```
device# show ip ospf config
Router OSPF: Enabled
Nonstop Routing: Disabled
Graceful Restart: Disabled
Graceful Restart Helper: Enabled
Graceful Restart Time: 120
Graceful Restart Notify Time: 0
Redistribution: Disabled
Default OSPF Metric: 50
OSPF Auto-cost Reference Bandwidth: Disabled
Default Passive Interface: Enabled
OSPF Redistribution Metric: Type2
OSPF External LSA Limit: 1447047
OSPF Database Overflow Interval: 0
RFC 1583 Compatibility: Enabled
Router id: 10.95.11.128
Interface State Change Trap:                        Enabled
Virtual Interface State Change Trap:                Enabled
Neighbor State Change Trap:                         Enabled
Virtual Neighbor State Change Trap:                 Enabled
Interface Configuration Error Trap:                 Enabled
Virtual Interface Configuration Error Trap:         Enabled
Interface Authentication Failure Trap:              Enabled
Virtual Interface Authentication Failure Trap:      Enabled
Interface Receive Bad Packet Trap:                  Enabled
Virtual Interface Receive Bad Packet Trap:          Enabled
Interface Retransmit Packet Trap:                   Disabled
Virtual Interface Retransmit Packet Trap:           Disabled
Originate LSA Trap:                                 Disabled
Originate MaxAge LSA Trap:                          Disabled
Link State Database Overflow Trap:                  Disabled
Link State Database Approaching Overflow Trap:      Disabled
OSPF Area currently defined:
Area-ID          Area-Type Cost
0                normal    0
OSPF Interfaces currently defined:
Ethernet Interface: 3/1-3/2
ip ospf md5-authentication-key-activation-wait-time 300
ip ospf cost 0
ip ospf area 0
Ethernet Interface: v1
ip ospf md5-authentication-key-activation-wait-time 300
ip ospf cost 0
ip ospf area 0
```

Syntax: show ip ospf config

The information related to the OSPF interface state is shown in bold text in the previous output.

TABLE 18 show ip ospf config output descriptions

Field	Description
Router OSPF	Shows whether or not the router OSPF is enabled.
Nonstop Routing	Shows whether or not the non-stop routing is enabled.
Graceful Restart	Shows whether or not the graceful restart is enabled.
Graceful Restart Helper	Shows whether or not the OSPF graceful restart helper mode is enabled.
Graceful Restart Time	Shows the maximum restart wait time advertised to neighbors.

TABLE 18 show ip ospf config output descriptions (Continued)

Field	Description
Graceful Restart Notify Time	Shows the graceful restart notification time.
Redistribution	Shows whether or not the redistribution is enabled.
Default OSPF Metric	Shows the default OSPF metric value.
OSPF Auto-cost Reference Bandwidth	Shows whether or not the auto-cost reference bandwidth option is enabled.
Default Passive Interface	Shows whether or not the default passive interface state is enabled.
OSPF Redistribution Metric	Shows the OSPF redistribution metric type, which can be one of the following: • Type1 • Type2
OSPF External LSA Limit	Shows the external LSA limit value.
OSPF Database Overflow Interval	Shows the database overflow interval value.
RFC 1583 Compatibility	Shows whether or not the RFC 1583 compatibility is enabled.
Router id	Shows the ID of the OSPF router.
OSPF traps	Shows whether or not the following OSPF traps generation is enabled. • Interface State Change Trap • Virtual Interface State Change Trap • Neighbor State Change Trap • Virtual Neighbor State Change Trap • Interface Configuration Error Trap • Virtual Interface Configuration Error Trap • Interface Authentication Failure Trap • Virtual Interface Authentication Failure Trap • Interface Receive Bad Packet Trap • Virtual Interface Receive Bad Packet Trap • Interface Retransmit Packet Trap • Virtual Interface Retransmit Packet Trap • Originate LSA Trap • Originate MaxAge LSA Trap • Link State Database Overflow Trap • Link State Database Approaching Overflow Trap
Area-ID	Shows the area ID of the interface.

TABLE 18 show ip ospf config output descriptions (Continued)

Field	Description
Area-Type	Shows the area type, which can be one of the following: • nssa • normal • stub
Cost	Shows the cost of the area.
Ethernet Interface	Shows the OSPF interface.
ip ospf md5-authentication-key-activation-wait-time	Shows the wait time of the device until placing a new MD5 key into effect.
ip ospf area	Shows the area of the interface.
ip ospf cost	Shows the overhead required to send a packet across an interface.

Displaying CPU utilization and other OSPF tasks

You can display CPU utilization statistics for OSPF and other tasks. To display CPU utilization statistics, enter the following command

```
device# show tasks
Task Name    Pri    State      PC      Stack   Size   CPU Usage(%) task id  task vid
----------   ---   -----   ---------  --------  -----  ---------  ------ -------
       idle 0     ready  00001904  04058fa0  4096    99           0        0
    monitor 20     wait   0000d89c  0404bd80  8192    0            0        0
        int 16     wait   0000d89c  04053f90  16384   0            0        0
      timer 15     wait   0000d89c  04057f90  16384   0            0        0
        dbg 30     wait   0000d89c  0404ff08  8192    0            0        0
      flash 17     wait   0000d89c  0409ff90  8192    0            0        0
         wd 31     wait   0000d89c  0409df80  8192    0            0        0
       boot 17     wait   0000d89c  04203e28  65536   0            0        0
       main 3      wait   0000d89c  2060cf38  65536   0            0        1
        itc 6      wait   0000d89c  20612ae8  16384   0            0        1
        tmr 5      wait   0000d89c  20627628  16384   0            0        1
      ip_rx 5      wait   0000d89c  2062ff48  16384   0            0        1
        scp 5      wait   0000d89c  20635628  16384   0            0        1
    console 5      wait   0000d89c  2063e618  32768   0            0        1
       vlan 5      wait   0000d89c  20648618  16384   0            0        1
    mac_mgr 5      wait   0000d89c  20657628  16384   0            0        1
    mrp_mgr 5      wait   0000d89c  2065c628  16384   0            0        1
       vsrp 5      wait   0000d89c  20663620  16384   0            0        1
       snms 5      wait   0000d89c  20667628  16384   0            0        1
        rtm 5      wait   0000d89c  20674628  16384   0            0        1
       rtm6 5      wait   0000d89c  2068a628  16384   0            0        1
      ip_tx 5     ready   0000d89c  206a9628  16384   0            0        1
        rip 5      wait   0000d89c  20762628  16384   0            0        1
        bgp 5      wait   0000d89c  207e6628  16384   0            0        1
     bgp_io 5      wait   0000d89c  2082ef00  16384   0            0        1
       ospf 5      wait   0000d89c  20832628  16384   1            0        1
ospf_r_calc 5      wait   0000d89c  2089ff10  16384   0            0        1
  isis_task 5      wait   0000d89c  208a3628  16384   0            0        1
   isis_spf 5      wait   0000d89c  208a8f10  16384   0            0        1
      mcast 5      wait   0000d89c  208ac628  16384   0            0        1
       vrrp 5      wait   0000d89c  208b4628  16384   0            0        1
      ripng 5      wait   0000d89c  208b9628  16384   0            0        1
      ospf6 5      wait   0000d89c  208c3628  16384   0            0        1
   ospf6_rt 5      wait   0000d89c  208c7f08  16384   0            0        1
     mcast6 5      wait   0000d89c  208cb628  16384   0            0        1
```

```
         l4 5     wait  0000d89c  208cf620  16384   0           0        1
        stp 5     wait  0000d89c  209a7620  16384   0           0        1
       snmp 5     wait  0000d89c  209c3628  32768   0           0        1
       rmon 5     wait  0000d89c  209cc628  32768   0           0        1
        web 5     wait  0000d89c  209d6628  32768   0           0        1
       lacp 5     wait  0000d89c  209da628  16384   0           0        1
      dot1x 5     wait  0000d89c  209e0620  16384   0           0        1
 hw_access 5     wait  0000d89c  209e6628  16384   0           0
```

Syntax: show tasks

The displayed information shows the following:

TABLE 19 show tasks output descriptions

Field	Description
Task Name	Name of task running on the device.
Pri	Priority of the task in comparison to other tasks.
State	Current state of the task.
PC	Current instruction for the task.
Stack	Stack location for the task.
Size	Stack size of the task.
CPU Usage (%)	Percentage of the CPU being used by the task.
task id	Task ID number assigned by the operating system.
task vid	A memory domain ID.

Displaying OSPF area information

To display OSPF area information, enter the following command at any CLI level.

```
device# show ip ospf area
Indx  Area       Type   Cost  SPFR  ABR  ASBR  LSA  Chksum(Hex)
1   0.0.0.0      normal  0    1      0     0    1    0000781f
2  10.147.60.0   normal  0    1      0     0    1    0000fee6
3  10.147.80.0   stub    1    1      0     0    2    000181cd
```

Syntax: show ip ospf area [*area-id*] [*num*]

The *area-id* parameter shows information for the specified area.

The *num* parameter identifies the position of the entry number in the area table.

TABLE 20 show ip ospf area output descriptions

This field	Displays
Index	The row number of the entry in the router's OSPF area table.

TABLE 20 show ip ospf area output descriptions (Continued)

This field	Displays
Area	The area number.
Type	The area type, which can be one of the following: • nssa • normal • stub
Cost	The area's cost.
SPFR	The SPFR value.
ABR	The ABR number.
ASBR	The ABSR number.
LSA	The LSA number.
Chksum(Hex)	The checksum for the LSA packet. The checksum is based on all the fields in the packet except the age field. The device uses the checksum to verify that the packet is not corrupted.

Displaying OSPF neighbor information

To display OSPF neighbor information, enter the following command at any CLI level.

```
device# show ip ospf neighbor
Port  Address       Pri  State     Neigh Address  Neigh ID     Ev Op Cnt
v10   10.1.10.1     1    FULL/DR   10.1.10.2      10.65.12.1   5  2  0
v11   10.1.11.1     1    FULL/DR   10.1.11.2      10.65.12.1   5  2  0
v12   10.1.12.1     1    FULL/DR   10.1.12.2      10.65.12.1   5  2  0
v13   10.1.13.1     1    FULL/DR   10.1.13.2      10.65.12.1   5  2  0
v14   10.1.14.1     1    FULL/DR   10.1.14.2      10.65.12.1   5  2  0
```

Syntax: show ip ospf neighbor [**router-id** *ip-addr* | *num* | **extensive**]

The **router-id** *ip-addr* parameter displays only the neighbor entries for the specified router.

The *num* parameter displays only the entry in the specified index position in the neighbor table. For example, if you enter "1", only the first entry in the table is displayed.

The **extensive** option displays detailed information about the neighbor.

TABLE 21 show ip ospf neighbor output descriptions

Field	Description
Port	The port through which the device is connected to the neighbor.
Address	The IP address of the port on which this device is connected to the neighbor.

TABLE 21 show ip ospf neighbor output descriptions (Continued)

Field	Description
Pri	The OSPF priority of the neighbor. • For multi-access networks, the priority is used during election of the Designated Router (DR) and Backup designated Router (BDR). • For point-to-point links, this field shows one of the following values: • 1 = point-to-point link • 3 = point-to-point link with assigned subnet
State	The state of the conversation between the device and the neighbor. This field can have one of the following values: • Down - The initial state of a neighbor conversation. This value indicates that there has been no recent information received from the neighbor. • Attempt - This state is only valid for neighbors attached to non-broadcast networks. It indicates that no recent information has been received from the neighbor. • Init - A Hello packet has recently been seen from the neighbor. However, bidirectional communication has not yet been established with the neighbor. (The router itself did not appear in the neighbor's Hello packet.) All neighbors in this state (or higher) are listed in the Hello packets sent from the associated interface. • 2-Way - Communication between the two routers is bidirectional. This is the most advanced state before beginning adjacency establishment. The Designated Router and Backup Designated Router are selected from the set of neighbors in the 2-Way state or greater. • ExStart - The first step in creating an adjacency between the two neighboring routers. The goal of this step is to decide which router is the master, and to decide upon the initial Database Description (DD) sequence number. Neighbor conversations in this state or greater are called adjacencies. • Exchange - The router is describing its entire link state database by sending Database Description packets to the neighbor. Each Database Description packet has a DD sequence number, and is explicitly acknowledged. Only one Database Description packet can be outstanding at any time. In this state, Link State Request packets can also be sent asking for the neighbor's more recent advertisements. All adjacencies in Exchange state or greater are used by the flooding procedure. In fact, these adjacencies are fully capable of transmitting and receiving all types of OSPF routing protocol packets. • Loading - Link State Request packets are sent to the neighbor asking for the more recent advertisements that have been discovered (but not yet received) in the Exchange state. • Full - The neighboring routers are fully adjacent. These adjacencies will now appear in router links and network link advertisements.
Neigh Address	The IP address of the neighbor. For point-to-point links, the value is as follows: • If the **Pri** field is "1", this value is the IP address of the neighbor router's interface. • If the **Pri** field is "3", this is the subnet IP address of the neighbor router's interface.
Neigh ID	The neighbor router's ID.
Ev	The number of times the neighbor's state changed.
Opt	The sum of the option bits in the Options field of the Hello packet. This information is used by Brocade technical support. Refer to Section A.2 in RFC 2178 for information about the Options field in Hello packets.

TABLE 21 show ip ospf neighbor output descriptions (Continued)

Field	Description
Cnt	The number of LSAs that were retransmitted.

Displaying OSPF interface information

To display OSPF interface information, enter the following command at any CLI level. The details of interface options are highlighted in the output.

```
device# show ip ospf interface ethernet 2/1
eth 2/1 admin up, oper down, ospf enabled, state down
     IP Address 1.1.78.8, Area 1
     Database Filter: Not Configured
     State down, Pri 1, Cost 1, Options ------E-, Type broadcast Events 0
     Timers(sec): Transmit 1, Retrans 5, Hello 10, Dead 40
```

If you specify an interface that is not configured within a specified VRF, then the following error message will display as shown in the example below:

```
device# show ip ospf vrf one interface ethernet 1/1
Error: Interface(eth 1/1) not part of VRF(one)
```

NOTE
You cannot display multiple ports for any interfaces. For example, when displaying OSPF interface information on ethernet 1/1 only one port can displayed at a given time.

Syntax: show ip ospf [**vrf** *vrf-name*] **interface** [*ip-addr*] [**brief**] [**ethernet** *port* | **loopback** *number* | **tunnel** *number* | **ve** *number*]

The **vrf** *vrf-name* parameter displays information for VRF, or a specific vrf-name.

The *ip-addr* parameter displays the OSPF interface information for the specified IP address.

The **brief** parameter displays interface information in the brief mode.

The **ethernet**, **loopback**, **tunnel**, and **ve** parameters specify the interface for which to display information. If you specify an Ethernet interface, you can also specify the port number associated with the interface. If you specify a loopback, tunnel, or VE interface, you can also specify the number associated with the interface.

TABLE 22 show ip ospf interface output descriptions

This field	Displays
Interface	The type of interface type and the port number or number of the interface.
IP Address	The IP address of the interface.
Area	The OSPF area configured on the interface
Database Filter	The router's configuration for blocking outbound LSAs on an OSPF interface. If Not Configured is displayed, there is no outbound LSA filter configured. This is the default condition.

TABLE 22 show ip ospf interface output descriptions (Continued)

This field	Displays
State	The state of the interface. Possible states include the following: • DR - The interface is functioning as the Designated Router for OSPFv2. • BDR - The interface is functioning as the Backup Designated Router for OSPFv2. • Loopback - The interface is functioning as a loopback interface. • P2P - The interface is functioning as a point-to-point interface. • Passive - The interface is up but it does not take part in forming an adjacency. • Waiting - The interface is trying to determine the identity of the BDR for the network. • None - The interface does not take part in the OSPF interface state machine. • Down - The interface is unusable. No protocol traffic can be sent or received on such a interface. • DR other - The interface is a broadcast or NBMA network on which another router is selected to be the DR. • Active - The interface sends or receives all the OSPFv2 control packets and forms the adjacency.
default	Shows whether or not the default passive state is set.
Pri	The interface priority.
Cost	The configured output cost for the interface.
Options	OSPF Options (Bit7 - Bit0): • unused:1 • opaque:1 • summary:1 • dont_propagate:1 • nssa:1 • multicast:1 • external route capable:1 • tos:1
Type	The area type, which can be one of the following: • Broadcast • Point to Point • non-broadcast • Virtual Link

TABLE 22 show ip ospf interface output descriptions (Continued)

This field	Displays
Events	OSPF Interface Event: • Interface_Up = 0x00 • Wait_Timer = 0x01 • Backup_Seen = 0x02 • Neighbor_Change = 0x03 • Loop_Indication = 0x04 • Unloop_Indication = 0x05 • Interface_Down = 0x06 • Interface_Passive = 0x07
Timer intervals	The interval, in seconds, of the transmit-interval, retransmit-interval, hello-interval, and dead-interval timers.
DR	The router ID (IPv4 address) of the DR.
BDR	The router ID (IPv4 address) of the BDR.
Neighbor Count	The number of neighbors to which the interface is connected.
Adjacent Neighbor Count	The number of adjacent neighbor routers.
Neighbor:	The IP address of the neighbor.

Displaying OSPF interface brief information

The following command displays the OSPF database brief information.

```
device# show ip ospf interface brief
Number of Interfaces is 1
Interface Area IP Addr/Mask Cost State Nbrs(F/C)
eth 1/2    0    16.1.1.2/24   1     down  0/0
```

TABLE 23 show ip ospf interface brief output descriptions

This field	Displays
Interface	The interface through which the router is connected to the neighbor.
Area	The OSPF Area that the interface is configured in.
IP Addr/Mask	The IP address and mask of the interface.
Cost	The configured output cost for the interface.

TABLE 23 show ip ospf interface brief output descriptions (Continued)

This field	Displays
State	The state of the conversation between the router and the neighbor. This field can have one of the following values: • Down - The initial state of a neighbor conversation. This value indicates that there has been no recent information received from the neighbor. • Attempt - This state is only valid for neighbors attached to non-broadcast networks. It indicates that no recent information has been received from the neighbor. • Init - A Hello packet has recently been seen from the neighbor. However, bidirectional communication has not yet been established with the neighbor. (The router itself did not appear in the neighbor's Hello packet.) All neighbors in this state (or higher) are listed in the Hello packets sent from the associated interface. • 2-Way - Communication between the two routers is bidirectional. This is the most advanced state before beginning adjacency establishment. The Designated Router and Backup Designated Router are selected from the set of neighbors in the 2-Way state or greater. • ExStart - The first step in creating an adjacency between the two neighboring routers. The goal of this step is to decide which router is the master, and to decide upon the initial Database Description (DD) sequence number. Neighbor conversations in this state or greater are called adjacencies. • Exchange - The router is describing its entire link state database by sending Database Description packets to the neighbor. Each Database Description packet has a DD sequence number, and is explicitly acknowledged. Only one Database Description packet can be outstanding at any time. In this state, Link State Request packets can also be sent asking for the neighbor's more recent advertisements. All adjacencies in Exchange state or greater are used by the flooding procedure. In fact, these adjacencies are fully capable of transmitting and receiving all types of OSPF routing protocol packets. • Loading - Link State Request packets are sent to the neighbor asking for the more recent advertisements that have been discovered (but not yet received) in the Exchange state. • Full - The neighboring routers are fully adjacent. These adjacencies will now appear in router links and network link advertisements.
Nbrs(F/C)	The number of adjacent neighbor routers. The number to the left of the "/" are the neighbor routers that are fully adjacent and the number to the right represents all adjacent neighbor routers.

Displaying OSPF route information

To display OSPF route information, enter the following command at any CLI level.

```
device# show ip ospf route
OSPF Area 0x00000000 ASBR Routes 1:
      Destination     Mask            Path_Cost Type2_Cost Path_Type
      10.65.12.1      255.255.255.255 1         0          Intra
      Adv_Router      Link_State      Dest_Type State      Tag       Flags
      10.65.12.1      10.65.12.1      Asbr      Valid      0          6000
      Paths Out_Port  Next_Hop        Type      State
      1     v49       10.1.49.2       OSPF      21 01
      2     v12       10.1.12.2       OSPF      21 01
      3     v11       10.1.11.2       OSPF      21 01
      4     v10       10.1.10.2       OSPF      00 00
OSPF Area 0x00000041 ASBR Routes 1:
      Destination     Mask            Path_Cost Type2_Cost Path_Type
      10.65.12.1      255.255.255.255 1         0          Intra
      Adv_Router      Link_State      Dest_Type State      Tag       Flags
      10.65.12.1      10.65.12.1      Asbr      Valid      0          6000
      Paths Out_Port  Next_Hop        Type      State
      1     v204      10.65.5.251     OSPF      21 01
      2     v201      10.65.2.251     OSPF      20 d1
      3     v202      10.65.3.251     OSPF      20 cd
      4     v205      10.65.6.251     OSPF      00 00
OSPF Area Summary Routes 1:
      Destination     Mask            Path_Cost Type2_Cost Path_    Type
```

```
      10.65.0.0       255.255.0.0     0         0          Inter
      Adv_Router      Link_State      Dest_Type State      Tag       Flags
      10.1.10.1       0.0.0.0         Network   Valid      0          0000
      Paths Out_Port  Next_Hop        Type      State
      1     1/1       0.0.0.0         DIRECT    00 00
OSPF Regular Routes 208:
      Destination     Mask            Path_Cost Type2_Cost Path_Type
      10.1.10.0       255.255.255.252 1         0          Intra
      Adv_Router      Link_State      Dest_Type State      Tag       Flags
      10.1.10.1       10.1.10.2       Network   Valid      0          0000
      Paths Out_Port  Next_Hop        Type      State
      1     v10       0.0.0.0         OSPF      00 00
      Destination     Mask            Path_Cost Type2_Cost Path_Type
      10.1.11.0       255.255.255.252 1         0          Intra
      Adv_Router      Link_State      Dest_Type State      Tag       Flags
      10.1.10.1       10.1.11.2       Network   Valid      0          0000
      Paths Out_Port  Next_Hop        Type      State
      1     v11       0.0.0.0         OSPF      00 00
```

Syntax: show ip ospf routes [*ip-addr*]

The *ip-addr* parameter specifies a destination IP address. If you use this parameter, only the route entries for that destination are shown.

TABLE 24 show ip ospf routes output descriptions

This field	Displays
Destination	The IP address of the route's destination.
Mask	The network mask for the route.
Path_Cost	The cost of this route path. (A route can have multiple paths. Each path represents a different exit port for the device.)
Type2_Cost	The type 2 cost of this path.
Path_Type	The type of path, which can be one of the following: • - Inter - The path to the destination passes into another area. - Intra - The path to the destination is entirely within the local area. - External1 - The path to the destination is a type 1 external route. - External2 - The path to the destination is a type 2 external route.
Adv_Router	The OSPF router that advertised the route to this device.
Link-State	The link state from which the route was calculated.
Dest_Type	The destination type, which can be one of the following: • - ABR - Area Border Router - ASBR - Autonomous System Boundary Router - Network - the network
State	The route state, which can be one of the following: • - Changed - Invalid - Valid This information is used by Brocade technical support.

TABLE 24 show ip ospf routes output descriptions (Continued)

This field	Displays
Tag	The external route tag.
Flags	State information for the route entry. This information is used by Brocade technical support.
Paths	The number of paths to the destination.
Out_Port	The router port through which the device reaches the next hop for this route path.
Next_Hop	The IP address of the next-hop router for this path.
Type	The route type, which can be one of the following: • - OSPF - Static Replaced by OSPF
State	State information for the path. This information is used by Brocade technical support.

Displaying the routes that have been redistributed into OSPF

You can display the routes that have been redistributed into OSPF. To display the redistributed routes, enter the following command at any level of the CLI.

```
device# show ip ospf redistribute route
  4.3.0.0 255.255.0.0 static
  3.1.0.0 255.255.0.0 static
  10.11.61.0 255.255.255.0 connected
  4.1.0.0 255.255.0.0 static
```

In this example, four routes have been redistributed. Three of the routes were redistributed from static IP routes and one route was redistributed from a directly connected IP route.

Syntax: show ip ospf redistribute route [*ip-addr ip-mask*]

The *ip-addr ip-mask* parameter specifies a network prefix and network mask. Here is an example.

```
Brocade# show ip ospf redistribute route 192.213.1.0 255.255.255.254
  192.213.1.0 255.255.255.254 fwd 0.0.0.0 (0) metric 10 connected
```

Displaying OSPF database information

The following command displays the OSPF database.

```
device# show ip ospf database
Graceful Link States
Area  Interface  Adv Rtr  Age Seq(Hex) Prd Rsn  Nbr Intf IP
0     eth 1/2 10.2.2.2  7   80000001 60  SW 10.1.1.2

Router Link States
Index AreaID            Type LS ID      Adv Rtr     Seq(Hex)   Age  Cksum
1     0                 Rtr 10.2.2.2    10.2.2.2     80000003   93   0xac6c
2     0                 Rtr 10.1.1.1    10.1.1.1     80000005   92   0x699e
3     0                 Net  10.1.1.2   10.2.2.2     80000002   93   0xbd73
4     0                 OpAr 10.0.0.3   10.1.1.1     80000005   83   0x48e7
5     0                 OpAr 10.0.0.2   10.2.2.2     80000006   80   0x50da
6     10.111.111.111    Rtr  10.1.1.1   10.1.1.1     80000004   142  0x0a38
```

```
7     10.111.111.111     Summ 10.1.1.1   10.1.1.1     80000001   147  0x292b
8     10.111.111.111     OpAr 10.0.0.2   10.1.1.1     80000002   179  0x063f

Type-5 AS External Link States
Index Age  LS ID     Router   Netmask   Metric    Flag  Fwd Address
1     147 10.9.1.13  10.1.1.1  ffffffff  0000000a  0000  0.0.0.0
2     147 10.9.1.26  10.1.1.1  ffffffff  0000000a  0000  0.0.0.0
```

Syntax: show ip ospf database

TABLE 25 show ip ospf databaseoutput descriptions

This field	Displays
Area	The OSPF area that the interface configured for OSPF graceful restart is in.
Interface	The interface that is configured for OSPF graceful restart.
Prd	Grace Period: The number of seconds that the router's neighbors should continue to advertise the router as fully adjacent, regardless of the state of database synchronization between the router and its neighbors. Since this time period began when grace-LSA's LS age was equal to 0, the grace period terminates when either: • the LS age of the grace-LSA exceeds the value of a Grace Period • the grace-LSA is flushed
Rsn	Graceful restart reason: The reason for the router restart defined as one of the following: • UK – unknown • RS – software restart • UP – software upgrade or reload • SW – switch to redundant control processor
Nbr Intf IP	The IP address of the OSPF graceful restart neighbor.
Index	ID of the entry.
Aging	The age of the LSA in seconds.
Area ID	ID of the OSPF area.
Type	Link state type of the route.
LS ID	The ID of the link-state advertisement from which the router learned this route
Adv Rtr	ID of the advertised route.
Seq (Hex)	The sequence number of the LSA. The OSPF neighbor that sent the LSA stamps the LSA with a sequence number. This number enables the device and other OSPF routers to determine which LSA for a given route is the most recent.
Age	The age of the LSA in seconds.
Chksum	The checksum for the LSA packet. The checksum is based on all the fields in the packet except the age field. The device uses the checksum to verify that the packet is not corrupted.

TABLE 25 show ip ospf databaseoutput descriptions (Continued)

This field	Displays
Router	The router IP address.
Netmask	The subnet mask of the network.
Metric	The cost (value) of the route.
Flag	State information for the route entry. This information is used by Brocade technical support.

Displaying OSPF external link state information

To display the details of the external link state information for LSA with options, enter the following command at any CLI level. The options are highlighted under the LSA Header.

```
device# show ip ospf database external-link-state extensive

AS-external LSA
Index Age  LS ID           Router          Netmask  Metric   Flag Fwd Address
SyncState
1     1064 5.1.1.0         21.21.21.21     ffffff00 0000000a 0000 0.0.0.0
Done
  LSA Header:  age: 1064, options: -------E-, seq-nbr: 0x80000001, length: 36
  NetworkMask: 255.255.255.0
  TOS 0:  metric_type: 2, metric: 10
          forwarding_address: 0.0.0.0
          external_route_tag: 0
```

Syntax:show ip ospf [vrf *vrf-name*] database external-link-state [**advertise** *num* | **extensive** | **link-state-id** *A.B.C.D* | **router-id** *A.B.C.D* | **sequence-number** *num(Hex)*]

The **vrf** *vrf-name* parameter displays information for a VRF, or a specific *vrf-name*.

The **advertise** *num* parameter displays the decoded data in the specified LSA packet. The *num* parameter identifies the LSA packet by its position in the router's External LSA table. To determine an LSA packet's position in the table, enter the **show ip ospf external-link-state** command to display the table.

The **extensive** option displays the LSAs in the decoded format.

NOTE
You cannot use the **extensive** option in combination with other display options. The entire database is displayed.

The **link-state-id** *A.B.C.D* parameter displays the External LSAs for the LSA source specified by *A.B.C.D* (link state ID).

The **router-id** *A.B.C.D* (advertising router ID) parameter shows the External LSAs for the specified OSPF router.

The **sequence-number** *num(Hex)* parameter displays the External LSA entries for the specified hexadecimal LSA sequence number.

TABLE 26 show ip ospf database external-link-state output descriptions

This field	Displays
Area ID	ID of the OSPF area.
Type LS ID	Link state type of the route. The ID of the link-state advertisement.
Adv Rtr	ID of the advertising router.
Seq(Hex)	The sequence number of the LSA. The OSPF neighbor that sent the LSA stamps the LSA with a sequence number. This number enables the device and other OSPF routers to determine which LSA for a given route is the most recent.
Age	The age of the LSA in seconds.
Chksum	The checksum for the LSA packet. The checksum is based on all the fields in the packet except the age field. The device uses the checksum to verify that the packet is not corrupted.
SyncState	This field indicates whether the synchronization is complete or not.

The following tables display the OSPF options, flags, and their abbreviations.

TABLE 27 OSPF option descriptions

BIT	Displayed as	Description
0	-	-
1	E	External route capable
2	M	Multicast
3	N/P	N-NSSA translation capable (router LSA), P - translate to type-5 LSA.
4	D	Do not propagate bit.
5	Dc	Demand Circuit
6	O	Opaque LSA Capable
7	Dn	Down Bit

Displaying OSPF database-summary information

To display database-summary information, enter the following command at any CLI level.

```
device# show ip ospf database database-summary
Area ID    Router Network Sum-Net Sum-ASBR NSSA-Ext Opq-Area Subtotal
0.0.0.0    104    184     19      42       0        0        349
```

```
AS External                                                    308
Total      104     184      19       42        0         0
657
```

Syntax: show ip ospf database database-summary

TABLE 28 show ip ospf database database-summary output descriptions

This field	Displays
Area ID	The area number.
Router	The number of router link state advertisements in that area.
Network	The number of network link state advertisements in that area.
Sum-Net	The number of summary link state advertisements in that area.
Sum-ASBR	The number of summary autonomous system boundary router (ASBR) link state advertisements in that area
NSSA-Ext	The number of not-so-stubby
Opq-area	the number of Type-10 (area-scope) Opaque LSA.s

Displaying OSPF database link state information

To display the details of the extensive link state information for LSA with options and flags, enter the following command at any CLI level. The options and flags are highlighted under the LSA Header.

```
device#show ip ospf database link-state extensive
Router LSA:
Area ID         Type LS ID           Adv Rtr         Seq(Hex) Age  Cksum  SyncState
0               Rtr  21.21.21.21     21.21.21.21     80000006 338  0xcd13 Done
  LSA Header: options: -------E-, seq-nbr: 0x80000006, length: 60, flags:-----EB
  link id = 31.31.31.31, link data = 106.50.50.10, type = virtual(4)
  tos count = 0, tos0_metric = 1
  link id = 106.10.10.10, link data = 106.10.10.10, type = transit(2)
  tos count = 0, tos0_metric = 1
  link id = 106.20.20.10, link data = 106.20.20.10, type = transit(2)
  tos count = 0, tos0_metric = 1

Network LSA:
Area ID         Type LS ID           Adv Rtr         Seq(Hex) Age  Cksum  SyncState
0               Net  106.20.20.10    21.21.21.21     80000002 353  0x6285 Done
  LSA Header:  options: -------E-, seq-nbr: 0x80000002, length: 32
  NetworkMask: 255.255.255.0
  attached router: 21.21.21.21
  attached router: 11.11.11.11

NSSA LSA:
Area ID         Type LS ID           Adv Rtr         Seq(Hex) Age  Cksum  SyncState
2               NSSA 2.0.0.0         130.130.130.3   80000001 426  0x780b Done
  LSA Header:  age: 426, options: -----P---, seq-nbr: 0x80000001, length: 36
  NetworkMask: 255.255.255.0
  TOS 0:  metric_type: 2, metric: 10
          forwarding_address: 106.30.30.10
          external_route_tag: 0
```

Syntax:show ip ospf [vrf *vrf-name*] database link-state [**advertise** *num* | **asbr** [*ip-addr*] [**adv-router** *ip-addr*] | **extensive** |**link-state-id** *ip-addr* | **network** [*ip-addr*] [**adv-router** *ip-addr*] | **nssa** [

ip-addr] [**adv-router** *ip-addr*] | **router** [*ip-addr*] [**adv-router** *ip-addr*] | **router-id** *ip-addr* | **self-originate l sequence-number** *num(Hex)* | **summary** [*ip-addr*] [**adv-router** *ip-addr*]

The **vrf** *vrf-name* parameter displays information for a VRF, or a specific *vrf-name*.

The **advertise** *num* parameter displays the decoded data in the specified LSA packet. The *num* parameter identifies the LSA packet by its position in the router's LSA table. To determine an LSA packet's position in the table, enter the **show ip ospf link-state** command to display the table.

The **asbr** option shows ASBR LSAs.

The **extensive** option displays the LSAs in the decoded format.

NOTE
You cannot use the **extensive** option in combination with other display options. The entire database is displayed.

The **link-state-id** *A.B.C.D* parameter displays the LSAs for the LSA source specified by *A.B.C.D* (link state ID).

The **network** option shows network LSAs.

The **nssa** option shows NSSA LSAs.

The **router-id** *A.B.C.D* (advertising router ID) parameter shows the LSAs for the specified OSPF router.

The **sequence-number** *num* parameter displays the LSA entries for the specified hexadecimal LSA sequence number.

The **self-originate** option shows self-originated LSAs.

TABLE 29 show ip ospf database link-state output descriptions (Continued)

This field	Displays
Area ID	ID of the OSPF area
Type LS ID	Type and ID of the link state advertisement.
Adv Rtr	ID of the advertising router.
Seq(Hex)	The sequence number of the LSA. The OSPF neighbor that sent the LSA stamps the LSA with a sequence number. This number enables the device and other OSPF routers to determine which LSA for a given route is the most recent.
Age	The age of the LSA in seconds.
Cksum	The checksum for the LSA packet. The checksum is based on all the fields in the packet except the age field. The device uses the checksum to verify that the packet is not corrupted.
SyncState	This field indicates whether the synchronization is complete or not.

The following tables display the OSPF options, flags, and their abbreviations.

TABLE 30 OSPF option descriptions

BIT	Displayed as	Description
0	-	-
1	E	External route capable
2	M	Multicast
3	N/P	N-NSSA translation capable (router LSA), P - translate to type-5 LSA.
4	D	Do not propagate bit.
5	Dc	Demand Circuit
6	O	Opaque LSA Capable
7	Dn	Down Bit

TABLE 31 Router LSA flag descriptions

BIT	Displayed as	Description
0	B	Area Border Router
1	E	External route capable
2	V	Virtual Link
3	-	-
4	Nt	NSSA translator
5	-	-
6	-	-
7	-	-

Displaying OSPF ABR and ASBR information

To display OSPF ABR and ASBR information, enter the following command at any CLI level.

```
device# show ip ospf border-routers
```

Syntax: show ip ospf border-routers [*ip-addr*]

The *ip-addrip-addr* parameter displays the ABR and ASBR entries for the specified IP address.

```
device# show ip ospf border-routers
```

```
      router ID       router type next hop router  outgoing interface  Area
1     10.65.12.1      ABR         10.1.49.2        v49                 0
1     10.65.12.1      ASBR        10.1.49.2        v49                 0
1     10.65.12.1      ABR         10.65.2.251      v201                65
1     10.65.12.1      ASBR        10.65.2.251      v201                65
```

Syntax: show ip ospf border-routers

TABLE 32 show ip ospf border-routersoutput descriptions

This field	Displays
(Index)	Displayed index number of the border router.
Router ID	ID of the OSPF router
Router type	Type of OSPF router: ABR or ASBR
Next hop router	ID of the next hop router
Outgoing interface	ID of the interface on the router for the outgoing route.
Area	ID of the OSPF area to which the OSPF router belongs

Displaying OSPF trap status

All traps are enabled by default when you enable OSPF.

To display the state of each OSPF trap, enter the following command at any CLI level.

```
device# show ip ospf trap
Interface State Change Trap:                      Enabled
Virtual Interface State Change Trap:              Enabled
Neighbor State Change Trap:                       Enabled
Virtual Neighbor State Change Trap:               Enabled
Interface Configuration Error Trap:               Enabled
Virtual Interface Configuration Error Trap:       Enabled
Interface Authentication Failure Trap:            Enabled
Virtual Interface Authentication Failure Trap:    Enabled
Interface Receive Bad Packet Trap:                Enabled
Virtual Interface Receive Bad Packet Trap:        Enabled
Interface Retransmit Packet Trap:                 Disabled
Virtual Interface Retransmit Packet Trap:         Disabled
Originate LSA Trap:                               Disabled
Originate MaxAge LSA Trap:                        Disabled
Link State Database Overflow Trap:                Disabled
Link State Database Approaching Overflow Trap:    Disabled
```

Syntax: show ip ospf trap

Viewing Configured OSPF point-to-point links

You can use the show ip ospf interface command to display OSPF point-to-point information. Enter the following command at any CLI level.

```
device# show ip ospf interface 192.168.1.1
Ethernet 2/1,OSPF enabled
IP Address 192.168.1.1, Area 0
OSPF state ptr2ptr, Pri 1, Cost 1, Options 2, Type pt-2-pt Events 1
Timers(sec): Transit 1, Retrans 5, Hello 10, Dead 40
DR: Router ID 0.0.0.0 Interface Address 0.0.0.0
```

```
BDR: Router ID 0.0.0.0 Interface Address 0.0.0.0
Neighbor Count = 0, Adjacent Neighbor Count= 1
Neighbor: 2.2.2.2
Authentication-Key:None
MD5 Authentication: Key None, Key-Id None, Auth-change-wait-time 300
```

Syntax: show ip ospf interface [**ip-addr**]

The *ip-addr* parameter displays the OSPF interface information for the specified IP address.

TABLE 33 show ip ospf interfaceoutput descriptions

This field	Displays
IP Address	The IP address of the interface.
OSPF state	The OSPF state of the interface.
Pri	The router priority.
Cost	The configured output cost for the interface.
Options	OSPF Options (Bit7 - Bit0): • unused:1 • opaque:1 • summary:1 • dont_propagate:1 • nssa:1 • multicast:1 • externals:1 • tos:1
Type	The area type, which can be one of the following: • Broadcast = 0x01 • NBMA = 0x02 • Point to Point = 0x03 • Virtual Link = 0x04 • Point to Multipoint = 0x05
Events	OSPF Interface Event: • Interface_Up = 0x00 • Wait_Timer = 0x01 • Backup_Seen = 0x02 • Neighbor_Change = 0x03 • Loop_Indication = 0x04 • Unloop_Indication = 0x05 • Interface_Down = 0x06 • Interface_Passive = 0x07
Adjacent Neighbor Count	The number of adjacent neighbor routers.
Neighbor:	The IP address of the neighbor.

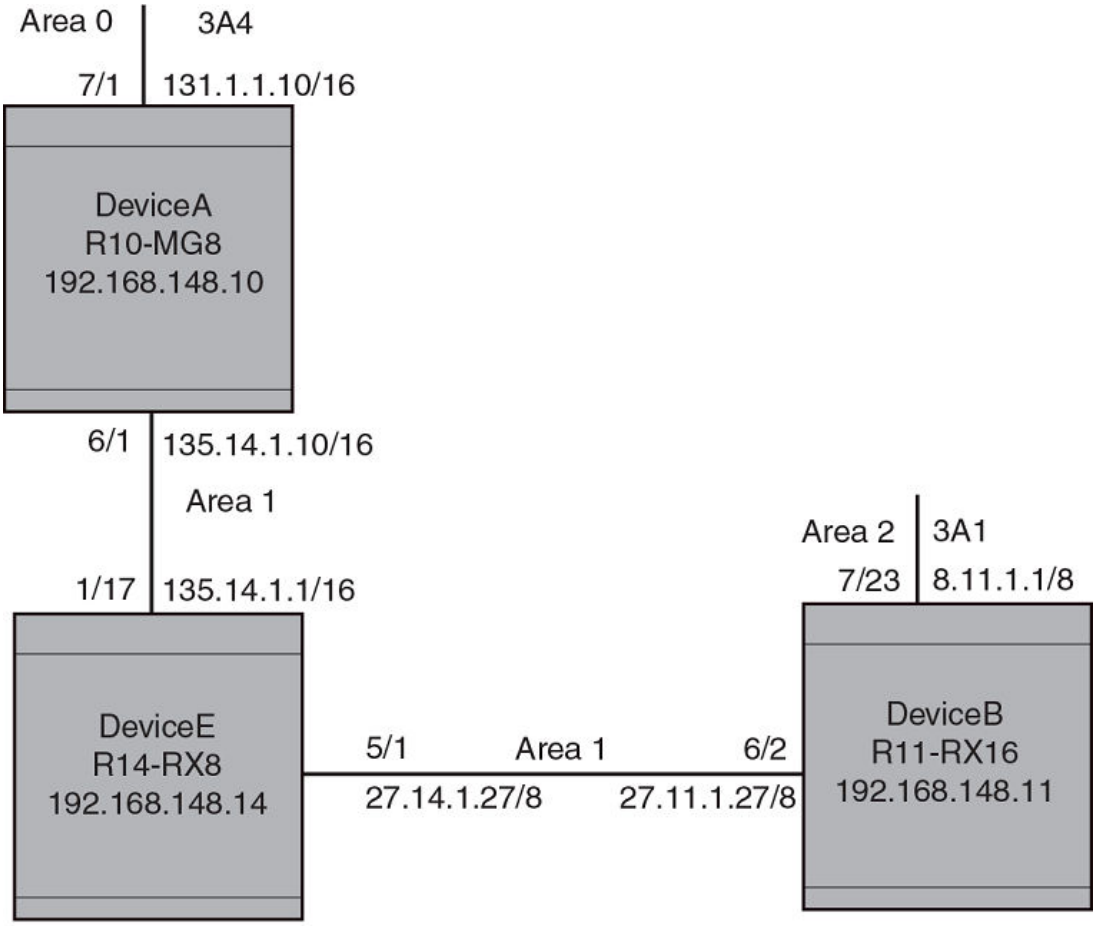
Displaying OSPF virtual neighbor and link information

You can display OSPF virtual neighbor and virtual link information.

You can display OSPF virtual neighbor and virtual link information. The **show run** output shows the configuration in the example.

```
device# show run
Current configuration:
!
ver V2.2.1T143
module 1 rx-bi-1g-24-port-fiber
module 2 rx-bi-10g-4-port
module 6 rx-bi-10g-4-port
module 7 rx-bi-1g-24-port-copper
!
!
no spanning-tree
!
vlan 1 name DEFAULT-VLAN
!
!
clock summer-time
clock timezone us Pacific
hostname R11-RX8
router ospf
 area 2
 area 1
 area 1 virtual-link 10.1.1.10
```

FIGURE 16 OSPF virtual neighbor and virtual link example



Displaying OSPF virtual neighbor

Use the **show ip ospf virtual neighbor** command to display OSPF virtual neighbor information.

```
device# show ip ospf virtual neighbor
Indx Transit Area    Router ID      Neighbor address options
1    1               131.1.1.10     135.14.1.10       2
     Port    Address         state   events          count
     6/2     27.11.1.27      FULL    5               0
```

Syntax: show ip ospf virtual neighbor [*num*]

The *num* parameter displays the table beginning at the specified entry number.

Displaying OSPF virtual link information

Use the **show ip ospf virtual link** command to display OSPF virtual link information.

```
device# show ip ospf virtual link
```

```
Indx Transit Area     Router ID        Transit(sec) Retrans(sec) Hello(sec)
1    1                131.1.1.10       1            5            10
     Dead(sec)        events           state        Authentication-Key
     40               1                ptr2ptr      None
     MD5 Authentication-Key:       None
     MD5 Authentication-Key-Id:    None
     MD5 Authentication-Key-Activation-Wait-Time:      300
```

Syntax: show ip ospf virtual link [*num*]

The *num* parameter displays the table beginning at the specified entry number.

Clearing OSPF neighbors

You can clear all OSPF neighbors or a specified OSPF neighbor using the following command.

```
device# clear ip ospf neighbor all
```

Syntax: clear ip ospf neighbor { **all** | *ip-address* }

Selecting the **all** option clears all of the OSPF neighbors on the router.

The *ip-address* variable allows you to clear a specific OSPF neighbor.

Displaying OSPF Graceful Restart information

To display OSPF Graceful Restart information for OSPF neighbors use the **show ip ospf neighbors** command as shown in the following.

```
device# show ip ospf neighbors
Port  Address       Pri State       Neigh Address  Neigh ID      Ev Opt Cnt
2/7   50.50.50.10   0   FULL/OTHER  50.50.50.1     10.10.10.30   21 66  0
  < in graceful restart state,  helping 1, timer 60 sec >
```

Use the following command to display Type 9 Graceful LSAs on a router.

```
device# show ip ospf database grace-link-state
Graceful Link States
Area  Interface  Adv Rtr  Age Seq(Hex) Prd Rsn  Nbr Intf IP
0     eth 1/2    2.2.2.2  7   80000001 60  SW   6.1.1.2
```

TABLE 34 show ip ospf database grace-link-state output descriptions

This field	Displays
Area	The OSPF area that the interface configured for OSPF graceful restart is in.
Interface	The interface that is configured for OSPF graceful restart.
Adv Rtr	ID of the advertised route.
Age	The age of the LSA in seconds.
Seq(Hex)	The sequence number of the LSA. The OSPF neighbor that sent the LSA stamps the LSA with a sequence number. This number enables the device and other OSPF routers to determine which LSA for a given route is the most recent.

TABLE 34 show ip ospf database grace-link-state output descriptions (Continued)

This field	Displays
Prd	Grace Period: The number of seconds that the router's neighbors should continue to advertise the router as fully adjacent, regardless of the state of database synchronization between the router and its neighbors. Since this time period began when grace-LSA's LS age was equal to 0, the grace period terminates when either: • the LS age of the grace-LSA exceeds the value of a Grace Period • the grace-LSA is flushed.
Rsn	Graceful restart reason: The reason for the router restart defined as one of the following: • UK - unknown • RS - software restart • UP - software upgrade or reload • SW - switch to redundant control processor
Nbr Intf IP	The IP address of the OSPF graceful restart neighbor.

Displaying OSPF Router Advertisement information

Using the **show ip ospf** command you can display the current OSPF Router Advertisement configuration. The text show below in bold is displayed for an OSPF Router Advertisement configuration.

```
device# show ip ospf
OSPF Version                          Version 2
Router Id                             10.10.10.10
ASBR Status                           No
ABR Status                            No          (0)
Redistribute Ext Routes from
External LSA Counter                  5
External LSA Checksum Sum         0002460e
Originate New LSA Counter         5
Rx New LSA Counter                    8
External LSA Limit                    14447047
Database Overflow Interval        0
Database Overflow State :             NOT OVERFLOWED
RFC 1583 Compatibility :              Enabled
Originating router-LSAs with maximum metric
   Condition: Always Current State: Active
   Link Type: PTP STUB TRANSIT
   Additional LSAs originated with maximum metric:
     LSA Type              Metric Value
     AS-External             16711680
     Type 3 Summary          16711680
     Type 4 Summary          16711680
     Opaque-TE               4294967295
```

Multi-Service IronWare version 03.5.00 enhances the **show ip ospf** command to display LSAs that have been configured with a maximum metric.

Displaying the OSPF area translator status information

Run the **show ip ospf area** command at the OSPF router level, to display the status of the area translator..

```
device (config)# router ospf
device (config-ospf-router)# show ip ospf area
Number of Areas  is 3, NSSA area 2
Indx Area       Type   Cost     SPFR    ABR   ASBR  LSA   Chksum(Hex) Translator
```

```
1     1          nssa   1         56      1     1     10802 153598b6   Candidate
2     0        normal   0         56      0     0     4     0001a1f0      --
3     2          nssa   7         56      0     0     798   018c972d   Elected
```

TABLE 35 show ip ospf area output descriptions

This field	Displays
Indx	The row number of the entry in the router's OSPF area table.
Area	The area number.
Type	The area type, which can be one of the following: • nssa • normal • stub
Cost	The area's cost.
SPFR	The SPFR value. Number of times the SPF has run for this area.
ABR	The ABR number.
ASBR	The ASBR number.
LSA	The LSA number.
Chksum (hex)	The checksum for the LSA packet. The checksum is based on all the fields in the packet except the age field. The device uses the checksum to verify that the packet is not corrupted.
Translator	The translator role. The types of translator roles are: • Candidate • Elected

Clearing OSPF information

You can use the **clear ip ospf** commands to clear OSPF data on an router as described in the following:

- Neighbor information
- Reset the OSPF process
- Clear and re-add OSPF routes

Clearing OSPF neighbors

You can use the following command to delete and relearn all OSPF neighbors, all OSPF neighbors for a specified interface or a specified OSPF neighbor.

```
device# clear ip ospf neighbor all
```

Syntax: clear ip ospf [**vrf** *vrf-name*] **neighbor all** [*interface*] | *interface* | *ip-address* [*interface*]

Selecting the **all** option without specifying an interface clears all of the OSPF neighbors on the router.

The *interface* variable specifies the interface that you want to clear all of the OSPF neighbors on. The following types of interfaces can be specified:

- **ethernet** *slot/port*
- **tunnel** *tunnel-ID*
- **ve***ve-ID*

The *ip-address* variable allows you to clear a specific OSPF neighbor.

Disabling and re-enabling the OSPF process

You can use the following command to disable and re-enable the OSPF process on a router.

```
device# clear ip ospf all
```

Syntax: clear ip ospf [**vrf** *vrf-name*] **all**

This command resets the OSPF process and brings it back up after releasing all memory used while retaining all configurations.

Clearing OSPF routes

You can use the following command to clear all OSPF routes or to clear a specific OSPF route.

```
device# clear ip ospf routes all
```

Syntax: clear ip ospf [**vrf** *vrf-name*] **routes** { **all** | *ip-address/prefix-length* }

Selecting the **all** option resets the OSPF routes including external routes, and OSPF internal routes.

The *ip-address* and *prefix-length* variables specify a particular route to delete and then reschedules the SPF calculation.

IS-IS (IPv4)

- Relationship to IP route table....258
- IS-IS CLI levels....262
- Globally configuring IS-IS on a device....264
- Configuring IPv4 address family route parameters....275
- Configuring IS-IS point-to-point over Ethernet....283
- Configuring IS-IS over a GRE IP tunnel....284
- IS-IS Non-Stop Routing....287
- Configuring ISIS properties on an interface....294
- Displaying IPv4 IS-IS information....298
- Clearing the IS-IS SPF Log....321
- Triggering the router to run SPF....322
- Clearing IS-IS information....322
- Clearing a specified LSP from IS-IS database....323

Table 36 displays the individual Brocade devices and the IPv4 IS-IS features they support.

TABLE 36 Supported Brocade IPv4 IS-IS features

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
IS-IS	Yes	Yes	No	Yes	Yes	Yes	Yes
Level-1 Routing	Yes	Yes	No	Yes	Yes	Yes	Yes
Level-2 Routing	Yes	Yes	No	Yes	Yes	Yes	Yes
Restart helper-mode	Yes	Yes	No	Yes	Yes	Yes	Yes
Broadcast Pseudonode	Yes	Yes	No	Yes	Yes	Yes	Yes
Three-Way Handshake for Point-to-Point Adjacencies	Yes	Yes	No	Yes	Yes	Yes	Yes
IS-IS PSPF Exponential back-off	Yes	Yes	No	Yes	Yes	Yes	Yes

TABLE 36 Supported Brocade IPv4 IS-IS features (Continued)

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
New encryption code for passwords, authentication keys, and community strings	Yes	Yes	No	No	No	No	No
IS-IS Flooding	Yes	Yes	No	Yes	Yes	Yes	Yes
IS-IS Point-to-Point over Ethernet	Yes	Yes	No	Yes	Yes	Yes	Yes
IS-IS over a GRE IP Tunnel	Yes	Yes	No	No	No	No	No
Formation of Adjacencies	Yes	Yes	No	Yes	Yes	Yes	Yes
IS-IS Blackhole Avoidance (Setting the Overload Bit)	Yes	Yes	No	Yes	Yes	Yes	Yes
Priority for Designated IS Election	Yes	Yes	No	Yes	Yes	Yes	Yes
Limiting Access to Adjacencies With a Neighbor	Yes	Yes	No	Yes	Yes	Yes	Yes
Changing the IS-IS Level on an Interface	Yes	Yes	No	Yes	Yes	Yes	Yes
Disabling and Enabling Hello Padding on an Interface	Yes	Yes	No	Yes	Yes	Yes	Yes
Displaying IPv4 IS-IS Information	Yes	Yes	No	Yes	Yes	Yes	Yes
IS-IS SPF Scaling	Yes	Yes	No	No	No	No	No
Disabling and enabling IS-IS graceful restart helper mode	Yes	Yes	No	Yes	Yes	Yes	Yes

TABLE 36 Supported Brocade IPv4 IS-IS features (Continued)

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
IS-IS interface metric enhancement	Yes	Yes	No	Yes	Yes	Yes	Yes
IS-IS reverse metric	Yes	Yes	No	Yes	Yes	Yes	Yes

The Intermediate System to Intermediate System (IS-IS) protocol is a link-state Interior Gateway Protocol (IGP) that is based on the International Standard for Organization/International Electrotechnical Commission (ISO/IEC) Open Systems Internet Networking model (OSI). In IS-IS, an intermediate system (router) is designated as either a Level 1 or Level 2 router. A Level 1 router routes traffic only within the area in which the router resides. A Level 2 router routes traffic between areas within a routing domain.

The implementation of IS-IS is based on the following specifications and draft specifications:

- ISO/IEC 10589 - "Information Technology - Telecommunication and information exchange between systems - Intermediate system to Intermediate system intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connection less-mode Network Service (ISO 8473)", 1992
- ISO/IEC 8473 - "Information processing systems - Data Communications - Protocols for providing the connectionless-mode network service", 1988
- ISO/IEC 9542 - "Information Technology - Telecommunication and information exchange between systems - End system to Intermediate system intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connection less-mode Network Service (ISO 8473)", 1988
- RFC 1195 - "Use of OSI IS-IS for Routing in TCP/IP and Dual Environments", 1990.
- RFC 2763 - "Dynamic Host Name Exchange Mechanism for IS-IS", 2000.
- RFC 2966 - "Domain-wide Prefix Distribution with Two-Level IS-IS", 2000
- RFC 3373 - "Three-Way Handshake for Intermediate System to Intermediate System (IS-IS) Point-to-Point Adjacencies", 2002
- Portions of the Internet Draft "IS-IS extensions for Traffic Engineering" draft-ieff-isis-traffic-02.txt (dated 2000). that describe the Extended IP reachability type-length-value (TLV type 135) and the extended Intermediate System (IS) reachability TLV (TLV type 22). These portions provide support for the wide metric version of IS-IS. No other portion is supported on Brocade's implementation of IS-IS.

NOTE
The Brocade device does not support routing of Connectionless-Mode Network Protocol (CLNP) packets. The Brocade device uses IS-IS for TCP/IP only.

Relationship to IP route table

The IS-IS protocol has the same relationship to the Brocade device's IP route table that OSPF has to the IP route table. The IS-IS routes are calculated and first placed in the IS-IS route table. The routes are then transferred to the IP route table.

The protocol sends the best IS-IS path for a given destination to the IP route table for comparison to the best paths from other protocols to the same destination. The CPU selects the path with the lowest administrative distance and places that path in the IP route table:

- If the path provided by IS-IS has the lowest administrative distance, then the CPU places that IS-IS path in the IP route table.
- If a path to the same destination supplied by another protocol has a lower administrative distance, the CPU installs the other protocol's path in the IP route table instead.

The **administrative distance** is a protocol-independent value from 1 - 255. Each path sent to the CPU, regardless of the source of the path (IS-IS, OSPF, static IP route, and so on) has an administrative distance.

Each route source has a default administrative distance. The default administrative distance for IS-IS is 115.

You can change the administrative distance for IS-IS and other routes sources.

Intermediate systems and end systems

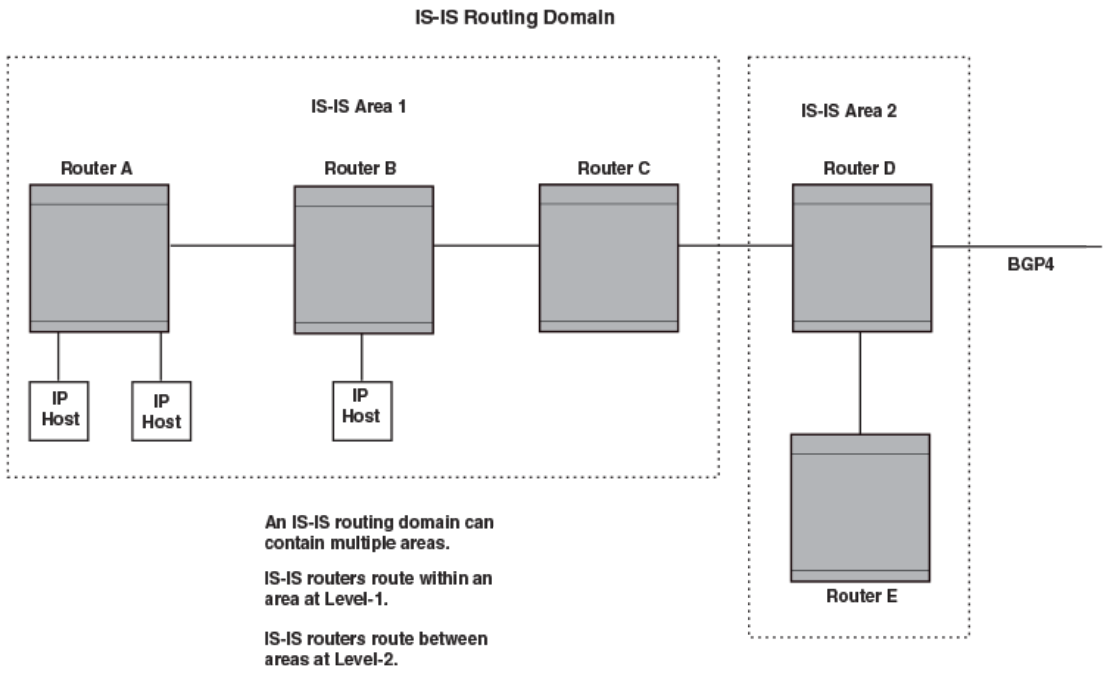
IS-IS uses the following categories to describe devices within an IS-IS routing domain (similar to an OSPF Autonomous System):

- Intermediate System (IS) - A device capable of forwarding packets from one device to another within the domain. In Internet Protocol (IP) terminology, an IS is a router.
- End System (ES) - A device capable of generating or receiving packets within the domain. In IP terminology, an ES is an end node or IP host.

When you configure IS-IS on a Brocade device, the device is an IS.

Figure 17 shows an example of an IS-IS network.

FIGURE 17 An IS-IS network contains Intermediate Systems (ISs) and host systems



NOTE
Since the implementation of IS-IS does not route OSI traffic but instead routes IP traffic, IP hosts are shown instead of ESs.

The other basic IS-IS concepts illustrated in this figure are explained in the following sections.

Domain and areas

IS-IS is an IGP, and thus applies only to routes within a single routing domain. However, you can configure multiple areas within a domain. A Brocade device can be a member of one area for each Network Entity Title (NET) you configure on the Brocade device. The NET contains the area ID for the area the NET is in.

In Intermediate systems and end systems on page 258, Routers A, B, and C are in area 1. Routers D and E are in area 2. All the routers are in the same domain.

Level-1 routing and Level-2 routing

You can configure an IS-IS router such as a Brocade device to perform one or both of the following levels of IS-IS routing:

NOTE
The ISO/IEC specifications use the spelling "routeing", but this document uses the spelling "routing" to remain consistent with other Brocade documentation.

- Level-1 - A Level-1 router routes traffic only within the area the router is in. To forward traffic to another area, the Level-1 router sends the traffic to its nearest Level-2 router.
- Level-2 - A Level-2 router routes traffic between areas within a domain.

In Intermediate systems and end systems on page 258, Routers A and B are Level-1s only. Routers C and D are Level-1 and Level-2 ISs. Router E is a Level-1 IS only.

Neighbors and adjacencies

A Brocade device configured for IS-IS forms an adjacency with each of the IS-IS devices to which it is directly connected. An adjacency is a two-way direct link (a link without router hops) over which the two devices can exchange IS-IS routes and other protocol-related information. The link is sometimes called a "circuit". The devices with which the Brocade device forms adjacencies are its neighbors , which are other ISs.

In Intermediate systems and end systems on page 258, Router A has an IS-IS adjacency with Router B. Likewise, Router B has an IS-IS adjacency with Router A and Router C.

Designated IS

A Designated IS is an IS-IS router that is responsible for gathering and distributing link state information to other Level-1 or Level-2 ISs within the same broadcast network (LAN). The Level-1 and Level-2 Designated ISs within a broadcast network are independent, although the same Brocade device can be a Level-1 Designated IS and a Level-2 Designated IS at the same time.

The Designated IS is elected based on the priority of each IS in the broadcast network. When an IS becomes operational, it sends a Level-1 or Level-2 Hello PDU to advertise itself to other ISs. If the IS is configured to be both a Level-1 and a Level-2 IS, the IS sends a separate advertisement for each level:

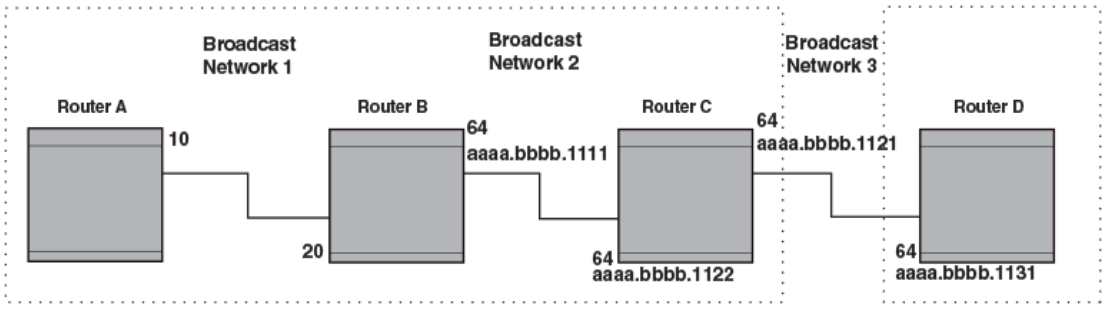
- The Level-1 IS that has the highest priority becomes the Level-1 Designated IS for the broadcast network.
- The Level-2 IS that has the highest priority becomes the Level-2 Designated IS for the broadcast network.

If the Designated IS becomes unavailable (for example, is rebooted), the IS with the next highest priority becomes the new IS. If two or more ISs have the highest priority, the IS with the highest MAC address becomes the Designated IS.

The priority is an interface parameter. Each interface that is enabled for IS-IS can have a different priority.

Figure 18 shows an example of the results of Designated IS elections. For simplicity, this example shows four of the five routers in Intermediate systems and end systems on page 258, with the same domain and areas.

FIGURE 18 Each broadcast network has a Level-1 Designated IS and a Level-2 Designated IS



Designated IS election has the following results in this network topology:

- Router B is the Level-1 Designated IS for broadcast network 1
- Router C is the Level-1 Designated IS for broadcast network 2
- Router D is the Level-2 Designated IS for broadcast network 3

In this example, the IS-IS priorities for the IS-IS interfaces in broadcast network 1 have been changed by an administrator. The priorities for the interfaces in the other broadcast networks are still set to the default (64). When there is a tie, IS-IS selects the interface with the highest MAC address.

Broadcast pseudonode

In a broadcast network, the Designated IS maintains and distributes link state information to other ISs by maintaining a pseudonode . A pseudonode is a logical host representing all the Level-1 or Level-2 links among the ISs in a broadcast network. Level-1 and Level-2 have separate pseudonodes, although the same device can be the pseudonode for Level-1 and Level-2.

Route calculation and selection

The Designated IS uses a Shortest Path First (SPF) algorithm to calculate paths to destination ISs and ESs. The SPF algorithm uses Link State PDUs (LSPDUs) received from other ISs as input, and creates the paths as output.

After calculating the paths, the Designated IS then selects the best paths and places them in the IS-IS route table. The Designated IS uses the following process to select the best paths.

1. Prefer the Level-1 path over the Level-2 path.
2. If there is no Level-1 path, prefer the internal Level-2 path over the external Level-2 path.
3. If there is still more than one path, prefer the path with the lowest metric.
4. If there is more than one path with the lowest metric, load share among the paths.

After selecting the best path to a destination, the software places the path in the IS-IS route table.

Three-way handshake for point-to-point adjacencies

Support was provided for Three-Way Handshake for Point-to-Point adjacencies as described in RFC 3373. This feature provides three-way handshake mechanisms on point-to-point interfaces for the following benefits:

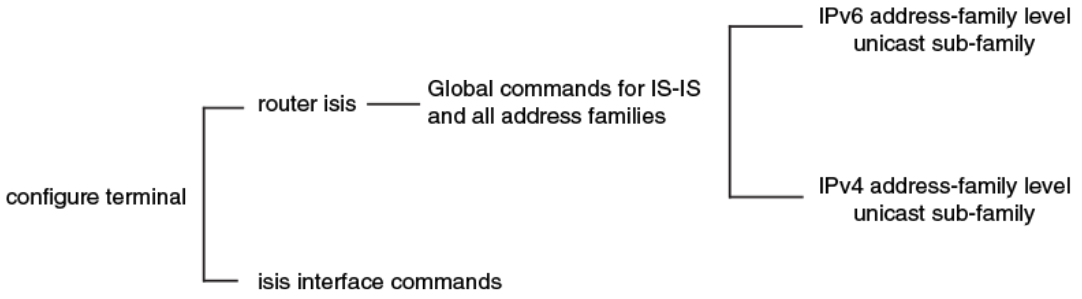
- Identifies neighbor restarts within the holding time period
- Identifies uni-directional link failures and stops forming of an adjacency with a peer where such link failures occur.

NOTE
This feature is the default operation and cannot be turned off. Brocade devices with this feature are fully backward compatible with Brocade devices running an earlier release.

IS-IS CLI levels

The CLI includes various levels of commands for IS-IS. Figure 19 diagrams these levels.

FIGURE 19 IS-IS CLI levels



The IS-IS CLI levels are as follows:

- A global level for the configuration of the IS-IS protocol. At this level, all IS-IS configurations at this level apply to IPv4 and IPv6. You enter this layer using the **router isis** command.
  - Under the global level, you specify an address family. Address families to separate the IS-IS configurations for IPv4 and IPv6. You enter configurations that are for a specific You enter this level by entering the **address-family** command at the router isis level.
  - Under the address family level, you select a sub-address family, which is the type of routes for the configuration. For IS-IS, you specify **unicast**.
- An interface level.

Global configuration level

You enter the global configuration level of IS-IS by entering the following command.

```
device(config)#router isis
device(config-isis-router)#
```

Syntax: [no] router isis

The **config-isis-router#** prompt indicates that you are at the global level for IS-IS. A configuration that you enter at this level applies to both IS-IS IPv4 and IS-IS IPv6.

Address family configuration level

The Brocade device's implementation of IS-IS includes the address family configuration level. Address families allow you to configure IPv4 IS-IS unicast settings that are separate and distinct from IPv6 IS-IS unicast settings (when IPv6 is supported).

Under the address family level, Brocade devices currently support the unicast address family configuration level only. The Brocade device enters the IPv4 IS-IS unicast address family configuration level when you enter the following command while at the global IS-IS configuration level.

```
device(config-isis-router)# address-family ipv4 unicast
device(config-isis-router-ipv4u)#
```

Syntax: address-family ipv4 unicast

The **(config-isis-router-ipv4u)#** prompt indicates that you are at the IPv4 IS-IS unicast address family configuration level. While at this level, you can access several commands that allow you to configure IPv4 IS-IS unicast settings.

NOTE
Each address family configuration level allows you to access commands that apply to that particular address family only. To enable a feature in a particular address family, you must specify any associated commands for that feature in that particular address family. You cannot expect the feature, which you may have configured in the IPv4 IS-IS unicast address family, to work in the IPv6 IS-IS unicast address family unless it is explicitly configured in the IPv6 IS-IS unicast address family.

To exit from the ipv4 IS-IS unicast address family configuration level, enter the following command.

```
device(config-isis-router-ipv4u)# exit-address-family
device(config-isis-router)#
```

Entering this command returns you to the global IS-IS configuration level.

Interface level

Some IS-IS definitions are entered at the interface level. To enable IS-IS at the interface level, enter the following command.

```
device(config)# interface ethernet 2/3
device(config-if-e1000-2/3)#ip router isis
```

Syntax: [no] ip router isis

Enabling IS-IS globally

To configure IPv4 IS-IS, perform the tasks listed below.

1. Globally enable IS-IS by entering the following command.

   ```
   device(config)# router isis
   ISIS: Please configure NET!
   ```

   Once you enter **router isis** , the device enters the IS-IS router configuration level.

   Syntax: [no] router isis

   To disable IS-IS, use the **no** form of this command.
2. If you have not already configured a NET for IS-IS, enter commands such as the following.

   ```
   device(config-isis-router)# net 49.2211.0000.00bb.cccc.00
   device(config-isis-router)#
   ```

   The commands in the example above configure a NET that has the area ID 49.2211, the system ID 0000.00bb.cccc (the device's base MAC address), and SEL value 00.

   Syntax: [no] net *area-id.system-id.sel*

The *area-id* parameter specifies the area and has the format xx or xx.xxxx. For example, 49 and 49.2211 are valid area IDs.

The *system-id* parameter specifies the Brocade device's unique IS-IS router ID and has the format xxxx.xxxx.xxxx. You can specify any value for the system ID. A common practice is to use the device's base MAC address as the system ID. The base MAC address is also the MAC address of port 1. To determine the base MAC address, enter the following command at any level of the CLI: **show interfaces brief** . The base MAC address is listed in the first row of information, in the MAC column.

You must use the same system ID in all the NETs on the Brocade device.

NOTE
The parameter descriptions above are the recommended values for the NET. However, the CLI accepts any value that fits within the following lengths and formats: xx.xxxx.xxxx.xxxx.00 - minimum length of NETxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.00 - maximum length of NET

The *sel* parameter specifies the NSAP Selector (SEL). This value must always be 00 (two zeros). The value 00 indicates that this address is an NET.

To delete a NET, use the **no** form of this command.

3. Configure IS-IS parameters. Refer to the sections Globally configuring IS-IS on a device on page 264, Configuring IPv4 address family route parameters on page 275, and Configuring ISIS properties on an interface on page 294.

   None of the IS-IS parameters require a software reload to places changes into effect and most parameter changes take effect immediately. However, changes for the following parameters take effect only after you disable and then re-enable redistribution:

   - Change the default metric.
   - Add, change, or negate route redistribution parameters.

   Some IS-IS parameter changes take effect immediately while others do not take full effect until you disable, then re-enable route redistribution.

Globally configuring IS-IS on a device

This section describes how to change the global IS-IS parameters. These parameter settings apply to both IS-IS IPv4 and IS-IS IPv6.

Setting the overload bit

If an IS's resources are overloaded and are preventing the IS from properly performing IS-IS routing, the IS can inform other ISs of this condition by setting the overload bit in LSPDUs sent to other ISs from 0 (off) to 1 (on).

When an IS is overloaded, other ISs will not use the overloaded IS to forward traffic. An IS can be in the overload state for Level-1, Level-2, or both as described in the following section:

- If an IS is in the overload state for Level-1, other Level-1 ISs stop using the overloaded IS to forward Level-1 traffic. However, the IS can still forward Level-2 traffic, if applicable.
- If an IS is in the overload state for Level-2, other Level-2 ISs stop using the overloaded IS to forward Level-2 traffic. However, the IS can still forward Level-1 traffic, if applicable.
- If an IS is in the overload state for both levels, the IS cannot forward traffic at either level.

By default, the Brocade device automatically sets the overload bit to 1 (on) in its LSPDUs to other ISs if an overload condition occurs.

You can set the overload bit on to administratively shut down IS-IS without disabling the protocol. Setting the overload bit on is useful when you want to make configuration changes without removing the Brocade device from the network.

In addition, you can configure the Brocade device to set the overload bit on for a specific number of seconds during startup, to allow IS-IS to become fully active before the device begins IS-IS routing. By default, there is no delay (0 seconds).

To immediately set the overload bit on, enter the following command.

```
device(config-isis-router)# set-overload-bit
```

This command administratively shuts down IS-IS by configuring the Brocade device to immediately set the overload bit to 1 (on) in all LSPs sent to other ISs.

To configure the Brocade device to temporarily set the overload bit on after a software reload, enter a command such as the following.

```
device(config-isis-router)# set-overload-bit on-startup 5
```

This command configures the Brocade device to set the overload bit on in all its IS-IS LSPs sent to other ISs during the first five seconds following a successful software reload. After the five seconds expire, the Brocade device resets the overload bit to off in all its IS-IS LSPs.

Syntax: [no] set-overload-bit [**on-startup** *secs*]

The **on-startup***secs* parameter specifies the number of seconds following a reload to set the overload bit on. You can specify a number from 5 - 86400 (24 hours).

A new option has been added to the **set-overload-bit** command to prevent route black holing in support of RFC 3277. With this option set, the behavior of IS-IS will be changed during a device reboot. During a device reboot, IS-IS sets the overload bit in its LSPDUs until BGP has converged.

This feature is configured using the **set-overload-bit** command as shown in the following.

```
device(config-isis-router)# set-overload-bit on-startup wait-for-bgp 1000
```

Syntax: [no] set-overload-bit on-startup wait-for-bgp *max-bgp-wait-time*

The *max-bgp-wait-time* variable is the maximum time IS-IS will wait for BGP convergence to complete. Once this time has been exceeded without BGP converging, IS-IS will exit the overload state. The default value is 600 seconds (10 minutes), possible values range: 5 to 86400 seconds.

Configuring authentication

By default, a Brocade device does not authenticate packets sent to or received from an end system (ES) or other intermediate system (IS). In previous releases, the Multi-Service IronWare software let you configure area, domain, and circuit passwords to direct the Brocade device to check for a password in packets sent from the device.

The new method of configuring an authentication password introduces the option of using the Hashed Message Authentication codes - Message Digest 5 (HMAC-MD5) algorithm.

This implementation is in conformance with RFC 3567 - Intermediate System to Intermediate System (IS-IS) Cryptographic Authentication.

NOTE
The commands for setting the password used in previous versions of the Multi-Service IronWare software are now hidden in the CLI, however they are backward compatible and will operate in this release.

Configuring IS-IS authentication at the Router IS-IS mode

To configure IS-IS authentication at the Router IS-IS mode on a Brocade device, you must perform the following tasks:

- Configure IS-IS Authentication Mode
- Configure IS-IS Authentication Key
- Disable IS-IS Authentication Check (optional)

Configuring IS-IS authentication mode

The following commands configure the IS-IS for the authentication mode.

```
device(config)# router isis
device(config-isis-router)# auth-mode md5 level-1
```

Syntax: [no] auth-mode [**cleartext** | **md5**] [**level-1** | **level-2**]

The **cleartext** parameter specifies that the IS-IS PDUs will be authenticated using a cleartext password.

The **md5** parameter specifies that the IS-IS PDUs will be authenticated using the Hashed Message Authentication codes - Message Digest 5 (HMAC-MD5) algorithm.

The **level-1** parameter specifies that the authentication type-length-value (TLV) tuple be added to the L1 LSP, L1 CSNP, and LI PSNP packets.

The **level-2** parameter specifies that the authentication TLV tuple be added to the L2 LSP, L2 CSNP, and L2 PSNP packets.

NOTE
If the IS-IS interface is configured for point-to-point, the level-1 interface-level IS-IS authentication configuration is applied.

Configuring IS-IS authentication key

The following commands configure an authentication key to be used with the mode specified in Configuring IS-IS authentication mode on page 266.

```
device(config)# router isis
device(config-isis-router)# auth-mode md5 level-1
device(config-isis-router)# auth-key supervisor level-1
device(config-isis-router)# auth-key supervisor level-2
```

Syntax: [no] auth-key *string* [**level-1** | **level-2**]

The *string* variable specifies a text string that is used as an authentication password. The authentication mode must be configured before this value can be configured.

By default, the authentication key is encrypted. If you want the authentication key to be in clear text, insert a **0** between **auth-key** and *string*.

```
device(config-isis-router)# auth-key 0 supervisor level-1
```

The software adds a prefix to the authentication key string in the configuration. For example, the following portion of the code has the encrypted code "2".

```
auth-key 2
 $on-n level-1
```

The prefix can be one of the following:

- 0 = the key string is not encrypted and is in clear text
- 1 = the key string uses proprietary simple cryptographic 2-way algorithm (only for Brocade NetIron CES Series)
- 2 = the key string uses proprietary base64 cryptographic 2-way algorithm (only for Brocade NetIron XMR Series and Brocade NetIron MLX Series devices)

The **level-1** parameter specifies that the authentication key specified here is used to authenticate the L1 LSP, L1 CSNP and LI PSNP packets.

The **level-2** parameter specifies that the authentication key specified here is used to authenticate the L2 LSP, L2 CSNP and L2 PSNP packets.

You must enter a configuration for both level-1 and level-2 in order to enter the auth-key string.

NOTE
If the authentication mode is reset for the level specified, the authentication key must also be reset.

Disabling IS-IS authentication checking

When transitioning from one authentication mode to another, changing the authentication mode can cause packets to drop because only some of the routers have been reconfigured. During such a transition, it can be useful to disable IS-IS authentication checking temporarily until all routers are reconfigured and the network is stable.

You can use the following commands to disable IS-IS authentication checking.

```
device(config)# router isis
device(config-isis-router)# no auth-check level-1
```

Syntax: [no] auth-check [**level-1** | **level-2**]

This command enables and disables IS-IS authentication checking. The default is enabled and the **no** parameter disables authentication checking.

The **level-1** parameter specifies that authentication checking is enabled/ disabled for L1 LSP, L1 CSNP and LI PSNP packets.

The **level-2** parameter specifies that authentication checking is enabled/disabled for L2 LSP, L2 CSNP and L2 PSNP packets.

Configuring IS-IS MD5 authentication on a specified interface

To configure IS-IS MD5 authentication on a specified interface on a Brocade device, you must perform the following tasks:

- Configure IS-IS Interface Authentication Mode for a Specified Interface
- Configure IS-IS Authentication Key on the Interface
- Disable IS-IS Authentication Check on an Interface (optional)

Configuring IS-IS authentication mode for a specified interface

The following commands configure the IS-IS for the authentication mode on a specified interface.

```
device(config)# interface ethernet 3/1
device(config-if-e10000-3/1)# isis auth-mode md5 level-1
```

Syntax: [no] isis auth-mode [**cleartext** | **md5**] [**level-1** | **level-2**]

The **cleartext** parameter specifies that the IS-IS PDUs will be authenticated using a cleartext password.

The **md5** parameter specifies that the IS-IS PDUs authenticated using the Hashed Message Authentication codes - Message Digest 5 (HMAC-MD5) algorithm.

The **level-1** parameter specifies that the authentication TLV tuple be added to the L1 Hello packets.

The **level-2** parameter specifies that the authentication TLV tuple be added to the L2 Hello packets.

NOTE
If either level-1 or level-2 are not specified, the configuration is applied to both level-1 and level-2.

Configuring an IS-IS authentication key for a specified interface

The following commands configure an authentication key to be used with the mode specified in Configuring IS-IS authentication mode for a specified interface on page 268.

```
device(config)# interface ethernet 3/1
device(config-if-e10000-3/1)# isis auth-key supervisor level-1
```

Syntax: [no] isis auth-key *key* [**level-1** | **level-2**]

The **key** value specifies a text string that is used as an authentication password. The authentication mode must be configured before this value can be configured.

The **level-1** parameter specifies that the authentication key specified here is used to authenticate the L1 Hello packets.

The **level-2** parameter specifies that the authentication key specified here is used to authenticate the L2 Hello packets.

NOTE
If either level-1 or level-2 are not specified, the configuration is applied to both level-1 and level-2.

NOTE
If the authentication mode is reset for the level specified, the authentication key must also be reset.

NOTE
The**isis auth-key** command allows the user to configure more 80 characters, but only the first 80 characters are used.

Disabling IS-IS authentication checking on a specified interface

When transitioning from one authentication mode to another, changing the authentication mode can cause packets to drop because only some of the routers have been reconfigured. During such a transition, it can be useful to disable IS-IS authentication checking temporarily until all routers are reconfigured and the network is stable.

You can use the following commands to disable IS-IS authentication checking on a specified interface.

```
device(config)# interface ethernet 3/1
device(if-e10000-3/1)# no isis auth-check level-1
```

Syntax: [no] isis auth-check [**level-1** | **level-2**]

This command enables and disables IS-IS authentication checking. The default is enabled and the **no** parameter disables authentication checking.

The **level-1** parameter specifies that authentication checking is enabled/ disabled for L1 Hello packets.

The **level-2** parameter specifies that authentication checking is enabled/disabled for L2 Hello packets.

NOTE
If either level-1 or level-2 are not specified, the configuration is applied to both level-1 and level-2.

Changing the IS-IS level globally

By default, a Brocade device can operate as both a Level-1 and IS-IS Level-2 router. To globally change the level supported from Level-1 and Level-2 to Level-1 only, enter the following command.

```
device(config-isis-router)# is-type level-1
```

Syntax: [no] is-type level-1 | **level-1-2** | **level-2**

The **level-1 | level-1-2 | level-2** parameter specifies the IS-IS type. If you want to re-enable support for both IS-IS types, re-enter the command you entered to change the IS-IS type, and use "no" in front of the command.

To change the IS-IS on an interface, refer to Changing the IS-IS level on an interface on page 295.

Disabling or re-enabling display of hostname

Brocade's implementation of IS-IS supports RFC 2763, which describes a mechanism for mapping IS-IS system IDs to the hostnames of the devices with those IDs. For example, if you set the hostname on the Brocade device to "IS-IS Router 1", the mapping feature uses this name instead of the Brocade device's IS-IS system ID in the output of the following commands:

- **show isis database**
- **show isis interface**
- **show isis neighbor**

The Brocade device's hostname is displayed in each CLI command prompt, for example.

```
device(config-isis-router)#
```

The name mapping feature is enabled by default. If you want to disable name mapping, enter the following command.

```
device(config-isis-router)# no hostname
```

Syntax: [no] hostname

To display the name mappings, enter the **show isis hostname** command.

Changing the Sequence Numbers PDU interval

A Complete Sequence Numbers PDU (CSNP) is a complete list of the LSPs in the Designated IS' link state database. The CSNP contains a list of all the LSPs in the database, as well as other information that helps IS neighbors determine whether their LSP databases are in sync with one another. The Designated IS sends CSNPs to the broadcast interface. Level-1 and Level-2 each have their own Designated IS.

A Partial Sequence Numbers PDU (PSNP) is a partial list of LSPs. ISs other than the Designated IS (that is, the non-Designated ISs) send PSNPs to the broadcast interface.

The CSNP interval specifies how often the Designated IS sends a CSNP to the broadcast interface. Likewise, the PSNP interval specifies how often other ISs (non-Designated ISs) send a PSNP to the broadcast interface.

The interval you can configure on the Brocade device applies to both Level-1 and Level-2 CSNPs and PSNPs. The default interval is 10 seconds. You can set the interval to a value from 0 - 65535 seconds.

To change the interval, enter a command such as the following.

```
device(config-isis-router)# csnp-interval 15
```

Syntax: [no] csnp-interval *secs*

The *secs* parameter specifies the interval and can be from 0 - 65535 seconds. The default is 10 seconds.

NOTE
PSNP has a default interval of 2 seconds and is not configurable.

Changing the maximum LSP lifetime

The maximum LSP lifetime is the maximum number of seconds an unrefreshed LSP can remain in the Brocade device's LSP database. The maximum LSP lifetime can be from 1 - 65535 seconds. The default is 1200 seconds (20 minutes).

To change the maximum LSP lifetime to 2400 seconds, enter a command such as the following.

```
device(config-isis-router)# max-lsp-lifetime 2400
```

Syntax: [no] max-lsp-lifetime *secs*

The *secs* parameter specifies the maximum LSP lifetime and can be from 1 - 65535 seconds. The default is 1200 seconds (20 minutes).

NOTE
The **max-lsp-lifetime** and the **lsp-refresh-interval** must be set in such a way that the LSPs are refreshed before the **max-lsp-lifetime** expires; otherwise, the Brocade device's originated LSPs may be timed out by it's neighbors. Refer to Changing the LSP refresh interval on page 271.

Changing the LSP refresh interval

The LSP refresh interval is the maximum number of seconds the Brocade device waits between sending updated LSPs to its IS-IS neighbors. The interval can be from 1 - 65535 seconds. The default is 900 seconds.

To change the LSP refresh interval to 20000 seconds, enter a command such as the following.

```
device(config-isis-router)# lsp-refresh-interval 20000
```

Syntax: [no] lsp-refresh-interval *secs*

The *secs* parameter specifies the maximum refresh interval and can be from 1 - 65535 seconds. The default is 900 seconds (15 minutes).

Changing the LSP generation interval

The LSP generation interval is the minimum number of seconds the Brocade device waits between sending updated LSPs to its IS-IS neighbors. The interval can be from 1 - 120 seconds. The default is 10 seconds.

To change the LSP generation interval to 45 seconds, enter a command such as the following.

```
device(config-isis-router)# lsp-gen-interval 45
```

Syntax: [no] lsp-gen-interval *secs*

The *secs* parameter specifies the minimum refresh interval and can be from 1 - 120 seconds. The default is 10 seconds.

Changing the LSP interval and retransmit interval

You LSP interval is the rate of transmission, in milliseconds of the LSPs. The retransmit interval is the time the device waits before it retransmits LSPs. To define an LSP interval, enter a command such as the following.

```
device(config-isis-router)# lsp-interval 45
```

Syntax: [no] lsp-interval *milliseconds*

Enter 1 - 4294967295 milliseconds for the LSP interval. The default is 33 milliseconds.

To define an interval for retransmission of LSPs enter a command such as the following.

```
device(config-isis-router)# retransmit-interval 3
```

Syntax: [no] retransmit-interval *seconds*

Enter 0 - 65535 seconds for the retransmission interval. The default is 5 seconds.

Changing the SPF timer

Every IS maintains a Shortest Path First (SPF) tree, which is a representation of the states of each of the IS's links to ESs and other ISs. If the IS is both a Level-1 and Level-2 IS, it maintains separate SPF trees for each level.

To ensure that the SPF tree remains current, the IS updates the tree at regular intervals following a change in network topology or the link state database. By default, the Brocade device recalculates its

IS-IS tree every five seconds following a change. You can change the SPF timer to a value from 1 - 120 seconds.

NOTE
This command has been superseded by the IS-IS PSPF Exponential back-off feature.

To change the SPF interval, enter a command such as the following.

```
device(config-isis-router)# spf-interval 30
```

Syntax: [no] spf-interval *secs*

The *secs* parameter specifies the interval and can be from 1 - 120 seconds. The default is 5 seconds.

Configuring the IS-IS PSPF exponential back-off feature

The Brocade device uses the exponential back-off mechanism to provide a more responsive approach to running the PSPF calculations. With this new feature, there is a new configurable command called **partial-spf-interval** that allows you to schedule PSPF processing as described in the following.

An **initial-wait** interval can be configured as a wait time after an LSP change until the first PSPF calculation. Optionally, this value is followed by another configurable variable called the **second-wait** interval that is used as a wait time between the first and second PSPF calculations. The **second-wait** interval (if configured) is then increased in multiples of 2 until it reaches the maximum hold time as configured by the **max-wait** variable. Once reached, the maximum hold time remains the hold interval between PSPF calculations until there are no further changes in the network. When there are no network changes in a hold down period, the gap between PSPF calculations returns to the **initial-wait** interval and the process begins again.

If an **initial-wait** interval is configured without a **second-wait** interval, the **max-wait** variable is used for the second and all subsequent intervals.

If the **initial-wait** and **second-wait** intervals are not configured, the **max-wait** variable is used for the first and all subsequent intervals.

The IS-IS PSPF exponential back-off mechanism is configured using the **partial-spf-interval** command, as shown in the following.

```
device(config-isis-router)# partial-spf-interval 60 1000 5000
```

Syntax: [no] partial-spf-interval *max-wait initial-waitsecond-wait*

The *max-wait* variable specifies the maximum interval between PSPF recalculations. The range of acceptable values is 0 - 120000 milliseconds. The default is 5000 milliseconds (5 seconds).

The *initial-wait* variable is an optional value that specifies the wait time after an LSP change until the first PSPF calculation. The range of acceptable values is 0 - 120000 milliseconds. The default for this variable is value of the **max-wait** time.

The *second-wait* variable is an optional value that specifies the wait time between the first and second PSPF calculations. If this optional value is configured, it will be doubled with each PSPF recalculation until the value is equal to the *spf-max-wait* value. The range of acceptable values is 0 - 120000 milliseconds. The default for this variable is value of the **max-wait** time.

Configuring the IS-IS flooding mechanism

The IS-IS fast flooding feature allows you to configure IS-IS on the router to flood Link State PDUs to other routers in the network before running SPF. This improves database synchronization by allowing

LSP changes to be propagated to neighbors before running SPF. The IS-IS fast-flood feature is implemented using the fast-flood command as shown in the following.

```
device(config-isis-router)# fast-flood 10
```

Syntax: [no] fast-flood *lsp-count*

The*lsp-count* variable sets the number of LSPs that trigger SPF that must be flooded before running SPF. The SPF run will be delayed until the configured number of LSPs have been flooded. If the number of changed LSPs is less than the configured number, then only the changed LSPs are flooded. The variable can be set to the following values: 1 - 25. This variable is optional and will be set to a value of 4 if not specified.

Globally disabling or re-enabling hello padding

By default, the Brocade device adds extra data to the end of a hello packet to make the packet the same size as the maximum length of PDU the Brocade device supports.

The padding applies to the following types of hello packets:

- ES hello (ESH PDU)
- IS hello (ISH PDU)
- IS to IS hello (IIH PDU)

The padding consists of arbitrarily valued octets. A padded hello PDU indicates the largest PDU that the Brocade device can receive. Other ISs that receive a padded hello PDU from the Brocade device can therefore ensure that the IS-IS PDUs they send the Brocade device. Similarly, if the Brocade device receives a padded hello PDU from a neighbor IS, the Brocade device knows the maximum size PDU that the Brocade device can send to the neighbor.

When padding is enabled, the maximum length of a Hello PDU sent by the Brocade device is 1514 bytes.

If you need to disable padding, you can do so globally or on individual interfaces. Generally, you do not need to disable padding unless a link is experiencing slow performance. If you enable or disable padding on an interface, the interface setting overrides the global setting.

To globally disable padding of IS-IS hello PDUs, enter the following command.

```
device(config-isis-router)# no hello padding
```

This command disables all hello PDU padding on the Brocade device. To re-enable padding, enter the following command.

```
device(config-isis-router)# hello padding
```

Syntax: [no] hello padding [**point-to-point**]

By default, hello padding is enabled. Enter the **no** form of the command to disable hello padding.

The **point-to-point** option enables hello PDU padding on Point-to-Point interfaces.

To disable hello padding on an interface, refer to Disabling and enabling hello padding on an interface on page 296.

Logging adjacency changes

The Brocade device can be configured to log changes in the status of an adjacency with another IS. Logging of the adjacency changes is disabled by default. To enable or disable them, use either of the following methods.

To enable logging of adjacency changes, enter the following command.

```
device(config-isis-router)# log adjacency
```

Syntax: [no] log adjacency

To disable logging of adjacency changes, enter the following command.

```
device(config-isis-router)# no log adjacency
```

Logging invalid LSP packets received

The Brocade device can be configured to provide logging of invalid LSP packets. Logging of the invalid LSP packets is disabled by default. To enable or disable this function, use either of the following methods.

To enable logging of invalid LSP packets, enter the following command.

```
device(config-isis-router)# log invalid-lsp-packets
```

Syntax: [no] log invalid-lsp-packets

To disable logging of invalid LSP packets, enter the following command.

```
device(config-isis-router)# no log invalid-lsp-packets
```

Disabling partial SPF optimizations

IS-IS employs certain partial SPF optimizations to make partial changes to the routing table in network change situations where the topology of the network has not changed but where there may be changes in the IP networks advertised by routers. These optimizations are termed partial SPF optimizations.

You can optionally configure IS-IS to perform a full SPF calculation when any network (non-topology) change occurs by using the **disable-partial-spf-opt** command. When **disable-partial-spf-opt** is configured, IS-IS always runs full SPF for all such network changes.

To disable partial SPF calculations for IS-IS, enter the following command.

```
device(config-isis-router)# disable-partial-spf-opt
```

Syntax: [no] disable-partial-spf-opt

To restore partial SPF optimizations, use the **no** form of this command.

Disabling incremental SPF optimizations

In the event of certain topology changes (for instance non-local adjacency flaps), IS-IS employs incremental SPF optimizations to efficiently update the routing table. An incremental SPF is faster and takes fewer CPU cycles than a full SPF.

You can optionally configure IS-IS to perform a full SPF calculation when any network topology change occurs by using the **disable-incremental-spf-opt** command. When **disable-incremental-spf-opt** is configured, IS-IS always runs full SPF for all such network topology changes.

To disable incremental SPF optimizations for IS-IS, enter the following command.

```
device(config-isis-router)# disable-incremental-spf-opt
```

Syntax: [no] disable-incremental-spf-opt

To restore incremental SPF optimizations, use the **no** form of this command.

NOTE
If you disable the partial SPF optimizations (by using the **disable-partial-spf-opt** command), IS-IS automatically disables the incremental SPF optimizations and always runs full SPF, too. However, the reverse is not true: disabling incremental SPF optimizations does not disable partial optimizations.

IS-IS incremental shortcut LSP SPF optimization

IS-IS can be configured to use an incremental shortcut LSP SPF optimization algorithm. Incremental shortcut LSP SPF optimization is more efficient when updating the routes in cases where the shortcut LSP state change does not influence the topology. Incremental Shortcut LSP SPF Optimizations are on by default.

NOTE
If you disable the partial SPF optimizations (by using the **disable-partial-spf-opt** command), IS-IS automatically disables the incremental SPF optimizations and always runs full SPF, too. However, the reverse is not true: disabling incremental SPF optimizations does not disable partial optimizations.

Configuration considerations

Incremental Shortcut SPF optimizations will not be applicable to LSP shortcuts with metrics configured on them

Incremental Shortcut SPF optimizations will not be applicable to LSP shortcuts with negative relative metrics configured.

Incremental Shortcut SPF optimizations will not be applicable to announced LSP shortcuts.

DIsabling IS-IS Incremental Shortcut LSP SPF Optimization

To disable incremental shortcut LSP SPF optimization, enter the following commands at the global configuration mode.

```
device((config)#router isis
device(config-isis-router)#disable-inc-stct-spf-opt
```

Syntax: [no] disable--inc-stct-spf-opt

To restore incremental shortcut LSP SPF optimization, use the **no** form of this command.

Configuring IPv4 address family route parameters

This section describes how to modify the IS-IS parameters for the IS-IS IPv4 unicast address family. To enter the IPv4 unicast address family, refer to the Address family configuration level on page 262.

Changing the metric style

The metric style specifies the Types, Lengths, and Values (TLVs) an IS-IS LSP can have. The TLVs specify the types of data, the maximum length of the data, and the valid values for the data. One of the types of data the TLVs control is a route's default-metric. By default, the Brocade device uses the standard IS-IS TLVs, which allows metric values from 1 - 63. The default metric style is called "narrow." You can increase the range of metric values supported by the Brocade device by changing the metric style to wide. The wide metric style allows metric values in the range 1 - 16777215.

To change the metric style to wide, enter the following command.

```
device(config-isis-router-ipv4)# metric-style wide
```

This command changes the metric style for both Level-1 and Level-2.

Syntax: [no] metric-style wide [**level-1** | **level-2**]

The **level-1 | level-2** parameter specifies the levels to which the change applies. If not specified, the changes are applied to both levels.

Changing the maximum number of load sharing paths

By default, IPv4 IS-IS can calculate and install four equal-cost paths into the IPv4 forwarding table. You can change the number of paths IPv4 IS-IS can calculate and install in the IPv4 forwarding table to a value from 1 to 32. If you change the number of paths to one, the Brocade device does not load share multiple route paths learned from IPv4 IS-IS.

NOTE
The maximum number of paths supported by the BR-MLX-10Gx24-DM module is 16.

For example, to change the number of paths IPv4 IS-IS can calculate and install in the IPv4 forwarding table to three, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv4u)# maximum-paths 3
```

Syntax: [no] maximum-paths *number*

The *number* parameter specifies the number of paths IPv4 IS-IS can calculate and install in the IPv4 forwarding table. The *number* value range is 2 to 32.

NOTE
The value specified in *number* is limited by the IP load-sharing value specified in the **ip load-sharing** command.

To return to the default number of maximum paths, enter the **no** form of this command.

Enabling advertisement of a default route

By default, the Brocade device does not generate or advertise a default route to its neighboring ISs. A default route is not advertised even if the device's IPv4 route table contains a default route. You can enable the device to advertise a default route to all neighboring ISs using one of the following methods. By default, the feature originates the default route at Level 2 only. However, you can apply a route map to originate the default route to Level 1 only or at both Level 1 and Level 2.

NOTE
This feature requires the presence of a default route in the IPv4 route table.

To enable the Brocade device to advertise a default route that is originated a Level 2, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv4u)# default-information-originate
```

This command enables the device to advertise a default route into the IPv4 IS-IS area to which the device is attached.

Syntax: [no] default-information-originate [**route-map** *name*]

The **route-map***name* parameter allows you to specify the level on which to advertise the default route. You can specify one of the following:

- Advertise to Level-1 ISs only.
- Advertise to Level-2 ISs only.
- Advertise to Level-1 and Level-2 ISs.

NOTE
The route map must be configured before you can use the route map as a parameter with the **default-information-originate** command.

To use a route map to specify the router to advertise a default route to Level 1, enter commands such as the following at the Global CONFIG level.

```
device(config)# route-map default_level1 permit 1
device(config-routemap default_level1)# set level level-1
device(config-routemap default_level1)# exit
device(config)# router isis
device(config-isis-router)# address-family ipv4 unicast
device(config-isis-router-ipv4u)# default-information-originate route-map
default_level1
```

These commands configure a route map to set the default advertisement level to Level 1 only.

Syntax: [no] route-map *map-name* **permit** | **deny** *sequence-number*

Syntax: [no] set level level-1 | **level-1-2** | **level-2**

For this use of a route map, use the **permit** option and do not specify a **match** statement. Specify a **set** statement to set the level to one of the following:

- **level-1** - Level 1 only.
- **level-1-2** - Level 1 and Level 2.
- **level-2** - Level 2 only (default).

Matching based on IS-IS protocol type

The**match** option has been added to the **route-map** command that allows IS-IS routes to be matched based on level-1 or level-2 or all IS-IS routes.

```
device(config-routemap test)# match protocol isis level-1
```

Syntax: [no] match protocol isis { **level-1** | **level-2** }

The **match protocol isis level-1** option can be used to match the IS-IS Level-1 routes.

The **match protocol isis level-2** option can be used to match the IS-IS Level-2 routes.

Changing the administrative distance for IPv4 IS-IS

When the Brocade device has paths from multiple routing protocols to the same destination, it compares the administrative distances of the paths and selects the path with the lowest administrative distance to place in the IPv4 route table.

For example, if the Brocade device has a path from RIP, from OSPF, and IPv4 IS-IS to the same destination, and all the paths are using their protocols' default administrative distances, the device selects the OSPF path, because that path has a lower administrative distance than the RIP and IPv4 IS-IS paths.

Here are the default IPv4 administrative distances on the Brocade device:

- Directly connected - 0 (this value is not configurable)
- Static - 1 (applies to all static routes, including default routes)
- EBGP - 20
- OSPF - 110
- IPv4 IS-IS - 115
- RIP - 120
- IBGP - 200
- Local BGP - 200
- Unknown - 255 (the device will not use this route)

Lower administrative distances are preferred over higher distances. For example, if the Brocade device receives routes for the same network from IPv4 IS-IS and from RIP, it will prefer the IPv4 IS-IS route by default.

To change the administrative distance for IPv4 IS-IS routes, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv4u)# distance 100
```

Syntax: [no] distance *number*

This command changes the administrative distance for all IPv4 IS-IS routes to 100.

The *number* parameter specifies the administrative distance. You can specify a value from 1 - 255. (Routes with a distance value of 255 are not installed in the routing table.) The default for IPv4 IS-IS is 115.

Configuring summary addresses

You can configure summary addresses to aggregate IS-IS route information. Summary addresses can enhance performance by reducing the size of the Link State database, reducing the amount of data the Brocade device needs to send to its neighbors, and reducing the CPU cycles used for IS-IS.

When you configure a summary address, the address applies only to Level-2 routes by default. You can specify Level-1 only, Level-2 only, or Level-1 and Level-2 when you configure the address.

To configure a summary address, enter a command such as the following.

```
device(config-isis-router-ipv4u)# summary-address 192.168.0.0 255.255.0.0
```

This command configures a summary address for all Level-2 IS-IS route destinations between 192.168.1.0 - 192.168.255.255.

Syntax: [no] summary-address *ip-addrsubnet-mask* [| **evel-1** | **level-1-2** | **level-2**]

The *ip-addr subnet-mask* parameters specify the aggregate address. The mask indicates the significant bits in the address. Ones are significant, and zeros allow any value. In the command example above, the mask 255.255.0.0 matches on all addresses that begin with 192.168 and contain any values for the final two octets.

The **level-1 | level-1-2 | level-2** parameter specifies the route types to which the aggregate route applies. The default is **level-2**.

Redistributing routes into IPv4 IS-IS

To redistribute routes into IPv4 IS-IS, you can perform the following configuration tasks:

- Change the default redistribution metric (optional).
- Configure the redistribution of a particular route type into IPv4 IS-IS (mandatory).

The Brocade device can redistribute routes from the following route sources into IPv4 IS-IS:

- BGP4+.
- RIP.
- OSPF.
- Static IPv4 routes.
- IPv4 routes learned from directly connected networks.

The Brocade device can also can redistribute Level-1 IPv4 IS-IS routes into Level-2 IPv4 IS-IS routes, and Level-2 IPv4 IS-IS routes into Level-1 IPv4 IS-IS routes.

Route redistribution from other sources into IPv4 IS-IS is disabled by default. When you enable redistribution, the device redistributes routes only into Level 2 by default. You can specify Level 1 only, Level 2 only, or Level 1 and Level 2 when you enable redistribution.

The device automatically redistributes Level-1 routes into Level-2 routes. Thus, you do not need to enable this type of redistribution. You also can enable redistribution of Level-2 routes into Level-1 routes.

The device attempts to use the redistributed route's metric as the route's IPv4 IS-IS metric. For example, if an OSPF route has an OSPF cost of 20, the device uses 20 as the route's IPv4 IS-IS metric. The device uses the redistributed route's metric as the IPv4 IS-IS metric unless the route does not a have a valid metric. In this case, the device assigns the default metric value to the route. For information about the default metric, refer to the Changing the default redistribution metric on page 279 section, which follows this section.

Changing the default redistribution metric

When IPv4 IS-IS redistributes a route from another route source (such as OSPF, BGP4+, or a static IPv4 route) into IPv4 IS-IS, it uses the route's metric value as its metric when the metric is not modified by a route map or metric parameter and the default redistribution metric is set to its default value of 0. You can change the default metric to a value from 0 - 65535.

NOTE
The implementation of IS-IS does not support the optional metric types Delay, Expense, or Error.

For example, to change the default metric to 20, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv4u)# default-metric 20
```

Syntax: [no] default-metric *value*

The *value* parameter specifies the default metric. You can specify a value from 0 - 65535. The default is 0.

To restore the default value for the default metric, enter the **no** form of this command.

Globally change the default redistribution metric

You can change the metric value for a specific interface by using the **isis metric** command or **isis ipv6 metric** command. This feature allows you to change the metric value globally for all the active ISIS interfaces using one command.

You can still configure the interface level metric. If ISIS metric is configured on the interface, it will take the precedence over the global configuration.

Configuration steps

1. Configure router ISIS using the router isis command.
2. Go to the appropriate address-family using address-family [ipv4/ipv6] unicast command.
3. Configure default metric using default-link-metric <value> command.

Configuration example

The following global configuration example ISIS default metric is for the IPv4 address-family. It can be similarly configured for IPv6 address-family.

```
device(config)#router isis
device(config-isis-router)#address-family ipv4 unicast
device(config-isis-router-ipv4u)# default-link-metric  40
```

Syntax: [no] default-link-metric *value* [**level-1** | **level-2**]

The *value* parameter is the default-link-metric value to be set for the given address-family. This is a required parameter for this command. There is no default value for this parameter. For metric-style narrow: 1 to 63. For metric-style wide: 1 to 16777215.

The *level* parameter is an optional parameter used to set the default-metric for only one of the levels. If this parameter is not given, the default-link-metric will be applied to both level-1 and level-2.

The **no** version of command will revert the metric value to default, which is 10.

Metric behavior with change in metric-style

There are two types of metric styles in ISIS, narrow metric and wide metric. The range of the metric value is different in both of these styles. If there is a change in the metric-style configuration, the default-link-metric will also change with it. The new value of the default-link-metric will be equal to the minimum of a) configured value and b) the maximum value supported for the new metric-style.

If the metric style changes from narrow metric to wide metric, there will be no change in the value of default-link-metric.

If the metric style changes from wide metric to narrow metric, and if the value of default-link-metric is greater than 63, the default-link-metric will now take the value 63, as it is the maximum supported in the narrow metric.

ISIS Show command

The show isis command and show ipv6 isis command output has been modified to reflect the default-link-metric configured.

```
device#sh isis
.....
 Default redistribution metric: 0
 Default link metric for level-1: 33
 Default link metric for level-2: 5
 Protocol Routes redistributed into IS-IS:
.....
device#
```

Redistributing static IPv4 routes into IPv4 IS-IS

To redistribute static IPv4 routes from the IPv4 static route table into IPv4 IS-IS routes, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv4u)# redistribute static
```

This command configures the Brocade device to redistribute all static IPv4 routes into Level-2 IS-IS routes.

Syntax: [no] redistribute static [**level-1** | **level-1-2** | **level-2**] | **metric** *num* | **metric-type** [**external** | **internal**] | **route-map** *name*

The **level-1**, **level-1-2**, and **level-2** keywords restrict redistribution to the specified IPv4 IS-IS level.

The **metric num** parameter changes the metric. You can specify a value from 0 - 4294967295.

The **metric-type external** | **internal** parameter restricts redistribution to one of the following:

- **external** - The metric value is not comparable to an IPv4 IS-IS internal metric and is always higher than the IPv4 IS-IS internal metric.
- **internal** - The metric value is comparable to metric values used by IPv4 IS-IS. This is the default.

The **route-map***name* parameter restricts redistribution to those routes that match the specified route map. The route map must already be configured before you use the route map name with the **redistribute** command. For example, to configure a route map that redistributes only the static IPv4 routes to the destination networks 192.168.0.0/24, enter commands such as the following:

```
device(config)# access-list 10 permit 192.168.0.0 0.0.255.255
device(config)# route-map static permit 1
device(config-routemap static)# match ip address 10
device(config-routemap static)# router isis
device(config-isis-router)# address-family ipv4 unicast
device(config-isis-router-ipv4u)# redistribute static route-map static
```

Redistributing directly connected routes into IPv4 IS-IS

To redistribute directly connected IPv4 routes into IPv4 IS-IS routes, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv4u)# redistribute connected
```

This command configures the Brocade device to redistribute all directly connected routes in the IPv4 route table into Level-2 IPv4 IS-IS.

Syntax: [no] redistribute connected [**level-1** | **level-1-2** | **level-2**] | **metric** *number* | **metric-type** [**external** | **internal**] | **route-map** *name*]

The parameters are the same as the parameters for the **redistribute static** command.

Redistributing RIP routes into IPv4 IS-IS

To redistribute RIP routes into IPv4 IS-IS, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv4u)# redistribute rip
```

This command configures the Brocade device to redistribute all RIP routes into Level-2 IS-IS.

Syntax: [no] redistribute rip [**level-1** | **level-1-2** | **level-2**] | **metric number** | **metric-type** [**external** | **internal**] | **route-map** *name*

The parameters are the same as the parameters for the **redistribute static** command.

Redistributing OSPF routes into IPv4 IS-IS

To redistribute OSPF routes into IPv4 IS-IS, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv4u)# redistribute ospf
```

This command configures the Brocade device to redistribute all OSPF routes into Level-2 IPv4 IS-IS.

Syntax: [no] redistribute ospf [**level-1** | **level-1-2** | **level-2**] | **match** [**external1** | **external2** | **internal**] | **metric number** | **metric-type** [**external** | **internal**] | **route-map** *name*

Most of the parameters are the same as the parameters for the **redistribute static** command. However, the **redistribute ospf** command also has the **match external1** | **external2** | **internal** parameter. This parameter specifies the OSPF route type you want to redistribute into IPv4 IS-IS. By default, the **redistribute ospf** command redistributes only internal routes.

- **external1** - An OSPF type 1 external route.
- **external2** - An OSPF type 2 external route.
- **internal** - An internal route calculated by OSPF.

Redistributing BGP4+ routes into IPv4 IS-IS

To redistribute BGP4+ routes into IPv4 IS-IS, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv4u)# redistribute bgp
```

This command configures the device to redistribute all its BGP4 routes into Level-2 IPv4 IS-IS.

Syntax: [no] redistribute bgp [**level-1** | **level-1-2** | **level-2**] | **metric** *number* | **metric-type** [**external** | **internal**] | **route-map** *name*

The parameters are the same as the parameters for the **redistribute static** command.

Redistributing IPv4 IS-IS routes within IPv4 IS-IS

In addition to redistributing routes from other route sources into IPv4 IS-IS, the Brocade device can redistribute Level 1 IPv4 IS-IS routes into Level 2 IPv4 IS-IS routes, and Level 2 IPv4 IS-IS routes into Level 1 IPv4 IS-IS routes. By default, the device redistributes routes from Level 1 into Level 2.

NOTE
The Brocade device automatically redistributes Level 1 routes into Level 2 routes, even if you do not enable redistribution.

For example, to redistribute all IPv4 IS-IS routes from Level 2 into Level 1, enter the following command at the IPv4 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv4u)# redistribute isis level-2 into level-1
```

The device automatically redistributes Level-1 routes into Level 2.

Syntax: [no] redistribute isis level-1 into level-2 | **level-2 into level-1** [**prefix-list** *name*]

The **level-1 into level-2** | **level-2 into level-1** parameter specifies the direction of the redistribution:

- **level-1 into level-2** - Redistributes Level 1 routes into Level 2. This is the default.
- **level-2 into level-1** - Redistributes Level 2 routes into Level 1.

The **prefix-list***name* specifies an IP prefix list.

Configuring IS-IS point-to-point over Ethernet

IS-IS uses its neighbor's MAC address to form an adjacency and stores the neighbors MAC address to recognize the adjacency in the future. This is no problem with directly adjacent routers but can become a problem when adjacency is required between routers that are more than one hop away. To accommodate an IS-IS network with this type of configuration, the IS-IS Point-to-Point over Ethernet feature has been developed.

Using the IS-IS Point-to-Point feature over ethernet, routers that are several hops away or available through an IP GRE tunnel (as described in Configuring IS-IS over a GRE IP tunnel on page 285) can form an IS-IS adjacency. It can be used when only two IS's are part of the broadcast network. This feature is configured at the interface level of the routers that are forming an adjacency. For example, Figure 20 shows two Brocade devices several hops away from each other that are configured for IS-IS adjacency.

FIGURE 20 IS-IS Point-to-Point configuration

You can use the commands in the following configurations to enable the IS-IS Point-to-Point feature:

Brocade IS-IS Router A configuration

To configure Brocade IS-IS Router A for the IS-IS Point-to-Point feature use the following commands.

```
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# ip router isis
device(config-if-e10000-1/1)# ip address 10.10.2.2
device(config-if-e10000-1/1)# isis point-to-point
```

Brocade IS-IS Router B configuration

To configure Brocade IS-IS Router B for the IS-IS Point-to-Point feature use the following commands.

```
device(config)# interface ethernet 2/1
device(config-if-e10000-2/1)# ip router isis
device(config-if-e10000-2/1)# ip address 10.10.1.2
device(config-if-e10000-2/1)# isis point-to-point
```

Syntax: [no] isis point-to-point

Displaying IS-IS point-to-point configuration

Use the **show isis interface** command to determine if IS-IS point-to-point is configured on an interface. In the example below, the lines in bold identify IS-IS point-to-point configuration.

```
device# show isis interface
Total number of IS-IS Interfaces: 2
Interface : v128 Local Circuit Number: 0000000c
   Circuit Type : PTP Circuit Mode : LEVEL-1-2
   Circuit State: UP Passive State: FALSE
   MTU : 1497
   Level-1 Metric: 10, Level-1 Priority: 64
   Circuit State Changes: 1 Circuit Adjacencies State Changes: 1
   Rejected Adjacencies: 0
   Circuit Authentication Fails: 0 Bad LSP 0
   Control Messages Sent: 45600 Control Messages Received: 6778
   IP Enabled: TRUE
   IP Address and Subnet Mask:
    10.1.1.1 255.255.255.0
   IPv6 Enabled: FALSE
```

To determine if IS-IS point-to-point link is being used by ISs, use the **show isis neighbor** command.

```
device# show isis neighbor
System Id   Interface  SNPA           State Holdtime Type Pri StateChgeTime
SFO-RX16    eth1/1     0000.00db.0eee UP    10       ISL2 64  0 :5 :5 :12
SFO-RX16    eth1/1     0000.00db.0eee UP    10       ISL1 64 0 :5 :5 :12
SFO-RX16    ve 128     0000.0000.0005 UP    30       PTPT 127 0 :4 :46:59
```

Configuring IS-IS over a GRE IP tunnel

As described in Configuring IS-IS point-to-point over Ethernet on page 283, IS-IS adjacency can be established over ethernet between routers that are more than one hop away using the IS-IS Point-to-Point feature. IS-IS over a GRE IP tunnel extends this capability by allowing you to configure IS-IS adjacency between routers on either end of a GRE IP tunnel. To configure IS-IS over a GRE IP Tunnel you must configure the following:

- Configure the routers that you want to establish adjacency for IS-IS point-to-point as described in Configuring IS-IS point-to-point over Ethernet on page 283.
- Configure a GRE IP Tunnel.
- Configure the routers used for the GRE IP Tunnel for IS-IS using the**router isis** command.
- Configure the tunnel interfaces on the routers used for the GRE IP Tunnel for IS-IS point-to-point using the **isis point-to-point** command.

Configuration considerations

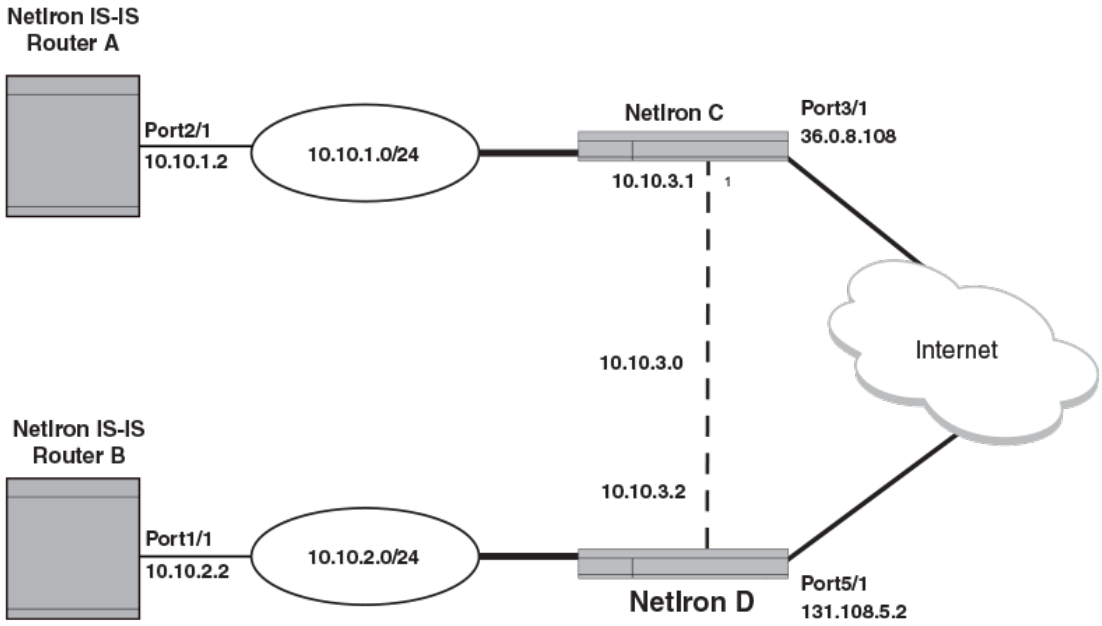
The configuration considerations are as follows:

- When a GRE tunnel is configured, you cannot configure the same routing protocol on the tunnel through which the device learns the route to the tunnel destination. For example, if a device learns the tunnel destination route through the OSPF protocol, you cannot configure the OSPF protocol on the same Tunnel and vice-versa. When a tunnel has OSPF configured, the device cannot learn the tunnel destination route through OSPF. This will cause the system to become unstable.
- When you have keepalive configured on both sides of a GRE tunnel, we recommend that you disable the tunnel before changing any tunnel configurations. You can then re-enable the tunnel to restore it to normal functionality.
- When configuring a GRE IP Tunnel, the device must be configured with one of the following CAM Profiles: ipv4, ipv6, mpls-l3vpn, ipv4-vpn, multi-service-2 or mpls-l3vpn-2.

Configuring IS-IS over a GRE IP tunnel

Figure 21 displays a network configured for IS-IS over a GRE IP tunnel. In the example, Brocade IS-IS Router A and Brocade IS-IS Router B are configured for adjacency. Routers Brocade C and Brocade D are configured with a GRE IP tunnel. Following the illustration are examples of the configurations required for IS-IS over a GRE IP tunnel.

FIGURE 21 IS-IS over a GRE IP tunnel



The following examples describe the configurations that support IS-IS over a GRE IP tunnel for each of the routers in Figure 21 .

Brocade IS-IS Router A configuration

To configure Brocade IS-IS Router A for the IS-IS Point-to-Point feature use the following commands.

```
device(config)# interface ethernet 2/1
device(config-if-e10000-2/1)# ip router isis
device(config-if-e10000-2/1)# ip address 10.10.1.2
device(config-if-e10000-2/1)# isis point-to-point
```

Brocade IS-IS Router B configuration

To configure Brocade IS-IS Router B for the IS-IS Point-to-Point feature use the following commands.

```
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# ip router isis
device(config-if-e10000-1/1)# ip address 10.10.2.2
device(config-if-e10000-1/1)# isis point-to-point
```

Brocade C configuration

To configure the Brocade C router for the IS-IS over a GRE IP tunnel feature, use the following commands.

```
device(config)# router isis
device(config-isis-router)# exit
device(config)# interface tunnel 1
device(config-tnif-1)tunnel source 10.0.8.108
device(config-tnif-1)tunnel destination 10.108.5.2
device(config-tnif-1)tunnel mode gre ip
device(config-tnif-1)isis point-to-point
device(config-tnif-1)ip address 10.10.3.1/24
device(config-tnif-1)exit
device(config) ip route 10.10.2.0/24 10.10.3.1
```

Brocade D configuration

To configure the Brocade D router for the S-IS over a GRE IP tunnel feature, use the following commands.

```
device(config)# router isis
device(config-isis-router)# exit
device(config)# interface tunnel 1
device(config-tnif-1) tunnel source ethernet 5/1
device(config-tnif-1) tunnel destination 10.0.8.108
device(config-tnif-1) tunnel mode gre ip
device(config-tnif-1) isis point-to-point
device(config-tnif-1) ip address 10.10.3.2/24
device(config-tnif-1) exit
device(config) ip route 10.10.1.0/24 10.10.3.1
```

Displaying IS-IS over GRE IP tunnel

You can use the **show isis interface** command to determine if IS-IS point-to-point is configured on a tunnel interface. In the example below, the lines in bold identify IS-IS point-to-point configuration in the gre_tnl 1 interface.

```
device# show isis interface
Total number of IS-IS Interfaces: 2
Interface : gre_tnl 1
   Circuit State: UP Circuit Mode: LEVEL-1-2
   Circuit Type : PTP Passive State: FALSE
   Circuit Number: 0x02, MTU: 1497
   Level-1 Metric: 10, Level-1 Priority: 64
   Level-1 Auth-mode: None
   Level-2 Auth-mode: None
   Level-1 Metric: 10, Level-1 Priority: 50
   Level-1 Hello Interval: 10 Level-1 Hello Multiplier: 3
   Level-1 Designated IS: XMR1-02 Level-1 DIS Changes: 0
   Level-2 Metric: 10, Level-2 Priority: 50
   Level-2 Hello Interval: 10 Level-2 Hello Multiplier: 3
   Level-2 Designated IS: MLX2-02 Level-2 DIS Changes: 0
   Circuit State Changes: 1 Circuit Adjacencies State Changes: 1
   Rejected Adjacencies: 0
   Circuit Authentication L1 failures: 0
   Circuit Authentication L2 failures: 0
   Bad LSPs 0
   Control Messages Sent: 318 Control Messages Received: 229
   IP Enabled: TRUE
   IP Address and Subnet Mask:
10.50.50.20         255.255.255.0
   IPv6 Enabled: FALSE
```

To determine if IS-IS point-to-point link is being used by ISs, use the **show isis neighbor** command. In the example below, the line in bold identifies a point-to-point configuration on the XMR1 system for the gre_tnl 1 interface.

```
device# show isis neighbor
Total number of IS-IS Neighbors: 3
System Id      Interface  SNPA            State Holdtime Type  Pri    StateChgeTime
0000.0000.0004 eth 6/2    0000.0076.4805  UP    30       ISL1  0   0  :0 :8 :42
0000.0000.0004 eth 6/2    0000.0076.4805  UP    30       ISL2  0   0  :0 :8 :42
XMR1           gre_tnl 1  0000.0000.0005  UP    30       PTPT  127 0  :0 :9 :16
```

You can use the **show ip route isis** command to determine if next hop is a tunnel. For example.

```
device# show ip route isis
Type Codes - B:BGP D: Connected I: ISIS S: Static R: RIP O:O SPF; Cost - Dist/Metric
        Destination        Gateway         Port         Cost       Type
1       10.30.30.0/24      10.50.50.10     gre_tnl 1    115/20     IL1
2       10.100.100.0/24    10.50.50.10     gre_tnl 1    115/20     IL1
3       10.100.101.0/24    10.50.50.10     gre_tnl 1    115/20     IL1
4       10.100.102.0/24    10.50.50.10     gre_tnl 1    115/20     IL1
5       10.100.103.0/24    10.50.50.10     gre_tnl 1    115/20     IL1
6       10.100.104.0/24    10.50.50.10     gre_tnl 1    115/20     IL1
7       10.100.105.0/24    10.50.50.10     gre_tnl 1    115/20     IL1
8       10.100.106.0/24    10.50.50.10     gre_tnl 1    115/20     IL1
9       10.100.107.0/24    10.50.50.10     gre_tnl 1    115/20     IL1
```

IS-IS Non-Stop Routing

Overview

NOTE
IS-IS Non-Stop Routing (NSR) is applicable only to IPv4 routes computed by IS-IS and does not apply to IPv6 routes.

NOTE
IS-IS NSR is not supported on the Brocade NetIron CES Series and Brocade NetIron CER Series platforms.

IS-IS Non-Stop Routing (NSR) enables the IS-IS router to maintain topology and data flow to avoid re-convergence in the network during a processor switchover or hitless-reload event. The IS-IS Bidirectional Forwarding Detection (BFD) sessions survive the switchover and hitless-reload conditions. In general, a router restart causes its peer to remove the routes originated from the router and reinstalls them. This IS-IS NSR feature enables the router to maintain neighbors and LSA database with its peer on the event of a router restart.

In IS-IS NSR, the processor switchovers and the hitless-reloads are treated the same as they are during startup and the overload bit is set in the same way as it is after a reboot. For more information on overload bit setup, refer to Setting the overload bit on page 264.

NOTE
IS-IS NSR is independent of Graceful Restart (GR) and GR help role mechanisms.

Limitations

- The IS-IS over GRE tunnel feature does not support IS-IS NSR. The GRE tunnel interface types are not supported.
- The IS-IS shortcuts are not supported because they depend on the MPLS tunnel.
- If the IS-IS hellos are forwarded at Layer 2 and the device executes a hitless-reload, hellos will not be forwarded for a brief time. The IS-IS adjacencies are lost for 12 seconds and there will be data traffic loss.
- The configuration events that occur close to switchover or hitless-reload may get lost due to CLI synchronization issues.
- The neighbor or interface state changes close to switchover or hitless-reload cannot be handled.
- The IS-IS neighbor hold timer is restarted upon IS-IS NSR switchover or hitless-reload.
- The traffic counters are not synchronized because the neighbor and LSP database counters are recalculated on the standby module during synchronization.
- With IS-IS NSR enabled, after switchover or hitless-reload to standby MP, IS-IS routes, LSP database and neighbor adjacencies are maintained so that there will be no loss of existing traffic to the IS-IS destinations.
- The IS-IS NSR hitless failover event may not be completely invisible to the network because, after switchover, additional flooding of CSNP packets will occur in the directly connected neighbors.

Enabling and disabling IS-IS NSR

To globally enable IS-IS NSR, enter the following commands.

```
device(config)# router isis
device(config-isis-router)# nonstop-routing
```

To globally disable IS-IS NSR, enter the following commands.

```
device(config)# router isis
device(config-isis-router)# no nonstop-routing
```

Syntax: [no] nonstop-routing

Disabling and enabling IS-IS graceful restart helper mode

Graceful Restart allows for the forwarding of data packets to continue along known routes while the routing protocol information is being restored following a processor switchover.

NOTE
The ISIS GR helper mode is enabled by default on the the router and there is no configuration required.

To disable ISIS graceful restart (GR) helper mode, enter the following commands.

```
device(config)# router isis
device(config-isis-router)# graceful-restart helper-disable
```

To enable the disabled ISIS graceful restart (GR) helper mode, enter the following commands.

```
device(config)# router isis
device(config-isis-router)# no graceful-restart helper-disable
```

[no] graceful-restart helper-disable

Displaying the IS-IS NSR status

To display the IS-IS NSR status, enter the following command.

```
device(config-isis-router)# show isis
IS-IS Routing Protocol Operation State: Enabled
 IS-Type: Level-2
 System ID: cccc.bbbb.aaaa
 Manual area address(es):
   22.6666
 Level-1-2 Database State: On
 Administrative Distance: 210
 Maximum Paths: 4
 Default redistribution metric: 0
 Protocol Routes redistributed into IS-IS:
    None
 Number of Routes redistributed into IS-IS: 0
 Level-1 Auth-mode: None
 Level-2 Auth-mode: None
 Metric Style Supported for Level-1: Narrow
 Metric Style Supported for Level-2: Narrow
Graceful-Restart Helper support enabled
 IS-IS Partial SPF Optimizations: Enabled
 Timers:
    L1 SPF: Max-wait 5s Init-wait 5000ms Second-wait 5000ms
    L2 SPF: Max-wait 5s Init-wait 5000ms Second-wait 5000ms
    L1 SPF will run in 800msec
    L2 SPF is not scheduled
    PSPF: Max-wait 5000ms Init-wait 2000ms Second-wait 5000ms
    PSPF will run in 300msec
    LSP: max-lifetime 45s, refresh-interval 7s, gen-interval 10s
         retransmit-interval 5s, lsp-interval 33ms
    SNP: csnp-interval 10s, psnp-interval 2s
 Global Hello Padding : Enabled
 Global Hello Padding For Point to Point Circuits: Enabled
 Ptpt Three Way HandShake Mechanism: Enabled
 BGP Ipv4 Converged: FALSE, Ipv6 Converged: FALSE
 IS-IS Traffic Engineering Support: Disabled
```

```
No ISIS Shortcuts Configured
BFD: Disabled
NSR: Enabled
   NSR State: Normal
   Standby MP: Ready
   Sync State: Enabled
Interfaces with IPv4 IS-IS configured:
   ethernet 2/1 ve 20 ve 165 loopback 1 loopback 2 loopback 3
```

The following table describes the output of the **show isis** command.

TABLE 37 Output from the **show isis** command

This field...	Displays...
IS-IS Routing Protocol Operation State	This field indicates the operating state of IS-IS and the possible states includes the following: • Enabled - IS-IS is enabled. • Disabled - IS-IS is disabled.
IS-Type	This field indicates the intermediate system type and the possible types includes the following: • Level 1 only - The Brocade device routes traffic only within the area in which it resides. • Level 2 only - The Brocade device routes traffic between areas of a routing domain. • Level 1-2 - The Brocade device routes traffic within the area in which it resides and between areas of a routing domain.
System ID	This field indicates the unique IS-IS router ID. Typically, the router base MAC address is used as the system ID.
Manual area address(es)	This field indicates the Area address(es) of the Brocade device.
Level-1-2 Database State	This field indicates the state of the Level 1-2 Database: • On • Off
Administrative Distance	This field specifies the current setting of the IS-IS administrative distance.
Maximum Paths	This field specifies the number of paths IS-IS can calculate and install in the forwarding table.
Default redistribution metric	This field specifies the value of the default redistribution metric, which is the IS-IS cost of redistributing the route into IS-IS.
Number of Routes redistributed into IS-IS	This field specifies the number of routes distributed into IS-IS.
Level-1 Auth-mode	This field indicates one of the following authentication modes set for Level-1 on the router: • None • md5 • cleartext

TABLE 37 Output from the **show isis** command (Continued)

This field...	Displays...
Level-2 Auth-mode	This field indicates one of the following authentication modes set for Level-2 on the router: • None • md5 • cleartext
Metric Style Supported for Level-1	This field indicates the metric style supported for Level-1 and the following values are supported: • Wide - Wide Metric Style • Narrow - Narrow Metric Style
Metric Style Supported for Level-2	This field indicates the metric style supported for Level-2 and the following values are supported: • Wide - Wide Metric Style • Narrow - Narrow Metric Style
IS-IS Graceful restart helper mode	This field indicates the IS-IS GR helper mode function and the parameter can contain one of the following values: • Enabled • Disabled
IS-IS Partial SPF Optimizations	This field indicates the IS-IS partial SPEF optiomization and the parameter can contain one of the following values: • Enabled • Disabled
Timers: L1 or L2 SPF:	The following values are displayed individually for IS-IS levels 1 and 2.
max-wait	This fields indicates the maximum time gap that will occur between running of SPF calculations. It is the value configured as the spf-max-wait variable in the**spf-interval** command as described in Configuring the IS-IS PSPF exponential back-off feature on page 272.
Init-wait	This fields indicates the initial time gap between an SPF event and the first running of SPF. This value reflects the spf-initial-time variable that is configured using the **spf-interval** command as described in Configuring the IS-IS PSPF exponential back-off feature on page 272.
Second-wait	This fields indicates the interval between the first running of SPF and the first recalculation of the SPF tree. If this optional value is configured, it will be doubled with each recalculation of the SPF tree until the value is equal to the max-wait value This value reflects the spf-second-wait variable that is configured using the **spf-interval** command as described in Configuring the IS-IS PSPF exponential back-off feature on page 272.
SPF run status.	This field is not specifically labeled but it is displayed directly under the SPF timers. It can be any of the three values shown below: • SPF is running • SPF will run in *sec* where the *sec* variable is a value in seconds until the next time that SPF will be run. • SPF is not scheduled

TABLE 37 Output from the **show isis** command (Continued)

This field...	Displays...
Timers: PSPF:	
max-wait	This fields indicates the maximum time gap that will occur between running of PSPF calculations. It is the value configured as the max-wait value in the**partial-spf-interval** command as described in Configuring the IS-IS PSPF exponential back-off feature on page 272.
Init-wait	This fields indicates the initial time gap between the wait time after an LSP change until the first PSPF calculation. This value reflects the initial-wait variable that is configured using the **partial-spf-interval** command as described in Configuring the IS-IS PSPF exponential back-off feature on page 272.
Second-wait	This fields indicates the wait time between the first and second PSPF calculations. If this optional value is configured, it will be doubled with each PSPF recalculation until the value is equal to the max-wait value This value reflects the second-wait variable that is configured using the **partial-spf-interval** command as described in Configuring the IS-IS PSPF exponential back-off feature on page 272.
PSPF run status.	This field is not specifically labeled but it is displayed directly under the PSPF timers. It can be any of the three values shown below: • PSPF is running • PSPF will run in *sec* where the *sec* variable is a value in seconds until the next time that PSPF will be run. • PSPF is not scheduled
Timers: LSP:	
max-lifetime	This fields indicates the maximum number of seconds an unrefreshed LSP can remain in the Brocade device's LSP database. The default value is 1000 sec.
refresh-interval	This fields indicates the maximum number of seconds that a Brocade device waits between sending updated LSPs to its IS-IS neighbors. The default value is 1 sec.
gen-interval	This fields indicates the minimum number of seconds that a Brocade device waits between sending updated LSPs to its IS-IS neighbors. The default value is 10 sec.
retransmit-interval	This fields indicates the amount of time the Brocade device waits before it retransmits LSPs. The default value is 5 sec.
lsp-interval	This fields indicates the rate of transmission (in milliseconds) of the LSPs. The default rate is 33 ms.
Timers: SNP:	

TABLE 37 Output from the **show isis** command (Continued)

This field...	Displays...
csnp-interval	This fields indicates how often the designated IS sends a CSNP to the broadcast interface. The default value is 10 sec.
psnp-interval	This fields indicates how often the IS sends a PSNP. The default value is 2 sec.
Global Hello Padding	The value can be: • Enabled • Disabled
Global Hello Padding For Point to Point Circuits	The value can be: • Enabled • Disabled
Ptpt Three Way HandShake Mechanism	The value can be: • Enabled • Disabled
IS-IS Traffic Engineering Support	The value can be: • Enabled • Disabled
BFD	The value can be: • Enabled • Disabled
Interfaces with IPv4 IS-IS configured	This field specifies the interfaces on which IPv4 IS-IS is configured.
NSR state	This field indicates the state of the IS-IS NSR and takes the following values: • Normal - This indicates that the switchover is either compete or the switchover event is not triggerd. • SwitchOver Detected - This indicates that the switchover event is recognized by the IS-IS. • All Card Done - This is an internal event after which the IS-IS starts sending hellos to its neighbors and schedules SPF. • SPF Run Complete - This indicates that the SPF run and updating of the IS-IS routes to RTM is complete. • Wait for BGP - This event indicates that the IS-IS is waiting for redistribution to complete. After redistribution to IS-IS is complete, the IS-IS NSR state will change to Normal.
Standby MP	This field indicates the standby MP is active, ready, or inactive: • Active - This indicates the standby MP is active. • Inactive - This indicates the standby MP is either down or not present. • Ready - This indicates the standby MP is ready to accept configuration updates or database updates.

TABLE 37 Output from the **show isis** command (Continued)

This field...	Displays...
Sync State	This field indicates whether the synchronization state is enabled or disabled. The state changes depending on whether or not the **Non Stop-Routing** command is configured under the router IS-IS.
Interfaces with IPv4 IS-IS configured	This field specifies the interfaces configured with IPv4 IS-IS.

Configuring ISIS properties on an interface

This section describe the IS-IS parameters for an interface.

Disabling and enabling IS-IS on an interface

In addition to enabling IS-IS globally, you also must enable the protocol on the individual interfaces connected to ISs or ESs. To enable IS-IS locally on specific interfaces, enter commands such as the following.

```
device(config)# interface ethernet 1/1
device(config-if-1/1)# ip router isis
device(config-if-1/1)# exit
device(config)# interface pos 2/3
device(config-if-2/3)# ip router isis
```

These commands enable IS-IS on ports 1/1 and 2/3. The NET configured above (at the IS-IS configuration level) applies to both interfaces.

Syntax: [no] ip router isis

Disabling or re-enabling formation of adjacencies

When you enable IS-IS on any type of interface except a loopback interface, the interface also is enabled to send advertisements and form an adjacency with an IS at the other end of the link by default. Adjacency formation and advertisements are disabled by default on loopback interfaces.

You can enable or disable adjacency formation and advertisements on an interface.

NOTE
The Brocade device advertises an IS-IS interface to its area regardless of whether adjacency formation is enabled.

To disable IS-IS adjacency formation on an interface, enter commands such as the following.

```
device(config)# interface ethernet 2/8
device(config-if-e1000-2/8)# isis passive
```

This command disables IS-IS adjacency formation on port 2/8. The device still advertises this IS-IS interface into the area, but does not allow the port to form an adjacency with the IS at the other end of the link.

Syntax: [no] isis passive

Setting the priority for designated IS election

The priority of an IS-IS interface determines the priority of the interface for being elected as a Designated IS. Level-1 has a Designated IS and Level-2 has a Designated IS. The Level-1 and Level-2 Designated ISs are independent, although the same device can become both the Level-1 Designated IS and the Level-2 Designated IS.

By default, the Level-1 and Level-2 priority is 64. You can configure an interface's priority to a value from 1 - 127. You can configure the same priority for both Level-1 and Level-2 or you can configure a different priority for each level. In case of a tie (if two or more devices have the highest priority within a given level), the device with the highest MAC address becomes the Designated IS for that level.

NOTE
You can set the IS-IS priority on an individual interface basis only. You cannot set the priority globally.

To set the IS-IS priority on an interface, enter commands such as the following.

```
device(config)# interface ethernet 2/8
device(config-if-e1000-2/8)# isis priority 127
```

This command sets the IS-IS priority on port 1/1 to 127. Since the command does not specify Level-1 or Level-2, the new priority setting applies to both IS-IS levels.

Syntax: [no] isis priority num [**level-1** | **level-2**]

The *num* parameter specifies the priority and can be from 1 - 127. A higher numeric value means a higher priority. The default is 64.

The **level-1 | level-2** parameter applies the priority to Level-1 only or Level-2 only. By default, the priority is applied to both levels.

Limiting access to adjacencies with a neighbor

In addition to limiting access to an area (level-1) or domain (level-2), you can limit access to forming an IS-IS adjacency on a specific interface by entering a password at the interface configuration level. To enter this password, enter a command such as the following.

```
device(config)# interface ethernet 2/8
device(config-if-e1000-2/8)# isis password my-password
```

Syntax: [no] isis password string

The *string* parameter specifies the password. You can enter an alphanumeric string up to 80 characters long. The password can contain blank spaces. If you use a blank space in the password, you must use quotation marks (" ") around the entire password; for example, **isis password "admin 2".**

Changing the IS-IS level on an interface

The section Changing the IS-IS level globally on page 269 explains how to change the IS-IS level globally. By default, a Brocade device can operate as both a Level-1 and IS-IS Level-2 router. You can change the IS-IS type on an individual interface to be Level-1 only or Level-2 only. You also can reset the type to both Level-1 and Level-2.

NOTE
If you change the IS-IS type on an individual interface, the type you specify must also be specified globally. For example, if you globally set the type to Level-2 only, you cannot set the type on an individual interface to Level-1. The software accepts the setting but the setting does not take effect.

To change the IS-IS type on a specific interface, enter commands such as the following.

```
device(config)# interface ethernet 2/8
device(config-if-e1000-2/8)# isis circuit-type level-1
```

Syntax: [no] isis circuit-type level-1 | **level-1-2** | **level-2**

The **level-1 | level-1-2 | level-2** parameter specifies the IS-IS type. If you want to re-enable support for both IS-IS types, re-enter the command you entered to change the IS-IS type, and use "**no**" in front of the command.

Disabling and enabling hello padding on an interface

The section Globally disabling or re-enabling hello padding on page 273 explains what hello padding is, why it is important and how to globally disable or enable it on a device. You can also disable hello padding on a specific interface by entering commands such as the following.

```
device(config)# interface ethernet 2/8
device(config-if-e1000-2/8)# no isis hello padding
```

Syntax: [no] isis hello padding

By default, hello padding is enabled. Enter the **no** form of the command to disable hello padding.

Changing the hello interval

The hello interval controls how often an IS-IS interface sends hello messages to its IS-IS neighbors. The default interval is 10 seconds for Level-1 and Level-2. You can change the hello interval for one or both levels to a value from 1 - 65535 seconds.

To change the hello interval for Ethernet interface 2/8, enter commands such as the following.

```
device(config)# interface ethernet 2/8
device(config-if-e1000-2/8)# isis hello-interval 20
```

This command changes the hello interval to 20 seconds. By default, the change applies to Level-1 and Level-2.

Syntax: [no] isis hello-interval num [**level-1** | **level-2**]

The *num* parameter specifies the interval, and can be from 1 - 65535 seconds. The default is 10 seconds.

The**level-1 | level-2** parameter applies the change to only the level you specify. If you do not use this parameter, the change applies to both levels.

Changing the hello multiplier

The hello multiplier is the number by which an IS-IS interface multiplies the hello interval to obtain the hold time for Level-1 and Level-2 IS-to-IS hello PDUs. The default multiplier is 3. You can set the multiplier to a value in the range 3 - 1000.

To change the hello multiplier for Ethernet interface 2/8, enter commands such as the following.

```
device(config)# interface ethernet 2/8
device(config-if-e1000-2/8)# isis hello-multiplier 50
```

This command changes the hello interval to 50. By default, the change applies to both Level-1 and Level-2.

Syntax: [no] isis hello-multiplier num [**level-1** | **level-2**]

The *num* parameter specifies the multiplier, and can be from 3 - 1000. The default is 3.

The **level-1 | level-2** parameter applies the change to only the level you specify. If you do not use this parameter, the change applies to both levels.

DIS hello interval

The DIS hello interval value is derived from the hello interval configured under the interface. The default ISIS hellos interval is 10 sec. The default DIS hello interval is 10/3 = sec. The default values of the DIS hello interval is not changed.

However, if you configure a hello interval of 20 for an interface, then the DIS hello interval for the interface becomes 20/3 =6 sec.

The DIS hello multipler is the same as the hello multipler configured under the interface.

Changing the metric added to advertised routes

When the Brocade device originates an IS-IS route or calculates a route, the Brocade device adds a metric (cost) to the route. Each IS-IS interface has a separate metric value. The default is 10.

The Brocade device applies the interface-level metric to routes originated on the interface and also when calculating routes. The Brocade device does not apply the metric to link-state information that the Brocade device receives from one IS and floods to other ISs.

The default interface metric is 10. You can change the metric on an individual interface to a value in one of the following ranges:

- 1 - 63 for the narrow metric style (the default metric style for IPv4 ISIS)
- 1 - 16777215 for the wide metric style (the default metric style for IPv4 ISIS)

NOTE
If the metric value you want to use is higher than 63 but you have not changed the metric style to wide, change the metric style first, then set the metric. The IS-IS neighbors that will receive the advertisements also must be enabled to receive wide metrics.

To change the IS-IS metric on an interface, use the following CLI method.

```
device(config)# interface ethernet 2/8
device(config-if-e1000-2/8)# isis metric 15
```

Syntax: [no] isis metric num [**level-1** | **level-2**]

The *num* parameter specifies the metric. The range of values you can specify depends on the metric style. You can specify 1 - 63 for the narrow metric style or 1 - 16777215 for the wide metric style. The default in either case is 10.

The **level-1 | level-2** parameter applies the change to only the level you specify. If you do not use this parameter, the change applies to both levels.

Displaying IPv4 IS-IS information

You can display the following information:

- General IS-IS Information - Displaying ISIS general information on page 298
- The active configuration (the IS-IS commands in the running-config) - refer to Displaying the IS-IS configuration in the running-config on page 302
- Name mappings - Displaying the name mappings on page 303
- Neighbor information - Displaying neighbor information on page 303
- Neighbor adjacency changes - Displaying IS-IS Syslog messages on page 306
- Interface information - Displaying interface information on page 306
- Route information - Displaying route information on page 311
- LSP database entries - Displaying LSP database entries on page 312
- Traffic statistics - Displaying traffic statistics on page 316
- Error statistics - Displaying error statistics on page 316
- IS-IS Log - Displaying the IS-IS SPF Log on page 318

Displaying ISIS general information

To display general IPv4 IS-IS information, enter the following command at any CLI level.

```
device#show isis
 IS-IS Routing Protocol Operation State: Enabled
IS-Type: Level-1-2
System ID: 0000.0011.1111
Manual area address(es):
47
Level-1-2 Database State: On
Administrative Distance: 115
Maximum Paths: 4
Default redistribution metric: 0
Protocol Routes redistributed into IS-IS:
Static
Number of Routes redistributed into IS-IS: 11
Level-1 Auth-mode: None
Level-2 Auth-mode: None
Metric Style Supported for Level-1: Wide
Metric Style Supported for Level-2: Wide
IS-IS Partial SPF Optimizations: Enabled
Timers:
L1 SPF: Max-wait 120s Init-wait 100ms Second-wait 120000ms
L2 SPF: Max-wait 100s Init-wait 100ms Second-wait 100000ms
L1 SPF is not scheduled
L2 SPF is not scheduled
PSPF: Max-wait 120000ms Init-wait 120000ms Second-wait 120000ms
PSPF is not scheduled
 LSP: max-lifetime 1200s, refresh-interval 900s, gen-interval 10s
retransmit-interval 5s, lsp-interval 33ms
SNP: csnp-interval 10s, psnp-interval 2s
Global Hello Padding : Enabled
Global Hello Padding For Point to Point Circuits: Enabled
Ptpt Three Way HandShake Mechanism: Enabled
IS-IS Traffic Engineering Support: Disabled
BFD: Disabled
Interfaces with IPv4 IS-IS configured:
eth 1/1
```

Syntax: show isis

This display shows the following information.

TABLE 38 IS-IS neighbor information

This field...	Displays...
IS-IS Routing Protocol Operation State	The operating state of IS-IS. Possible states include the following: • Enabled - IS-IS is enabled. • Disabled - IS-IS is disabled.
IS-Type	The intermediate system type. Possible types include the following: • Level 1 only - The Brocade device routes traffic only within the area in which it resides. • Level 2 only - The Brocade device routes traffic between areas of a routing domain. • Level 1-2 - The Brocade device routes traffic within the area in which it resides and between areas of a routing domain.
System ID	The unique IS-IS router ID. Typically, the Brocade device's base MAC address is used as the system ID.
Manual area address(es)	Area address(es) of the Brocade device.
Level-1-2 Database State	The state of the Level 1-2 Database: • On • Off
Administrative Distance	The current setting of the IS-IS administrative distance.
Maximum Paths	The number of paths IS-IS can calculate and install in the forwarding table
Default redistribution metric	The value of the default redistribution metric, which is the IS-IS cost of redistributing the route into IS-IS.
Number of Routes redistributed into IS-IS	The number of routes distributed into IS-IS.
Level-1 Auth-mode	One of the following authentication modes set for Level-1 on the router: • None • md5 • cleartext
Level-2 Auth-mode	One of the following authentication modes set for Level-2 on the router: • None • md5 • cleartext
Metric Style Supported for Level-1	The following values are supported: • Wide - Wide Metric Style • Narrow - Narrow Metric Style

TABLE 38 IS-IS neighbor information (Continued)

This field...	Displays...
Metric Style Supported for Level-2	The following values are supported: • Wide - Wide Metric Style • Narrow - Narrow Metric Style
IS-IS Partial SPF Optimizations	This parameter can contain one of the following values: • Enabled • Disabled
Timers: L1 or L2 SPF:	These values are displayed individually for IS-IS levels 1 and 2.
max-wait	The maximum time gap that will occur between running of SPF calculations. It is the value configured as the spf-max-wait variable in the**spf-interval** command as described in Configuring the IS-IS PSPF exponential back-off feature on page 272.
Init-wait	The initial time gap between an SPF event and the first running of SPF. This value reflects the spf-initial-time variable that is configured using the **spf-interval** command as described in Configuring the IS-IS PSPF exponential back-off feature on page 272.
Second-wait	The interval between the first running of SPF and the first recalculation of the SPF tree. If this optional value is configured, it will be doubled with each recalculation of the SPF tree until the value is equal to the max-wait value This value reflects the spf-second-wait variable that is configured using the **spf-interval** command as described in Configuring the IS-IS PSPF exponential back-off feature on page 272.
SPF run status.	This field is not specifically labeled but is displayed directly under the SPF timers.) It can any of the three values shown below: • SPF is running • SPF will run in *sec* where the *sec* variable is a value in seconds until the next time that SPF will be run. • SPF is not scheduled
Timers: PSPF:	
max-wait	The maximum time gap that will occur between running of PSPF calculations. It is the value configured as the max-wait value in the**partial-spf-interval** command as described in Configuring the IS-IS PSPF exponential back-off feature on page 272.
Init-wait	The initial time gap between the wait time after an LSP change until the first PSPF calculation. This value reflects the initial-wait variable that is configured using the **partial-spf-interval** command as described in Configuring the IS-IS PSPF exponential back-off feature on page 272.

TABLE 38 IS-IS neighbor information (Continued)

This field...	Displays...
Second-wait	The wait time between the first and second PSPF calculations. If this optional value is configured, it will be doubled with each PSPF recalculation until the value is equal to the max-wait value This value reflects the second-wait variable that is configured using the **partial-spf-interval** command as described in Configuring the IS-IS PSPF exponential back-off feature on page 272.
PSPF run status.	This field is not specifically labeled but is displayed directly under the PSPF timers. It can any of the three values shown below: • PSPF is running • PSPF will run in *sec* where the *sec* variable is a value in seconds until the next time that PSPF will be run. • PSPF is not scheduled
Timers: LSP:	
max-lifetime	The maximum number of seconds an unrefreshed LSP can remain in the Brocade device's LSP database. The default value is 1000 sec.
refresh-interval	The maximum number of seconds that a Brocade device waits between sending updated LSPs to its IS-IS neighbors. The default value is 1 sec.
gen-interval	The minimum number of seconds that a Brocade device waits between sending updated LSPs to its IS-IS neighbors. The default value is 10 sec.
retransmit-interval	The amount of time the Brocade device waits before it retransmits LSPs. The default value is 5 sec.
lsp-interval	The rate of transmission (in milliseconds) of the LSPs. The default rate is 33 ms.
Timers: SNP:	
csnp-interval	How often the designated IS sends a CSNP to the broadcast interface. The default value is 10 sec.
psnp-interval	How often the IS sends a PSNP. The default value is 2 sec.

TABLE 38 IS-IS neighbor information (Continued)

This field...	Displays...
Global Hello Padding	This value can be: • Enabled • Disabled
Global Hello Padding For Point to Point Circuits	This value can be: • Enabled • Disabled
Ptpt Three Way HandShake Mechanism	This value can be: • Enabled • Disabled
IS-IS Traffic Engineering Support	This value can be: • Enabled • Disabled
BFD	This value can be: • Enabled • Disabled
Interfaces with IPv4 IS-IS configured	Interfaces on which IPv4 IS-IS is configured.

Displaying the IS-IS configuration in the running-config

You can display the global IS-IS configuration commands that are in effect on the Brocade device using the following CLI method.

NOTE
The running-config does not list the default values. Only commands that change a setting or add configuration information are displayed.

To list the global IS-IS configuration commands in the Brocade device's running-config, enter the following command at any level of the CLI.

```
device# show isis config
router isis
 net  20.00e0.5200.0001.00
end
```

The running-config shown in this example contains the command that enables IS-IS and a command that configures a NET.

To display the interface configuration information in the running-config, enter one of the following commands at any level of the CLI:

- **show running-config**
- **write terminal**

Syntax: show isis config

Displaying the name mappings

To display the mappings, enter the following command at any level of the CLI.

```
device# show isis hostname
Total number of entries in IS-IS Hostname Table: 1
   System ID        Hostname            * = local IS
 * 0000.00cc.dddd   XMR
```

Syntax: show isis hostname

The table in this example contains one mapping, for this Brocade device. The Brocade device's IS-IS system ID is "0000.00cc.dddd" and its hostname is "XMR". The display contains one entry for each IS that supports name mapping.

NOTE
Name mapping is enabled by default. When name mapping is enabled, the output of the **show isis database** , **show isis interface** , and **show isis neighbor** commands uses the host name instead of the system ID. To disable mapping so that these displays use the system ID instead, refer to Disabling or re-enabling display of hostname on page 269.

Displaying neighbor information

To display IS-IS neighbor information, enter the following command at any level of the CLI

```
device# show isis neighbor
Total number of IS-IS Neighbors: 2
System ID      Interface SNPA                                State Holdtime Type
Pri StateChgeTime Protocol
00e0.52b5.7800 Ether2/4 00e0.52b5.7843 UP
10                                                   ISL2     64  0  :0 :
16:8     M-ISIS
00e0.52b5.7800 Ether2/4 00e0.52b5.7843 UP
10                                                   ISL1     64  0  :0 :
16:8                 ISIS
```

Syntax: show isis neighbor [**detail**]

The **detail** option displays more details for each neighbor.

This display shows the following information.

TABLE 39 IS-IS neighbor information

This field...	Displays...
Total number of IS-IS Neighbors	The number of ISs with which the Brocade device has formed IS-IS adjacencies.
System ID	The System ID of the neighbor or the hostname of the neighbor.

TABLE 39 IS-IS neighbor information (Continued)

This field...	Displays...
Interface	The Brocade device port or virtual interface attached to the neighbor.
SNPA	The Subnetwork Point of Attachment (SNPA), which is the MAC address of the Brocade device port or virtual interface attached to the neighbor.
State	The state of the adjacency with the neighbor. The state can be one of the following: • DOWN - The adjacency is down. • INIT - The adjacency is being established and is not up yet. • UP - The adjacency is up.
Holdtime	The neighbor's advertised hold time.
Type	The IS-IS type of the adjacency. The type can be one of the following: • ISL1 - Level-1 IS • ISL2 - Level-2 IS • ES - ES **NOTE** The Brocade device forms a separate adjacency for each IS-IS type. Thus, if the Brocade device has both types of IS-IS adjacencies with the neighbor, the display contains a separate row of information for each adjacency.
Pri	The priority of this IS to be elected as the Designated IS in this broadcast network.
StateChgeTime	The amount of time that has passed since the adjacency last changed state.
Protocol	The routing protocol supported by the neighbor. The protocol can be one of the following: • MT-ISIS - Multi-Topology is enabled on the neighbor. • ISIS - Multi-Topology is not enabled on the neighbor.

To display IS-IS neighbor detail information, enter the following command at any level of the CLI.

```
device# show isis neighbor detail
Total number of IS-IS Neighbors:
1
                                          System ID Interface SNPA State Holdtime
Type Pri StateChgeTime Protocol
Core2 ve 501 0900.2b00.0005 UP    30       PTPT 127 0  :0 :46:41  M-ISIS
3-Way HandShake TLV received: circuit-id 2
Area Address(es): 00.0000
Adj Usage L1
Protocols Supported: IP IPv6
IP Address: 191.28.1.2, circuit-id 2
```

Syntax: show isis neighbor [**detail**]

The **detail** option displays more information about each neighbor.

Table 40 describes the output parameters of the **show isis neighbor detail** command.

TABLE 40 IS-IS neighbor information

This field...	Displays...
Total number of IS-IS Neighbors	The number of ISs with which the Brocade device has formed IS-IS adjacencies.
System ID	The System ID of the neighbor or the hostname of the neighbor.
Interface	The Brocade device port or virtual interface attached to the neighbor.
SNPA	The Subnetwork Point of Attachment (SNPA), which is the MAC address of the Brocade device port or virtual interface attached to the neighbor.
State	The state of the adjacency with the neighbor. The state can be one of the following: • DOWN - The adjacency is down. • INIT - The adjacency is being established and is not up yet. • UP - The adjacency is up.
Holdtime	The neighbor's advertised hold time.
Type	The IS-IS type of the adjacency. The type can be one of the following: • ISL1 - Level-1 IS • ISL2 - Level-2 IS • ES - ES **NOTE** The Brocade device forms a separate adjacency for each IS-IS type. Thus, if the Brocade device has both types of IS-IS adjacencies with the neighbor, the display contains a separate row of information for each adjacency.
Pri	The priority of this IS to be elected as the Designated IS in this broadcast network.
StateChgeTime	The amount of time that has passed since the adjacency last changed state.
3-Way Handshake TLV received	The received 3-way handshake TLV for the interface.
Area Address (es)	The address of the area.
Protocols Supported	The topology supported by the neighbor.
IP Address	The IP address assigned to the neighbor interface.
Adj Usage L1	The adjacency level used by the neighbor.
circuit ID	The ID of the IS-IS circuit running on the neighbor interface.

TABLE 40 IS-IS neighbor information (Continued)

This field...	Displays...
Protocol	The routing protocol supported by the neighbor. The protocol can be one of the following: • MT-ISIS - Multi-Topology is enabled on the neighbor. • ISIS- Multi-Topology is not enabled on the neighbor.

Displaying IS-IS Syslog messages

When logging is enabled, the Brocade device generates Syslog messages and SNMP traps for the following IS-IS events:

- Overload state (the Brocade device entering or leaving the overload state)
- Memory overrun (IS-IS is demanding more memory than is available)

You also can enable the Brocade device to generate Syslog messages and SNMP traps when an adjacency with a neighbor comes up or goes down. To enable logging of adjacency changes, refer to Logging adjacency changes on page 273.

To display Syslog entries, enter the following command at any level of the CLI.

```
device# show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 3 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning
Static Log Buffer:
Dynamic Log Buffer (50 lines):
00d00h00m42s:N:BGP Peer 10.147.202.10 UP (ESTABLISHED)
00d00h00m18s:N:ISIS L2 ADJACENCY UP 0000.0034.1234 on interface 2/8
00d00h00m08s:N:ISIS L1 ADJACENCY UP 0000.0034.1234 on interface 2/8
00d00h00m08s:N:ISIS L2 ADJACENCY UP 0000.00de.5520 on interface 5/1
00d00h00m00s:I:Warm start
```

The messages in this example indicate that the software has been reloaded (Warm start) and adjacencies between the Brocade device and three ISs have come up.

Syntax: show logging

Displaying interface information

To display information about the Brocade device's IS-IS interfaces, enter the **show isis** commands at any level of the CLI, as the examples in this section illustrate.

The following is an example of the **show isis interface** command for an Ethernet Interface module configured for a Circuit Type BCAST.

```
device(config-if-e10000-1/1)#show isis interface
Total number of IS-IS Interfaces:
1
                                        Interface: eth 1/1
Circuit State: UP Circuit Mode: LEVEL-1-2
Circuit Type: BCAST Passive State: FALSE
Circuit Number: 0x01, MTU: 1500
Level-1 Auth-mode: None
Level-2 Auth-mode: None
Level-1 Metric: 10, Level-1 Priority: 64
Level-1 Hello Interval: 5 Level-1 Hello Multiplier: 3
Level-1 Designated IS: mu2-01 Level-1 DIS Changes: 3
Level-2 Metric: 10, Level-2 Priority: 64
```

```
Level-2 Hello Interval: 5 Level-2 Hello Multiplier: 3
Level-2 Designated IS: mu2-01 Level-2 DIS Changes: 3
Next IS-IS LAN Level-1 Hello in 1 seconds
Next IS-IS LAN Level-2 Hello in 4 seconds
Number of active Level-1 adjacencies: 0
Number of active Level-2 adjacencies: 0
Circuit State Changes: 1 Circuit Adjacencies State Changes: 0
Rejected Adjacencies: 0
Circuit Authentication L1 failures: 0
Circuit Authentication L2 failures: 0
Bad LSPs: 0
Control Messages Sent: 63 Control Messages Received: 27
Hello Padding: Enabled
IP Enabled: TRUE
IP Addresses:
10.1.1.2/24
IPv6 Enabled: TRUE
IPv6 Addresses:
1000::1/32
IPv6 Link-Local Addresses:
fe80::200:ff:fe02:c000
MPLS TE Enabled: FALSE
```

The following is an example of the **show isis interface** command for a POS Interface module configured with a Circuit Type: PTP.

```
device#show isis interface
Total number of IS-IS Interfaces:
1
                                        Interface: eth
1/1
Circuit State: UP Circuit Mode: LEVEL-1-2
Circuit Type: PTP Passive State: FALSE
Circuit Number: 0x01, MTU: 1500
Level-1 Auth-mode: None
Level-1 Metric: 10
Level-1 Hello Interval: 5 Level-1 Hello Multiplier: 3
Level-2 Metric: 10
Circuit State Changes: 1 Circuit Adjacencies State Changes: 0
Rejected Adjacencies: 0
Circuit Authentication L1 failures:
0
Bad LSPs: 0
Control Messages Sent: 9 Control Messages Received: 1
Hello Padding: Enabled
IP Enabled: TRUE
IP Addresses:
10.1.1.2/24
IPv6 Enabled: TRUE
IPv6 Addresses:
1000::1/32
IPv6 Link-Local Addresses:
fe80::200:ff:fe02:c000
MPLS TE Enabled: FALSE
```

Syntax: show isis interface [**brief** | **ethernet slot-number/port-number** | **pos slot-number/port-number** | **loopback number** | **ve number**]

This display shows the following information.

TABLE 41 IS-IS interface information

This field...	Displays...
Total number of IS-IS interfaces	The number of interfaces on which IS-IS is enabled.
Interface	The port or virtual interface number to which the information listed below applies.

TABLE 41 IS-IS interface information (Continued)

This field...	Displays...
Circuit State	The state of the circuit, which can be one of the following: • DOWN • UP
Circuit Mode	The IS-IS type in use on the circuit. The mode can be one of the following: • LEVEL-1 • LEVEL-2 • LEVEL-1-2
Circuit Type	The type of IS-IS circuit running on the interface. The circuit type can be one of the following: • BCAST (broadcast). • PTP (Point-to-Point)
Passive State	The passive state determines whether the interface is allowed to form an IS-IS adjacency with the IS at the other end of the circuit. The state can be one of the following: • FALSE - The passive option is disabled. The interface can form an adjacency with the IS at the other end of the link. • TRUE - The passive option is enabled. The interface cannot form an adjacency, but can still advertise itself into the area.
Circuit Number	The ID that the instance of IS-IS running on the interface applied to the circuit between this interface and the interface at the other end of the link.
MTU	The maximum length supported for IS-IS PDUs sent on this interface.
Level-1 Auth-mode	One of the following authentication modes set for Level-1 on the router: • None • md5 • cleartext
Level-2 Auth-mode	One of the following authentication modes set for Level-2 on the router: • None • md5 • cleartext This parameter is not displayed for interfaces that are configured with a Point-to-Point circuit type. This is because separate Level-2 Hello messages are not sent on Point-to-Point interfaces. Consequently, "Hello Interval", "Hello Multiplier", and "Authentication" parameters should always be configured in the Level-1 mode for Point-to-Point interfaces.
Level-1 Metric	The default-metric value that the Brocade device inserts in IS-IS Level-1 PDUs for this interface.

TABLE 41 IS-IS interface information (Continued)

This field...	Displays...
Level-1 Priority	The priority of this IS to be elected as the Designated IS for Level-1 in this broadcast network.
Level-1 Hello Interval	The number of seconds the software waits between sending Level-1 hello PDUs to the IS at the other end of the circuit.
Level-1 Hello Multiplier	The number by which the software multiplies the hello interval to calculate the hold time set in Level-1 Hello PDUs sent on the circuit.
Level-1 Designated IS	The NET of the Level-1 Designated IS.
Level-1 DIS Changes	The number of times the NET of the Level-1 Designated IS has changed.
Level-2 Metric	The default-metric value that the Brocade device inserts in IS-IS Level-2 PDUs for this interface.
Level-2 Priority	The priority of this IS to be elected as the Designated IS for Level-2 in this broadcast network.
Level-2 Hello Interval	The number of seconds the software waits between sending Level-2 Hello messages to the IS at the other end of the circuit.
Level-2 Hello Multiplier	The number by which the software multiplies the hello interval to calculate the hold time set for Level-2 Hello PDUs sent on this circuit. This parameter is not displayed for interfaces that are configured with a Point-to-Point circuit type. This is because separate Level-2 Hello messages are not sent on Point-to-Point interfaces. Consequently, "Hello Interval", "Hello Multiplier", and "Authentication" parameters should always be configured in the Level-1 mode for Point-to-Point interfaces.
Level-2 Designated IS	The NET of the Level-2 Designated IS.
Level-2 DIS Changes	The number of times the NET of the Level-2 Designated IS has changed.
Next IS-IS LAN Level-1 Hello	Number of seconds before next Level-1 Hello PDU will be transmitted by the Brocadedevice.
Next IS-IS LAN Level-2 Hello	Number of seconds before next Level-2 Hello PDU will be transmitted by the Brocade device.
Number of active Level-1 adjacencies	The number of ISs with which this interface has an active Level-1 adjacency.
Number of active Level-2 adjacencies	The number of ISs with which this interface has an active Level-2 adjacency.
Circuit State Changes	The number of times the state of the circuit has changed.

TABLE 41 IS-IS interface information (Continued)

This field...	Displays...
Circuit State Adjacencies Changes	The number of times an adjacency has started or ended on this circuit.
Rejected Adjacencies	The number of adjacency attempts by other ISs rejected by the Brocade device.
Circuit Authentication L1 failures	The number of times the Brocade device rejected a circuit because the authentication did not match the authentication configured for Level-1on the Brocade device.
Circuit Authentication L2 failures	The number of times the Brocade device rejected a circuit because the authentication did not match the authentication configured for Level-2 on the Brocade device. This parameter is not displayed for interfaces that are configured with a Point-to-Point circuit type. This is because separate Level-2 Hello messages are not sent on Point-to-Point interfaces. Consequently, "Hello Interval", "Hello Multiplier", and "Authentication" parameters should always be configured in the Level-1 mode for Point-to-Point interfaces.
Bad LSP	The number of times the interface received a bad LSP from an IS at the other end of the circuit. The following conditions can cause an LSP to be bad: • Invalid checksum • Invalid length • Invalid lifetime value
Control Messages Sent	The number of IS-IS control PDUs sent on this interface.
Control Messages Received	The number of IS-IS control PDUs received on this interface.
Hello Padding:	The Hello Padding configuration, which can be: • Enabled • Disabled
IP Enabled	If set to TRUE, the IP protocol is enabled for this circuit.
IP Address and Subnet Mask	The IP address and subnet mask for this interface.
IPv6 Enabled	If set to TRUE, the IPv6 protocol is enabled for this circuit.
IPv6 Address and Subnet Mask	The IPv6 address and subnet mask for this interface.
Ipv6 Link-Local Addresses	The IPv6 link local address for this interface.
MPLS TE Enabled:	If set to TRUE, MPLS Traffic Engineering protocol is enabled for this circuit.
BFD Enabled:	If set to TRUE, BiDirectional Forwarding Detection is enabled for this circuit.

Displaying route information

To display the routes in the Brocade device's IS-IS route table, use either of the following methods.

To display information about the routes in the Brocade device's IS-IS route table, enter the following command at any level of the CLI.

```
device# show isis routes
Total number of IS-IS routes: 173
Destination      Mask             Cost  Type Tag      Flags
10.0.0.0          255.255.255.0    21    L2   00000000 00000242
   Path: 1       Next Hop IP: 10.1.1.1          Interface: 7/1
10.0.0.0          255.255.255.255  30    L2   00000000 00000242
   Path: 1       Next Hop IP: 10.1.1.1          Interface: 7/1
10.0.0.1          255.255.255.255  30    L2   00000000 00000242
   Path: 1       Next Hop IP: 10.1.1.1          Interface: 7/1
10.0.10.0         255.255.255.0    30    L2   00000000 00000242
   Path: 1       Next Hop IP: 10.1.1.1          Interface: 7/1
```

Syntax: show isis routes [**ip-address subnet-mask** | **ip-address/prefix**]

You may enter **ip-address***subnet-mask* or **ip-address/prefix** if you want information for a specific route.

```
device# show isis routes 10.0.111.0 255.255.255.0
10.0.111.0        255.255.255.0    21    L2   00000000 00000242
   Path: 1       Next Hop IP: 10.1.1.1          Interface: 7/1
```

This display shows the following information.

TABLE 42 IS-IS route information

This field...	Displays...
Total number of IS-IS routes	The total number of routes in the Brocade device's IS-IS route table. The total includes Level-1 and Level-2 routes.
Destination	The IP destination of the route.
Mask	The subnet mask for the destination address.
Cost	The IS-IS default metric for the route, which is the cost of using this route to reach the next-hop router to this destination.
Type	The route type, which can be one of the following: • L1 - Level-1 route • L2 - Level-2 route
Tag	The tag value associated with the route.
Path	The path number in the table. The IS-IS route table can contain multiple equal-cost paths to the same destination, in which case the paths are numbered consecutively. When IP load sharing is enabled, the Brocade device can load balance traffic to the destination across the multiple paths.
Next Hop IP	The IP address of the next-hop interface to the destination.
Interface	The Brocade device interface (port or virtual interface) attached to the next hop.

TABLE 42 IS-IS route information (Continued)

This field...	Displays...
Flags	Values used by Brocade technical support for troubleshooting.

Displaying LSP database entries

Use the following methods to display summary or detailed information about the entries in the LSP database.

NOTE
The Brocade device maintains separate LSP databases for Level-1 LSPs and Level-2 LSPs.

Displaying summary information

To display summary information for all the LSPs in the Brocade device's LSP databases, enter the following command at any level of the CLI.

```
device)# show isis database
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
XMR-1.00-00           0x0000000c   0xd048        963               1/0/0
XMR-1.01-00           0x00000004   0x09b0        957               0/0/0
XMR-1.02-00           0x00000001   0xc57b        961               0/0/0
XMR.00-00*            0x0000000b   0x23fb        1030              1/0/0
IS-IS Level-2 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
XMR-1.00-00           0x0000000d   0x7d97        964               1/0/0
XMR-1.01-00           0x00000004   0x09b0        958               0/0/0
XMR-1.02-00           0x00000001   0x200f        962               0/0/0
XMR.00-00*            0x0000000b   0x5647        1030              1/0/0
0000.0100.0003.00-00  0x0000001f   0x761a        932               0/0/0
0000.0100.0003.00-01  0x0000001d   0x9c9d        606               0/0/0
```

The command in this example shows information for the LSPS in the Brocade device's Level-1 and Level-2 LSP databases. Notice that the display groups the Level-1 and Level-2 LSPs separately.

Syntax: show isis database [**lsp-id** | **detail** | **l1** | **l2** | **level1** | **level2**]

The *lsp-id* parameter displays summary information about a particular LSP. Specify an LSPID for which you want to display information in HHHH.HHHH.HHHH.HH-HH format, for example, 3333.3333.3333.00-00. You can also enter name.HH-HH, for example, XMR.00-00.

The **detail** parameter displays detailed information about the LSPs. Refer to Displaying detailed information on page 313.

The **l1** and **level1** parameters display the Level-1 LSPs only. You can use either parameter.

The **l2** and **level2** parameters display the Level-2 LSPs only. You can use either parameter.

The **show isis database** summary display shows the following information.

TABLE 43 IS-IS summary LSP database information

This field...	Displays...
LSPID	The LSP ID, which consists of the source ID (6 bytes), the pseudonode (1 byte), and LSPID (1 byte). **NOTE** If the address has an asterisk (*) at the end, this indicates that the LSP is locally originated.
LSP Seq Num	The sequence number of the LSP.
LSP Checksum	The checksum calculated by the device that sent the LSP and used by the Brocade device to verify that the LSP was not corrupted during transmission over the network.
LSP Holdtime	The maximum number of seconds during which the LSP will remain valid. **NOTE** The IS that originates the LSP sets the timer for the LSP. As a result, LSPs do not all have the same amount of time remaining when they enter the Brocade device's LSP database.
ATT	A 4-bit value extracted from bits 4 - 7 in the Attach field of the LSP.
P	The value in the Partition option field of the LSP. The field can have one of the following values: • 0 - The IS that sent the LSP does not support partition repair. • 1 - The IS that sent the LSP supports partition repair.
OL	The value in the LSP database overload field of the LSP. The field can have one of the following values: • 0 - The overload bit is off. • 1 - The overload bit is on, indicating that the IS that sent the LSP is overloaded and should not be used as a IS-IS transit router for that level.

Displaying detailed information

To display detailed information for all the LSPs in the Brocade device's LSP databases, enter the following command at any level of the CLI.

```
device# show isis database detail
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
XMR.00-00*            0x0000000b   0x23fb        971               1/0/0
   Area Address:  49
   NLPID:  CC(IP)
   Hostname:  XMR14
   IP Address: 10.1.1.1
   IPv6 Address: 2001:db8::14
   Metric:  10    IP-Internal 10.1.1.0/24          Up-bit: 0
   Metric:  10    IS XMR.01
IS-IS Level-2 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
XMR.00-00*            0x0000000d   0x7d97        903               1/0/0
   Area Address:  49
   NLPID:  IPv6IP
```

```
Hostname:  XMR14
IP address: 10.1.1.1
IPv6 address: 2001:db8::14
Flooding to 1 interface: eth 1/7
Metric:  10    IP-Internal 10.1.1.0/24          Up-bit: 0
Metric:  10    IP-Internal 10.85.1.0/24       Up-bit: 0
Metric:  10    IS XMR.01
Metric:  10    IS XMR.02
```

TABLE 44 IS-IS detailed LSP database information

This field...	Displays...
LSPID	Refer to the description of the summary display.
LSP Seq Num	Refer to the description of the summary display.
LSP Checksum	Refer to the description of the summary display.
LSP Holdtime	Refer to the description of the summary display.
ATT or P or OL	Refer to the description of the summary display.
Area Address	The address of the area.
NLPID	The Network Layer Protocol Identifier (NLPID), which specifies the protocol the IS that sent the LSP is using. Usually, this value is "CC(IP)".
IP address	The IP address of the interface that sent the LSP. The Brocade device can use this address as the next hop in routes to the addresses listed in the rows below.
Destination addresses	The rows of information below the IP address row are the destinations advertised by the LSP. The Brocade device can reach these destinations by using the IP address listed above as the next hop. Each destination entry contains the following information: • Metric - The value of the default metric, which is the IS-IS cost of using the IP address above as the next hop to reach this destination. • Device type - The device type at the destination. The type can be one of the following: - End System - The device is an ES. - IP-Internal - The device is an ES within the current area. The IP address and subnet mask are listed. - IS - The device is another IS. The NET (NSAP address) is listed. - IP-Extended - Same as IP-Internal, except the device uses the extended TLV fields described in draft-ietf-isis-traffic-02.txt to carry the information. - IS-Extended - Same as IS, except the device uses the extended TLV fields described in draft-ietf-isis-traffic-02.txt to carry the information.
Flooding to *num* interface:	Identifies the number of interfaces on which the specific LSP entry will be flooded and identifies the interfaces.
Acking to *num* interface:	Identifies the number of interfaces on which the specific LSP entry will be acknowledged and identifies the interfaces.

Displaying database summary information

The following command is used to display the ISIS database.

```
device# show isis database summary
IS-IS Level-1 Link State Database Summary
Number of LSPs :                    2
Number of LSPs loading :            0
Number of LSP fragments :           0
Number of Pseudo LSPs :             1
Number of Pseudo LSP fragments :    0
Number of My LSPs :                 1
Number of My LSP fragments :        0
Number of My Pseudo LSPs :          0
Number of My Pseudo LSP fragments : 0
Sum of LSPs Checksum :              0x00018004
IS-IS Level-2 Link State Database Summary
Number of LSPs :                    2
Number of LSPs loading :            0
Number of LSP fragments :           0
Number of Pseudo LSPs :             1
Number of Pseudo LSP fragments :    0
Number of My LSPs :                 1
Number of My LSP fragments :        0
Number of My Pseudo LSPs :          0
Number of My Pseudo LSP fragments : 0
Sum of LSPs Checksum :              0x00019775
```

Table 45 defines the fields shown in the above example output of the **show ip ospf interface brief** command.

TABLE 45 Output of the **show isis database summary** command

This field	Displays
Number of LSPs	Total number of LSPs in database (includes those in the loading state).
Number of LSPs loading	Number of LSPs pending a full LSP update. This value is generally non-zero during adjacency formation.
Number of LSP fragments	The number of LSPs with a non-zero LSP number (a fragment of an LSP)
Number of Pseudo LSPs	The number of pseudo LSPs.
Number of Pseudo LSP fragments	The number of pseudo LSPs with a non-zero LSP number (a fragment of an LSP).
Number of My LSPs	Total number of LSPs originated by this router.
Number of My LSP fragments	The number of LSPs originated by this router with a non-zero LSP number (a fragment of an LSP)
Number of My Pseudo LSPs	The number of pseudo LSPs originated by this router.
Number of My Pseudo LSP fragments	The number of pseudo LSPs originated by this router with a non-zero LSP number (a fragment of an LSP).
Sum of LSPs Checksum	Total checksum of all LSPs in database (including those in loading state). This number should be the same across ISIS routers during periods of network stability.

Displaying traffic statistics

The Brocade device maintains statistics for common IS-IS PDU types. To display the statistics, use either of the following methods.

To display IS-IS PDU statistics, enter the following command at any level of the CLI.

```
device# show isis traffic
                          Message Received     Message Sent
 Level-1 Hellos           1029                 115
 Level-2 Hellos           1027                 112
 Level-1 LSP              6                    3
 Level-2 LSP              6                    3
 Level-1 CSNP             0                    0
 Level-2 CSNP             0                    0
 Level-1 PSNP             107                  0
 Level-2 PSNP             107                  0
```

Syntax: show isis traffic

This display shows the following information.

TABLE 46 IS-IS traffic statistics

This field...	Displays...
Level-1 Hellos	The number of Level-1 hello PDUs sent and received by the Brocade device.
Level-2 Hellos	The number of Level-2 hello PDUs sent and received by the Brocade device.
Level-1 LSP	The number of Level-1 link-state PDUs sent and received by the Brocade device.
Level-2 LSP	The number of Level-2 link-state PDUs sent and received by the Brocade device.
Level-1 CSNP	The number of Level-1 Complete Sequence Number PDUs (CSNPs) sent and received by the Brocade device.
Level-2 CSNP	The number of Level-2 CSNPs sent and received by the Brocade device.
Level-1 PSNP	The number of Level-1 Partial Sequence Number PDUs (PSNPs) sent and received by the Brocade device.
Level-2 PSNP	The number of Level-2 PSNPs sent and received by the Brocade device.

Displaying error statistics

To display IS-IS error statistics, enter the following command at any level of the CLI.

```
device# show isis counts
Area Mismatch: 0
Max Area Mismatch: 0
System ID Length Mismatch: 0
LSP Sequence Number Skipped: 0
LSP Max Sequence Number Exceeded: 0
Level-1 Database Overload: 0
Level-2 Database Overload: 0
Our LSP Purged: 0
PDU Drop Count
CSNP Auth Failures : [L1: 100] [L2: 0]
```

```
PSNP Auth Failures : [L1: 100] [L2: 0]
HELLO Auth Failures : [L1: 100] [L2: 0]
Adjacency not found : [L1: 100] [L2: 200]
Adjacency Level Mismatch : [L1: 100] [L2: 200]
IS Level Mismatch : [L1: 100] [L2: 200]
Length Too Short : [L1: 100] [L2: 200]
Length Too Large : [L1: 100] [L2: 200]
Max Area Check Failure : [L1: 100] [L2: 200]
Zero Checksum : [L1: 100] [L2: 200]
Checksum Mismatch : [L1: 100] [L2: 200]
Invalid Length : [L1: 100] [L2: 200]
```

Syntax: show isis counts

This display shows the following information.

TABLE 47 IS-IS error statistics

This field...	Displays...
Area Mismatch	The number of times the Brocade device interface was unable to create a Level-1 adjacency with a neighbor because the Brocade device interface and the neighbor did not have any areas in common.
Max Area Mismatch	The number of times the Brocade device received a PDU whose value for maximum number of area addresses did not match the Brocade device's value for maximum number of area addresses.
System ID Length Mismatch	The number of times the Brocade device received a PDU whose ID field was a different length than the ID field length configured on the Brocade device.
LSP Sequence Number Skipped	The number of times the Brocade device received an LSP with a sequence number that was more than 1 higher than the sequence number of the previous LSP received from the same neighbor.
LSP Max Sequence Number Exceeded	The number of times the Brocade device attempted to set an LSP sequence number to a value higher than the highest number in the CSNP sent by the Designated IS.
Level-1 Database Overload	The number of times the Level-1 state on the Brocade device changed from Waiting to On or from On to Waiting. • Waiting to On - This change can occur when the Brocade device recovers from a previous Level-1 LSP database overload and is again ready to receive new LSPs. • On to Waiting - This change can occur when the Brocade device's Level-1 LSP database is full and the Brocade device receives an additional LSP, for which there is no room.
Level-2 Database Overload	The number of times the Level-2 state on the Brocade device changed from Waiting to On or from On to Waiting. • The change from Waiting to On can occur when the Brocade device recovers from a previous Level-2 LSP database overload and is again ready to receive new LSPs. • The change from On to Waiting can occur when the Brocade device's Level-2 LSP database is full and the Brocade device receives an additional LSP, for which there is no room.
Our LSP Purged	The number of times the Brocade device received an LSP that was originated by the Brocade device itself and had age zero (aged out).
PDU Drop Count	

TABLE 47 IS-IS error statistics (Continued)

This field...	Displays...
CSNP Auth Failures	The number of CSNP Authentication failures recorded for Level-1 and Level-2. This counter will only be displayed if it has a value greater than zero.
PSNP Auth Failures	The number of PSNP Authentication failures recorded for Level-1 and Level-2. This counter appears only if it has a value greater than 0.
HELLO Auth Failures	The number of HELLO Authentication failures recorded for Level-1 and Level-2. This counter will only be displayed if it has a value greater than zero.
Adjacency not found	The number of PDUs dropped at both Level-1 and Level-2 because there is no valid adjacency on the interface where they were received. This counter will only be displayed if it has a value greater than zero.
Adjacency Level Mismatch	The number of PDUs dropped at both Level-1 and Level-2 because the adjacency from which the PDU is received has a different level than the PDU level. This counter will only be displayed if it has a value greater than zero.
IS Level Mismatch	The number of PDUs dropped at both Level-1 and Level-2 because the IS-IS router level mismatches with the PDU level received. This counter will only be displayed if it has a value greater than zero.
Length Too Short	The number of PDUs dropped at both Level-1 and Level-2 because the received PDU length is less than the standard PDU header length. This counter will only be displayed if it has a value greater than zero.
Length Too Long	The number of PDUs dropped at both Level-1 and Level-2 because the received PDU length is greater than the MTU of the link. This counter will only be displayed if it has a value greater than zero.
Max Area Check Failure	The number of PDUs dropped at both Level-1 and Level-2 because the received PDU has a maximum area count different than what is configured on this IS-IS router. This counter will only be displayed if it has a value greater than zero.
Zero Checksum	The number of PDUs dropped at both Level-1 and Level-2 because the received PDU has a zero checksum. This counter will only be displayed if it has a value greater than zero.
Checksum Mismatch	The number of PDUs dropped at both Level-1 and Level-2 because the received PDU has a checksum different than the computed checksum on the received PDU. This counter will only be displayed if it has a value greater than zero.
Invalid Length	The number of PDUs dropped at both Level-1 and Level-2 because the received PDU has a different length than what is advertised in the PDU header. This counter will only be displayed if it has a value greater than zero.

Displaying the IS-IS SPF Log

The **show isis spf-log** command displays the ISIS Log, as shown in the following.

```
device#show isis spf-log detail
```

```
ISIS Level-1 SPF Log
When      Duration  Nodes  Count  Last-Trigger-LSP         Trigger
0h1m57s   0         3      2      mu1.00-00                Adjacency Change
 Ipv4 Route updates: 4000 Ipv6 Route updates: 0
     First Trigger: 0h1m45s       Adj TLV Changed in LSP mu2.00-00
     Last Trigger : 0h1m45s       Adj TLV Changed in LSP mu1.00-00
0h2m3s    0         3      2      mu2.00-00                New LSP
 Ipv4 Route updates: 4000 Ipv6 Route updates: 0
     First Trigger: 1h42m45s      Adjacency mu2 is added
     Last Trigger : 1h42m45s      New LSP mu2.00-00 Appeared in database
0h2m9s    0         0      3      mu1.00-00                New LSP
 Ipv4 Route updates: 4000 Ipv6 Route updates: 0
     First Trigger: 1h42m45s      Interface ve 3 is Up
     Last Trigger : 1h42m45s      New LSP mu1.00-00 Appeared in database
1h5m12s    0ms      0     1     XMR16.00-00          ISTCT_SPF Computation
ISIS Level-2 SPF Log
When      Duration  Nodes  Count  Last-Trigger-LSP          Trigger
0h2m9s    0         0      3      mu1.00-00                 New LSP
 Ipv4 Route updates: 4000         Ipv6 Route updates: 0
     First Trigger: 1h42m45s      Interface ve 3 is Up
     Last Trigger : 1h42m45s      New LSP mu1.00-00 Appeared in database
0h2m21s   0         0      6      mu1.00-00                 New LSP
 Ipv4 Route updates: 4000         Ipv6 Route updates: 0
     First Trigger: 1h42m45s      Interface eth 1/1 is Up
     Last Trigger : 1h42m45s      New LSP mu1.00-00 Appeared in database
0h3m21s   0         0      3      mu1.00-00                 Adjacency Change
 Ipv4 Route updates: 4000 Ipv6 Route updates: 0
     First Trigger: 1h42m45s      New LSP mu1.00-00 Appeared in database
     Last Trigger : 1h42m45s      Adj TLV is Changed in LSP mu1.00-00
```

Syntax: show isis spf-log { **detail** | **level-1** [**detail**] | **level-2** [**detail**] }

This display shows the following information.

TABLE 48 IS-IS SPF log information

This field...	Displays...
When	When (in hours: minutes : seconds) a full SPF calculation occurred. The last 20 occurrences are logged.
Duration	The time required to complete this SPF run, Elapsed time is normal clock time (not CPU time). Other options for this field are: • Running - the SPF is still running and the duration will be updated after the SFP has run. • Pending - the event is pending and another SPF will be run once the currently executing SPF has completed.
Nodes	The number of routers and pseudonodes (LANs) that make up the topology calculated in this SPF run.
Count	The number of events that triggered this SPF run. When a topology change has occurred, multiple link-state packets (LSPs) are received in a short time. Since a router waits about 5 seconds before running a full SPF run, it can include all new information. This count includes the number of events (such as receiving new LSPs) that occurred while the router was waiting the 5 second interval before running full SPF.
Last Trigger LSP	When a full SPF calculation is triggered by the arrival of a new LSP, the router stores the LSP ID. The LSP ID can provide a clue about the source of routing instability in an area. If multiple LSPs in a single level are causing SPF runs, only the LSP ID of the last received LSP is recorded.
Triggers	The reason that a full SPF calculations was triggered. Table 49 describes each of the triggers that can be displayed in this field.

For a description of the trigger types, refer to the following table.

TABLE 49 Trigger types and description

Trigger	Description
Alternate Route Check	PSPF deleted an IPv4 or IPv6 route. Full SPF must run to find the alternate route.
Route Change in L1 SPF Run	The L1 SPF run added or deleted an IPv4 or IPv6 route. The L2 SPF must run to accommodate this change.
LSP Purged	An LSP was purged. A full SPF calculation must process this change.
LSP Added	A new LSP has appeared in the database. A full SPF calculation is needed to process this new LSP.
Summary Address Change	A summary address configuration change has occurred.
Adjacency State Change	An adjacency was added or deleted.
Admin Distance Change	The administrative distance configuration has changed.
LSP Header Change	The LSP header (attached or overload bits) is changed.
IS Neighbor TLV Change	An IS neighbor TLV was added or deleted in an LSP.
Area Address TLV Change	The area address TLV changed.
Interface IP Address Change	The IP address configuration changed.
IP Address TLV Change	An IP address TLV changed in the LSP.
IPv6 Address TLV Change	An IPv6 address TLV changed in the LSP.
IS-IS Level Change	The IS-IS level configuration changed.
Interface Metric Change	The IS-IS interface metric configuration changed.
LSP Changed - PSPF Disabled	The LSP changed and PSPF is disabled.
LSP Overload Bit Change	The overload bit in the LSP header changed.
Interface State Change	The interface state changed to up or down.
Redist Prefix-List Change	The redistribution list configuration changed.
Redist Policy Change	The redistribution policy configuration changed.
Maximum Path Change	The IS-IS maximum path configuration changed.
IP Load Sharing Change	The IP load sharing configuration changed.

TABLE 49 Trigger types and description (Continued)

Trigger	Description
User Cleared IS-IS Route	The user cleared a specific IS-IS route.
User Cleared IS-IS Routes	The user cleared all IS-IS routes.
Neighbor NLPID Change	NLPID set is changed in received hellos.
ISIS Enable	IS-IS was enabled.
ISTCT_SPF Computation	The user issued the **disable-incremental-stct-spf-opt** command.
User Cleared IS-IS All	The user issued the **clear isis all** command.
Interface Config Change	ISIS was enabled or disabled on a port.
User Trigger	The user issued the **clear isis spf-trigger** command.
Recompute InterLeve Routes	The neighbor IS-type is changed either from L1 to L12 or L12 to L1
Exited Overload State	IS-IS exited from an overload condition.

By using the **detail** option with the**show isis spf-log** command, you can display more detail about the total number of IPv4 and IPv6 route updates and the reason for the first and last SPF events. Like SPF events, the incremental SPF events are displayed. However, for incremental SPF, only the first trigger is displayed, as the example below illustrates. In addition, the logging changes include the number of RTM updates that were carried out in each SPF or incremental SPF run.

To show details about the RTM updates, use the **show isis spf-log detail** command, as follows.

```
device#show isis spf-log detail
ISIS Level-2 SPF Log
 When     Duration Nodes Count Last-Trigger-LSP     Trigger
2h38m9s   0ms      3     2     XMR14.00-00          Adjacency State Change
 Ipv4 Route updates: 4000 Ipv6 Route updates: 0
       First Trigger: 2h39m23s loopback 1 State Changed to Up
       Last Trigger : 2h38m14s Adjacency 0000.0000.0001 Added
2h41m17s 0ms       26    1     XMR14.00-00          IS Neighbor TLV Change
 Ipv4 Route updates: 1 Ipv6 Route updates: 0
       First Trigger: 2h41m2s ISPF Run
```

Syntax: show isis spf-log detail

Clearing the IS-IS SPF Log

You can clear the IS-IS SPF Log accumulated since the last software reload or last clearing of the SPF Log through use of the following command.

```
device# isis clear spf-log
```

Syntax: clear isis spf-log [**level-1** | **level-2**]

When the **level-1** or **level-2** options are used, only the log for the specified level is cleared. If not specified, both will be cleared.

Triggering the router to run SPF

You can trigger the router to run the SPF calculations through use of the following command.

```
device# clear isis spf-trigger
```

Syntax: clear isis spf-trigger [**level-1** | **level-2**]

When the **level-1** or **level-2** options are used, the SPF calculation is only triggered for the specified level. If not specified, the SPF calculation will be triggered for both.

Clearing IS-IS information

To clear the IS-IS information that the Brocade device has accumulated since the last time you cleared information or reloaded the software, use either of the following methods.

To clear IS-IS information, enter the **clear isis all** command at any level of the CLI except the User EXEC level.

```
device# clear isis all
```

This command clears all the following:

- Neighbors (closes the Brocade device's adjacencies with its IS-IS neighbors)
- Routes
- PDU statistics
- Error statistics

Syntax: clear isis all | **counts** | **neighbor** | **route** [**ip-address subnet-mask** | **ip-address/prefix**] | **traffic**

The **all** parameter clears all the IS-IS information. Using this option is equivalent to entering separate commands with each of the other options.

The **counts** parameter clears the error statistics.

The **neighbor** parameter closes the Brocade device's adjacencies with its IS-IS neighbors and clears neighbor statistics.

The **route** [*ip-address subnet-mask* | *ip-address*/*prefix*] parameter clears the IS-IS route table or the specified matching route.

The **traffic** parameter clears the PDU statistics.

NOTE
The **traffic** option also clears the values displayed in the **show isis interface** command's Control Messages Sent and Control Messages Received fields.

The **neighbor** option of the **clear isis** command has been enhanced as described in the following:

Syntax: clear isis neighbor all [**ethernet slot/port** | **pos slot/port** | **tunnel tunnel-id** | **ve port-number**]

The **all** option directs the router to clear all neighbors on all IS-IS interfaces or clear all neighbors on an interface specified using one of the following options:

ethernet *slot / port* - clears all IS-IS neighbors on the specified Ethernet interface.

pos *slot / port* - clears all IS-IS neighbors on the specified POS interface.

ve *port-no* - clears all IS-IS neighbors on the specified virtual interface.

tunnel *tunnel-port* - clears all IS-IS neighbors on the specified tunnel interface.

Syntax: clear isis neighbor sys-id [**ethernet slot/port** | **pos slot/port** | **tunnel tunnel-id** | **ve port-number**]

This command directs the router to clear the IS-IS neighbor specified by the *sys-id* variable on all possible interfaces or to clear the IS-IS neighbor specified by the *sys-id* variable on an interface specified using one of the following options:

ethernet *slot/port* - clears the specified IS-IS neighbor on the specified Ethernet interface.

pos *slot/port* - clears the specified IS-IS neighbor on the specified POS interface.

ve *port-no* - clears the specified IS-IS neighbor on the specified virtual interface.

tunnel *tunnel-port* - clears the specified IS-IS neighbor on the specified tunnel interface.

Clearing a specified LSP from IS-IS database

A new command has been added that allows you to clear a specified LSP from the IS-IS database. Running this command causes the regeneration of the specified LSP where this LSP was originated by this router. For example, to clear the LSP named "XMR-1.00-00" from the IS-IS database, enter the following command.

```
device# clear isis database XMR-1.00-00
```

Syntax: clear isis database lsp-id [**level-1** | **level-2** | **level-1-2**]

The *lsp-id* parameter displays summary information about a particular LSP. Specify an LSPID for which you want to display information in HHHH.HHHH.HHHH.HH-HH format, for example, 3333.3333.3333.00-00. You can also enter name.HH-HH, for example, XMR.00-00.

The**level-1**parameter limits you to clear level-1 LSPs only.

The **level-2** parameter limits you to clear level-2 LSPs only.

The**level-1-2** parameter clears level-1 and level-2 LSPs. This is the default.

NOTE
The**clear isis all** command should be used to regenerate the complete database.

RIP

- RIP feature support....324
- RIP Overview....324
- RIP parameters and defaults....325
- Configuring RIP parameters....327
- Displaying RIP Information....334
- Displaying CPU utilization statistics....336

RIP feature support

The following features are documented in the *Multi-Service IronWare Routing Configuration Guide*.

Features supported	Brocade NetIron XMR Series	Brocade NetIron MLX Series	Brocade NetIron CES 2000 Series BASE package	Brocade NetIron CES 2000 Series ME_PREM package	Brocade NetIron CES 2000 Series L3_PREM package	Brocade NetIron CER 2000 Series Base package	Brocade NetIron CER 2000 Series Advanced Services package
RIP v1 and v2	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Route redistribution into RIP	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Route loop prevention - poison reverse and split horizon	Yes	Yes	Yes	Yes	Yes	Yes	Yes
RIP route filters	Yes	Yes	Yes	Yes	Yes	Yes	Yes
CPU utilization statistics for RIP	Yes	Yes	Yes	Yes	Yes	Yes	Yes

RIP Overview

Routing Information Protocol (RIP) is an IP route exchange protocol that uses a distance vector (a number representing distance) to measure the cost of a given route. The cost is a distance vector because the cost often is equivalent to the number of router hops between the Brocade device and the destination network.

A Brocade device can receive multiple paths to a destination. The software evaluates the paths, selects the best path, and saves the path in the IP route table as the route to the destination. Typically, the best path is the path with the fewest hops. A hop is another router through which packets must travel to reach the destination. If the Brocade device receives a RIP update from another router that contains a path with fewer hops than the path stored in the Brocade device route table, the device replaces the older route with the newer one. The device then includes the new path in the updates it sends to other RIP routers, including Brocade devices.

RIP routers, including the Brocade device, also can modify a route cost, generally by adding to it, to bias the selection of a route for a given destination. In this case, the actual number of router hops may be the same, but the route has an administratively higher cost and is thus less likely to be used than other, lower-cost routes.

A RIP route can have a maximum cost of 15. Any destination with a higher cost is considered unreachable. Although limiting to larger networks, the low maximum hop count prevents endless loops in the network.

Brocade devices support the following RIP versions:

- Version 1 (v1)
- V1 compatible with v2
- Version 2 (v2, the default)

RIP parameters and defaults

You can configure global RIP parameters for the protocol and interface RIP parameters on those interfaces that send and receive RIP information.

RIP global parameters

TABLE 50 RIP global parameters

Parameter	Description	Default
RIP state	The global state of the protocol **NOTE** You also must enable the protocol on individual interfaces. Globally enabling the protocol does not allow interfaces to send and receive RIP information.	Disabled
Administrative distance	The administrative distance is a numeric value assigned to each type of route on the device. When the device is selecting from among multiple routes (sometimes of different origins) to the same destination, the device compares the administrative distances of the routes and selects the route with the lowest administrative distance.	120

TABLE 50 RIP global parameters (Continued)

Parameter	Description	Default
Redistribution	RIP can redistribute routes from other routing protocols such as OSPF and BGP4 into RIP. A redistributed route is one that a router learns through another protocol, then distributes into RIP.	Disabled
Redistribution metric	RIP assigns a RIP metric (cost) to each external route redistributed from another routing protocol into RIP.	1 (one)
Update Interval	How often the router sends route updates to its RIP neighbors.	30 seconds
Learning default routes	The device can learn default routes from its RIP neighbors. **NOTE** You also can enable or disable this parameter on an individual interface basis.	Disabled
Advertising and learning with specific neighbors	The device learns and advertises RIP routes with all its neighbors by default. You can prevent the deivce from advertising routes to specific neighbors or learning routes from specific neighbors.	Learning and advertising permitted for all neighbors

RIP interface parameters

TABLE 51 RIP interface parameters

Parameter	Description	Default
RIP state and version	The state of the protocol and the version that is supported on the interface. The version can be one of the following: • Version 1 only • Version 2 only • Version 1, but also compatible with version 2 **NOTE** You also must enable RIP globally.	Disabled
Metric	A numeric cost the device adds to RIP routes learned on the interface. This parameter applies only to RIP routes.	1 (one)
Learning default routes	Locally overrides the global setting.	Disabled

TABLE 51 RIP interface parameters (Continued)

Parameter	Description	Default
Loop prevention	The method a device uses to prevent routing loops caused by advertising a route on the same interface as the one on which the device learned the route. • Split horizon - The device does not advertise a route on the same interface as the one on which the device learned the route. • Poison reverse - The device assigns a cost of 16 ("infinite" or "unreachable") to a route before advertising it on the same interface as the one on which the device learned the route.	Split horizon **NOTE** Enabling poison reverse disables split horizon on the interface.
Advertising and learning specific routes	You can control the routes that a device learns or advertises.	The device learns and advertises all RIP routes on all interfaces.

Configuring RIP parameters

Use the following procedures to configure RIP parameters on a system-wide and individual interface basis.

Enabling RIP

RIP is disabled by default. To enable RIP, you must enable it globally and also on individual interfaces on which you want to advertise RIP. Globally enabling the protocol does not enable it on individual interfaces. When you enable RIP on a port, you also must specify the version (version 1 only, version 2 only, or version 1 compatible with version 2).

To enable RIP globally, enter the **router rip** command.

```
device(config)# router rip
```

Syntax: [no] router rip

After globally enabling the protocol, you must enable it on individual interfaces. You can enable the protocol on physical interfaces as well as virtual routing interfaces. To enable RIP on an interface, enter commands such as the following.

```
device(config)# interface ethernet 1/1
device(config-if-e01000-1/1)# ip rip v1-only
```

Syntax: [no] ip rip {**v1-only** | **v1-compatible-v2** | **v2-only**}

Configuring metric parameters

By default, a Brocade device port increases the cost of a RIP route that is learned or advertised on the port by one. You can configure individual ports to add more than one to a learned or advertised route's cost.

Changing the administrative distance

By default, the Brocade device assigns the default RIP administrative distance (120) to RIP routes. When comparing routes based on administrative distance, the Brocade device selects the route with the lower distance. You can change the administrative distance for RIP routes.

To change the administrative distance for RIP routes, enter the **distance** command.

```
device(config-rip-router)# distance 140
```

The command changes the administrative distance to 140 for all RIP routes.

Syntax: [no] distance *number*

The *number* variable specifies a range from 1 through 255.

Configuring redistribution

You can configure the Brocade device to redistribute routes learned through Open Shortest Path First (OSPF) or Border Gateway Protocol version 4 (BGP4), connected into RIP, or static routes. When you redistribute a route from one of these other protocols into RIP, the Brocade device can use RIP to advertise the route to its RIP neighbors.

To configure redistribution, perform the following tasks:

1. Configure redistribution filters (optional). You can configure filters to permit or deny redistribution for a route based on its origin (OSPF, BGP4, and so on), the destination network address, and the route's metric. You also can configure a filter to set the metric based on these criteria.
2. Change the default redistribution metric (optional). The Brocade device assigns a RIP metric of 1 to each redistributed route by default. You can change the default metric to a value up to 15.
3. Enable redistribution.

Configuring redistribution filters

RIP redistribution filters apply to all interfaces. Use route maps to define how you want to deny or permit redistribution.

NOTE
The default redistribution action is permit, even after you configure and apply redistribution filters to the virtual routing interface. If you want to tightly control redistribution, apply a filter to deny all routes as the last filter (the filter with the highest ID), then apply filters to allow specific routes.

A route map is a named set of match conditions and parameter settings that the Brocade device can use to modify route attributes and to control redistribution of the routes into other protocols. A route map consists of a sequence of up to 50 instances . The Brocade device evaluates a route according to a route map's instances in ascending numerical order. The route is first compared against instance 1, then against instance 2, and so on. If a match is found, the Brocade device stops evaluating the route against the route map instances.

Route maps can contain **match** statements and **set** statements. Each route map contains a "permit" or "deny" action for routes that match the match statements:

- If the route map contains a permit action, a route that matches a match statement is permitted; otherwise, the route is denied.
- If the route map contains a deny action, a route that matches a match statement is denied.

- If a route does not match any match statements in the route map, the route is denied. This is the default action. To change the default action, configure the last match statement in the last instance of the route map to "permit any any".
- If there is no match statement, the software considers the route to be a match.
- For route maps that contain address filters, AS-path filters, or community filters, if the action specified by a filter conflicts with the action specified by the route map, the route map's action takes precedence over the individual filter's action.

If the route map contains set statements, routes that are permitted by the route map's match statements are modified according to the set statements.

In RIP, the match statements are based on prefix lists and access control lists. Set statements are based on tag values and metric values.

To configure redistribution filters, enter the following command.

```
device(config-rip-router)#redistribute bgp route-map longroute
```

Syntax: [no] redistribute {**connected** | **bgp** | **ospf** | **static** [**metric** *value* | **route-map** *name*]}

The **connected** parameter applies redistribution to connected types.

The **bgp** parameter applies redistribution to BGP4 routes.

The **ospf** parameter applies redistribution to OSPF routes.

The **static** parameter applies redistribution to IP static routes.

The **metric** *value* parameter sets the RIP metric value 1- 15 that will be applied to the routes imported into RIP.

The **route-map** *name* parameter indicates the route map's name.

Matching based on RIP protocol type

The **match** option has been added to the **route-map** command that allows statically configured routes or the routes learned from the IGP protocol RIP.

To configure the route map to match to RIP, enter a command such as the following.

```
device(config-routemap test)# match protocol rip
```

Syntax: [no] match protocol rip

Changing the default redistribution metric

When the Brocade device redistributes a route into RIP, the software assigns a RIP metric (cost) to the route. By default, the software assigns a metric of one to each route that is redistributed into RIP. You can increase the metric that the Brocade device assigns, up to 15.

To change the RIP metric the Brocade device assigns to redistributed routes, enter a command such as the following.

```
device(config-rip-router)# default-metric 10
```

This command assigns a RIP metric of 10 to each route that is redistributed into RIP.

Syntax: [no] default-metric*1-15*

Configuring route learning and advertising parameters

By default, a Brocade device learns routes from all its RIP neighbors and advertises RIP routes to those neighbors.

You can configure the following learning and advertising parameters:

- Update interval - The update interval specifies how often the device sends RIP route advertisements to its neighbors You can change the interval to a value from 3 through 65535 seconds. The default is 30 seconds.
- Learning and advertising of RIP default routes - The Brocade device can learn and advertise RIP default routes. You can disable learning and advertising of default routes on a global or individual interface basis.
- Learning of standard RIP routes - By default, the Brocade device can learn RIP routes from all its RIP neighbors. You can configure RIP neighbor filters to explicitly permit or deny learning from specific neighbors.

Enabling learning of RIP default routes

By default, the Brocade device does not learn default RIP routes. You can enable learning of RIP default routes on a global or interface basis.

To enable learning of default RIP routes on a global basis, enter the following command.

```
device(config-rip-router)# learn-default
```

Syntax: [no] learn-default

To enable learning of default RIP routes on an interface, enter the ip rip learn-default command.

```
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# ip rip learn-default
```

Syntax: [no] ip rip learn-default

Configuring a RIP neighbor filter

By default, a Brocade device learns RIP routes from all its RIP neighbors. Neighbor filters allow you to specify the neighbor routers from which the Brocade device can receive RIP routes. Neighbor filters apply globally to all ports.

To configure a RIP neighbor filters, enter the **neighbor** command.

```
device(config-rip-router)# neighbor 1 deny any
```

This command configures the Brocade device so that the device does not learn any RIP routes from any RIP neighbors.

Syntax: [no] neighbor *filter-num* {**permit** | **deny**} {*source-ip-address* | **any**}

The following commands configure the Brocade device to learn routes from all neighbors except 10.70.12.104. Once you define a RIP neighbor filter, the default action changes from learning all routes from all neighbors to denying all routes from all neighbors except the ones you explicitly permit. Thus, to deny learning from a specific neighbor but allow all other neighbors, you must add a filter that allows learning from all neighbors. Make sure you add the filter to permit all neighbors as the last filter

(the one with the highest filter number). Otherwise, the software can match on the permit all filter before a filter that denies a specific neighbor, and learn routes from that neighbor.

```
device(config-rip-router)# neighbor 2 deny 10.70.12.104
device(config-rip-router)# neighbor 1024 permit any
```

Changing the route loop prevention method

RIP uses the following methods to prevent routing loops:

- Split horizon - The device does not advertise a route on the same interface as the one on which the Brocade device learned the route. This is the default.
- Poison reverse - The device assigns a cost of 16 ("infinite" or "unreachable") to a route before advertising it on the same interface as the one on which the Brocade device learned the route.

These loop prevention methods are configurable on a global basis as well as on an individual interface basis. One of the methods is always in effect on an interface enabled for RIP. Thus, if you disable one method, the other method is enabled.

NOTE
These methods are in addition to RIP's maximum valid route cost of 15.

To disable poison reverse and enable split horizon on a global basis, enter the following command.

```
device(config-rip-router)# no poison-reverse
```

Syntax: [no] poison-reverse

To disable poison reverse and enable split horizon on an interface, enter commands such as the following.

```
device(config)#interface ethernet 1/1
device(config-if-e10000-1/1)# no ip rip poison-reverse
```

Syntax: [no] ip rip poison-reverse

To disable split horizon and enable poison reverse on an interface, enter commands such as the following.

```
device(config)#interface ethernet 1/1
device(config-if-e10000-1/1)# ip rip poison-reverse
```

You can configure the Brocade device to avoid routing loops by advertising local RIP routes with a cost of 16 ("infinite" or "unreachable") when these routes go down.

```
device(config-rip-router)# poison-local-routes
```

Syntax: [no] poison-local-routes

Suppressing RIP route advertisement on a VRRP or VRRPE backup interface

NOTE
This section applies only if you configure the device for Virtual Router Redundancy Protocol (VRRP) or VRRP Extended (VRRPE).

Normally, a VRRP or VRRPE Backup includes route information for the virtual IP address (the backed up interface) in RIP advertisements. As a result, other routers receive multiple paths for the backed up interface and might sometimes unsuccessfully use the path to the Backup rather than the path to the Master.

You can prevent the backups from advertising route information for the backed up interface by enabling suppression of the advertisements.

To suppress RIP advertisements for the backed up interface, enter the following commands.

```
device(config)# router rip
device(config-rip-router)# use-vrrp-path
```

Syntax: [no] use-vrrp-path

The syntax is the same for VRRP and VRRP-E.

Configuring RIP route filters using prefix-lists and route maps

You can configure prefix lists to permit or deny specific routes, then apply them globally or to individual interfaces and specify whether the lists apply to learned routes (in) or advertised routes (out).

You can configure route maps to permit or deny specific routes, then apply a route map to an interface, and specify whether the map applies to learned routes (in) or advertised routes (out).

NOTE
A route is defined by the destination's IP address and network mask.

NOTE
By default, routes that do not match a prefix list are learned or advertised. To prevent a route from being learned or advertised, you must configure a prefix list to deny the route.

To configure a prefix list, enter commands such as the following.

```
device(config)# ip prefix-list list1  permit 10.53.4.1 255.255.255.0
device(config)# ip prefix-list list2  permit 10.53.5.1 255.255.255.0
device(config)# ip prefix-list list3  permit 10.53.6.1 255.255.255.0
device(config)# ip prefix-list list4 deny 10.53.7.1 255.255.255.0
```

The prefix lists permit routes to three networks, and deny the route to one network.

Because the default action is permit, all other routes (routes not explicitly permitted or denied by the filters) can be learned or advertised.

Syntax: [no] ip prefix-list *name* {**permit** | **deny**} {*source-ip-address* | **any** *source-mask* | **any**}

To apply a prefix list at the global level of RIP, enter commands such as the following.

```
device(config-rip-router)# prefix-list list1 in
```

Syntax: no prefix-list *name* {**in** | **out**}

To apply prefix lists to a RIP interface, enter commands such as the following.

```
device(config-if-e1000-1/2)# ip rip prefix-list list2 in
device(config-if-e1000-1/2)# ip rip prefix-list list3 out
```

Syntax: no ip rip prefix-list *name* {**in** | **out**}

In is for Inbound filtering. It applies the prefix list to routes the Brocade device learns from its neighbor on the interface.

Out is for Outbound filtering. It applies the prefix list to routes the Brocade device advertises to its neighbor on the interface.

The commands apply RIP list2 route filters to all routes learned from the RIP neighbor on the port and applies the lists to all routes advertised on the port.

To configure a route-map, enter commands such as the following.

```
device(config)#access-list 21 deny 160.1.0.0 0.0.255.255
device(config)#access-list 21 permit any
device(config)# route-map routemap1 permit 21
device(config-routemap routemap1)# match ip address 21
device(config)# route-map routemap2 permit 22
```

The route-map permit routes to two networks, and denies the route to one network.

Syntax: [no] route-map *map-name* {**permit** | **deny**} *num*

To apply a route map to a RIP interface, enter commands such as the following.

```
device(config-if-e1000-1/2)# ip rip route-map map1 in
```

Syntax: [no] ip rip route-map *name* {**in** | **out**}

The **route-map** can be a prefix list or an ACL. Setting this command can change the metric.

In applies the route map to routes the Brocade device learns from its neighbor on the interface.

Out applies the route map to routes the Brocade device advertises to its neighbor on the interface.

The commands apply route map map1 as route filters to routes learned from the RIP neighbor on the port.

Setting RIP timers

You can set basic update timers for the RIP protocol. The protocol must be enabled in order to set the timers. The **timers** command specifies how often RIP update messages are sent.

To set the timers, enter the following commands.

```
device(config) router rip
device(config-rip-router)# 60 180 180 120
```

Syntax: [no] timers *update-timer timeout-timer hold-down-timer garbage-collection-timer*

The *update-timer* parameter sets the amount of time between RIP routing updates. The possible value ranges from 3 - 65535. The default is 30 seconds.

The *timeout-timer* parameter sets the amount of time after which a route is considered unreachable. The possible value ranges from 9 - 65535. The default is 180 seconds.

The *hold-down-timer* parameter sets the amount of time during which information about other paths is ignored. The possible value ranges from 0 - 65535. The default is 180 seconds.

The *garbage-collection-timer* sets the amount of time after which a route is removed from the rip routing table. The possible value ranges from 0 - 65535. The default is 120 seconds.

Displaying RIP Information

To display RIP filters, enter the following command at any CLI level.

```
device# show ip rip
RIP Summary
  Default port 520
      Administrative distance is 120
      Updates every 30 seconds, expire after 180
      Holddown lasts 180 seconds, garbage collect after 120
      Last broadcast 2, Next Update 26
      Need trigger update  0, Next trigger broadcast 3
      Minimum update  interval 25, Max update Offset 5
      Split horizon is on; poison reverse is off
      Import metric 1
      Prefix List, Inbound : Not set
      Prefix List, Outbound : Not set
      Route-map, Inbound : Not set
      Route-map, Outbound : Not set
      Redistribute:

  No Neighbors are configured in RIP Neighbor Filter Table
```

Syntax: show ip rip

TABLE 52 CLI display of neighbor filter information

Field.	Defiinition
RIP Summary area	Shows the current configuration of RIP on the device.
Static metric	Shows the static metric configuration. ".not defined" means the route map has not been distributed.
OSPF metric	Shows what OSPF route map has been applied.
Neighbor Filter Table area	
Index	The filter number. You assign this number when you configure the filter.
Action	The action the Brocade device takes for RIP route packets to or from the specified neighbor: • deny - If the filter is applied to an interface's outbound filter group, the filter prevents the Brocade device from advertising RIP routes to the specified neighbor on that interface. If the filter is applied to an interface's inbound filter group, the filter prevents the Brocade device from receiving RIP updates from the specified neighbor. • permit - If the filter is applied to an interface's outbound filter group, the filter allows the Brocade device to advertise RIP routes to the specified neighbor on that interface. If the filter is applied to an interface's inbound filter group, the filter allows the Brocade device to receive RIP updates from the specified neighbor.
Neighbor IP Address	The IP address of the RIP neighbor.

To display RIP filters for a specific interface, enter the following command.

```
device#show ip rip interface ethernet 1/1/1
Interface eth 1/20
```

```
 Rip Mode : Version 2 Running: TRUE
 Route summarization disabled
 Split horizon is on; poison reverse is off
 Default routes not accepted
 Metric-offset, Inbound 1
 Metric-offset, Outbound 0
 Prefix List, Inbound : Not set
      Prefix List, Outbound : Not set
 Route-map, Inbound : Not set
Interface ve 10
 RIP Mode : Compatible Running: TRUE
 Route summarization disabled
 Split horizon is off; poison reverse is on
 Default routes not accepted
 Metric-offset, Inbound 1
 Metric-offset, Outbound 0
 Prefix List, Inbound : Not set
 Prefix List, Outbound : Not set
 Route-map, Inbound : Not set
 Route-map, Outbound : Not set
Interface ve 20
 RIP Mode : Version1 Running: TRUE
 Route summarization enabled
 Split horizon is off; poison reverse is on
 Default routes not accepted
 Metric-offset, Inbound 1
 Metric-offset, Outbound 0
 Prefix List, Inbound : Not set
      Prefix List, Outbound : Not set
 Route-map, Inbound : Not set
 Route-map, Outbound : Not set
```

Syntax: show ip rip interface *ifName*

To display RIP route information, enter the following command.

```
device#show ip rip route
RIP Routing Table - 35 entries:
10.0.0.0/8, from 10.1.0.2, ve 10 (2)
         RIP, metric 4, tag 0, timers: aging 17 holddown -163
10.0.0.0/8, from 10.1.1.2, ve 10 (6)
         RIP, metric 16, tag 0, timers: holddown 19 garbage 19
10.1.1.0/24, from 10.0.0.0, eth 1/20 (34)
         MCAST, metric 1, tag 0, timers: none
10.1.0.0/24, from 10.0.0.0, ve 10 (1)
         MCAST, metric 1, tag 0, timers: none
```

Syntax: show ip rip route

To display current running configuration for interface 1/20, enter the following command.

```
device#show running-config interface ethernet 1/20
interface ethernet 1/20
 enable
 ip ospf area 0
 ip ospf priority 0
 ip rip v2-only
 ip address 10.1.1.2/24
 ipv6 address 2000::1/32
 ipv6 enable
!
```

To display current running configuration for ve 10, enter the following command.

```
device#show running-config interface ve 10
interface ve 10
 bfd interval 50 min-rx 50 multiplier 3
 ip ospf area 2
 ip rip v1-compatible-v2
 ip rip poison-reverse
 ip address 10.1.0.1/24
 ipv6 address 2001:db8:1::14/64
!
```

To display current running configuration for ve 20, enter the following command.

```
device#show running-config interface ve 20
interface ve 20
 ip ospf area 1
 ip rip v1-only
 ip rip poison-reverse
 ip address 10.2.0.1/24
!
```

Displaying CPU utilization statistics

You can display CPU utilization statistics for RIP and other IP protocols. To display CPU utilization statistics for RIP, enter the **show cpu-utilization** command at any level of the CLI.

```
device#show cpu-utilization

07:46:48 GMT+00 Mon Nov 18 2013

... Usage average for all tasks in the last 1 seconds  ...
============================================================
Name                    us/sec          %

idle                    978720          100
con                     18              0
mon                     110             0
flash                   116             0
dbg                     27              0
boot                    89              0
main                    0               0
itc                     0               0
tmr                     2201            0
ip_rx                   6413            0
sfm_mgr                 2309            0
scp                     49              0
lpagent                 0               0
console                 227             0
vlan                    163             0
mac_mgr                 84              0
mrp                     219             0
vsrp                    0               0
erp                     214             0
mxrp                    69              0
snms                    0               0
rtm                     796             0
rtm6                    686             0
ip_tx                   2858            0
rip                     0               0
l2vpn                   0               0
mpls                    0               0

(Output truncated)
```

Syntax: show cpu-utilization

The command lists the usage statistics for the previous five-second, one-minute, five-minute, and fifteen-minute intervals.

Policy-Based Routing (IPv4)

- Configuration considerations....338
- Configuring a PBR policy....339
- Configuration examples....344
- Policy based routing with the preserve VLAN option....348
- Configuration examples....349
- Policy-based routing support for preserve VLAN....351

Table 53 displays the individual Brocade devices and the Policy-Based Routing features they support.

TABLE 53 Supported Brocade Policy-Based routing features

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
Policy-Based Routing (PBR)	Yes	Yes	No	Yes	Yes	Yes	Yes
Policy based routing with the preserve VLAN option	Yes	Yes	No	No	No	No	No
Next Hop VLAN Flooding	Yes	Yes	No	No	No	No	No
Policy-Based Routing over a GRE Tunnel	Yes	Yes	No	Yes	Yes	Yes	Yes
Policy-Based Routing over a GRE interface	Yes	Yes	No	Yes	Yes	Yes	Yes
Setting the Output Interface to the Null Interface	Yes	Yes	No	Yes	Yes	Yes	Yes
Selectively Applying Normal Routing to Packets	Yes	Yes	No	Yes	Yes	Yes	Yes
Configure the Route Map	Yes	Yes	No	Yes	Yes	Yes	Yes

TABLE 53 Supported Brocade Policy-Based routing features (Continued)

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
Policy-based routing support for preserve VLAN	Yes	Yes	No	No	No	No	No

Policy-Based Routing (PBR) allows you to use ACLs and route maps to selectively modify and route IP packets in hardware. The ACLs classify the traffic. Route maps that match on the ACLs set routing attributes for the traffic.

A PBR policy specifies the next hop for traffic that matches the policy. Using standard ACLs with PBR, you can route IP packets based on their source IP address. With extended ACLs, you can route IP packets based on all of the match criteria in the extended ACL.

You can configure the Brocade device to perform the following types of PBR based on a packet's Layer 3 and Layer 4 information:

- Select the next-hop gateway.
- Send the packet to the null interface (null0).

When a PBR policy has multiple next hops to a destination, PBR selects the first live next hop specified in the policy that is up. If none of the policy's direct routes or next hops is available, the packets are forwarded as per the routing table.

Configuration considerations

The configuration considerations are as follows:

- A PBR policy on an interface takes precedence over a global PBR policy.
- You cannot apply PBR on a port if that port already has inbound ACLs, inbound ACL-based rate limiting, or TOS-based QoS.
- The number of route maps that you can define is limited by the system memory. When a route map is used in a PBR policy, the PBR policy uses up to 200 instances of a Layer 3 route map, up to 5 ACLs in a matching policy of each route map instance.

The following two conditions can cause more than 200 Layer 3 route-map instances to be used.

1. If one or more of first 200 instances have deny clause.
2. If the access-list used in the first 200 instances is not configured.

    - ACLs with the **log** option configured should not be used for PBR purposes.
    - PBR ignores implicit **deny ip any any** ACL entries, to ensure that for route maps that use multiple ACLs, the traffic is compared to all the ACLs. However, if an explicit **deny ip any any** is configured, traffic matching this clause will be routed normally using Layer 3 paths and will not be compared to any ACL clauses that follow this clause.
    - PBR always selects the first next hop from the next hop list that is up. If a PBR policy's next hop goes down, the policy uses another next hop if available. If no next hops are available, the device routes the traffic in the normal way.

- Any changes to route maps or ACL definitions will be effective immediately for the interfaces where the PBR routemap is applied. There is no need to rebind. However, rebinding is required if a change is made to an IPv6 ACL.
- If a PBR policy is applied globally, inbound ACLs, inbound ACL-based rate-limiting or TOS-based QoS cannot be applied to any port on the device.
- If an IPv4 option packet matches a **deny** ACL filter with the **option** keyword, the packet will be forwarded based on Layer-3 destination. If the **ignore-options** command is configured on the incoming physical port, the packet will be forwarded based on its Layer-3 destination in hardware, otherwise the packet will be sent to the CPU for software forwarding.
- If an IPv4 option packet matches a **permit** ACL filter with the option keyword, it is hardware-forwarded based on its PBR next-hop (if available). If no PBR next-hop is available, the packet is either software or hardware-forwarded (depending on whether **ignore-options** is configured), based on an IP forwarding decision.
- Policy Based Routing (PBR) currently does not support the IPv4 and IPv6 features for changing the MTU.
- Where the next hop is a GRE tunnel:
  - Packets that are larger than the tunnel's MTU are subject to IP fragmentation and PBR processing of the fragmented packets.
  - For route changes of the tunnel destination, the appropriate information is automatically propagated to the PBR feature. Depending on the configuration of the route map, a route change can change the active next hop of the PBR if it leads to the active next hop going down which triggers a new next hop selection process.
- PBR route-map cannot be applied on VPLS, VLL, or VLL-Local endpoints and vice-versa.
- PBR policies are not supported on Layer-3 VPNs.
- In a PBR route-map definition, if even one route-map instance contains a "set next-hop-flood-vlan" statement, all instances of that route-map will apply to both routed and switched traffic.
- Flooding traffic to a POS interface is not allowed. It can only be flooded to Ethernet ports on the VLAN, including the default VLAN.
- When an incoming port is POS then the SA of the outgoing flooded packets will be 0.
- IPv6 PBR to flood VLAN is not supported for switched traffic for the Brocade NetIron CES Series and Brocade NetIron CER Series.

Configuring a PBR policy

To configure PBR, you define the policies using IP ACLs and route maps, then enable PBR globally or on individual interfaces. The device programs the ACLs into the Layer 4 CAM on the interfaces and routes traffic that matches the ACLs according to the instructions in the route maps.

To configure a PBR policy:

- Configure ACLs that contain the source IP addresses for the IP traffic you want to route using PBR.
- Configure a route map that matches on the ACLs and sets the route information.
- Apply the route map to an interface.

Configure the route map

After you configure the ACLs, you can configure a PBR route map that matches based on the ACLs and sets routing information in the IP traffic.

NOTE
The "match" and "set" statements described in this section are the only route-map statements supported for PBR. Other route-map statements described in the documentation apply only to the protocols with which they are described.

NOTE
If none of the clauses of an IPv4 PBR routemap definition contains both 'match' and 'set' statements together, PBR doesn't work and normal routing takes place.

To configure a PBR route map, enter commands such as the following:

```
device(config)# route-map test-route permit 99
device(config-routemap test-route)# match ip address 99
device(config-routemap test-route)# set ip next-hop 192.168.2.1
device(config-routemap test-route)# exit
```

The commands in this example configure an entry in a route map named "test-route". The **match** statement matches on IP information in ACL 99. The **set** statement changes the next-hop IP address for packets that match to 192.168.2.1.

Syntax: [no] route-map map-name permit | deny num

The *map-name* is a string of characters that names the map. Map names can be up to 32 characters in length. You can define an unlimited number of route maps on the Brocade device, as long as system memory is available.

The **permit | deny** parameter specifies the action the Brocade device will take if a route matches a match statement:

- If you specify a **deny** routemap instance, it is ignored and not programmed in Layer- 4 CAM.
- If you specify **permit**, the Brocade device applies the match and set statements associated with this route map instance.

The *num* parameter specifies the instance of the route map you are defining. Routes are compared to the instances in ascending numerical order. For example, a route is compared to instance 1, then instance 2, and so on.

PBR uses up to 200 Layer 3 route map instances for comparison and ignores the rest.

Syntax: [no] match ip address ACL-num-or-name

The *ACL-num-or-name* parameter specifies a standard or extended ACL number or name. Multiple ACLs may be added when separated by spaces.

Setting the next hop

Traffic that matches a match statement in the route map is forwarded as defined by **set** commands. Multiple **set** commands can be configured and when a match condition is met, the device works sequentially through the list of **set** commands until it finds the first "next hop" that is operational and uses it. If that "next hop" goes down, the next hop as defined in a **set** command is chosen and if all next hop interfaces in the list are down, the packet is routed as determined in the IP Route Table. If a next hop interface that was down comes back up, the next hop selection process begins again and restarts its selection process from the top of the list.

Options for setting the next hop are described in the following:

- Setting the Next Hop to an IP Address
- Setting the Next Hop to a GRE Tunnel

- Setting the Next Hop to a GRE Interface
- Setting the Next Hop to a Null Interface
- Setting the Next Hop to an LSP
- Setting the Next Hop to VLAN Flooding

Setting the next hop to an IP address

You can set the next hop to an IP address as shown in the following:

```
device(config)# route-map net10web permit 101
device(config-routemap net10web)# match ip address 101
device(config-routemap net10web)# set ip next-hop 10.1.1.1
```

Syntax: [no] set ip next-hop ip-address

The *ip-address* variable specifies the IP address of the next-hop IP address for traffic that matches a match statement in the route map.

NOTE
If the IP address used in this command is the IP address of a configured GRE tunnel, the configuration will still be accepted but the next-hop selection will never choose this next-hop so it will not become active. If you want to set the next hop using a GRE tunnel, you must use the**set next-hop-ip-tunnel** command.

Setting the next hop to a GRE tunnel

You can set the next hop to a GRE Tunnel as shown in the following:

```
device(config)# interface tunnel 1
device(config-tnif-1)# tunnel mode gre ip
device(config-tnif-1)# tunnel source ethernet 1/2
device(config-tnif-1)# tunnel destination 10.0.8.108
device(config-tnif-1)# ip address 10.10.3.2/24
device(config-tnif-1)# exit
```

Syntax: [no] set next-hop-ip-tunnel tunnel-id

This command sets the next hop to the GRE tunnel identified by the*tunnel-id* variable. Only GRE tunnels are supported by this command. The system will verify if a valid GRE tunnel with the specified*tunnel-id* variable exists. If the*tunnel-id* variable points to a tunnel other than a GRE tunnel or to a non-existent tunnel, the configuration will be rejected.

Values for the *tunnel-id* variable can be from 1 to the maximum number of allowed Tunnel IDs in the system, The maximum number of Tunnel IDs allowed is set using the **system-max ip-tunnels** command.

For additional examples using this command, refer to Setting the next hop to a GRE tunnel on page 346.

Setting the next hop to a GRE physical interface

NOTE
This command is recommended only for use on the BR-MLX 24x10G-DM module.

You can set the next hop to a GRE physical interface as shown in the following:

Syntax: [no] set ip next-hop physical ip interface

This command sets the next hop to the GRE interface identified by the *physical ip interface* variable. Only GRE physical interfaces are supported by this command. The system will verify if a valid GRE interface with the specified*physical ip interface* variable exists.

Setting the next hop to a Null0 interface

NOTE
This feature is not currently supported on the Brocade NetIron CES Series or Brocade NetIron CER Series.

Sending traffic to a NullO Interface drops the traffic. You can set the next hop to a Null0 interface as shown in the following.

```
device(config)# route-map file-13 permit 56
device(config-routemap
file-13)# match ip address 56
device(config-routemap
file-13)# set interface null0
```

Syntax: [no] set interface null0

Setting the next hop to an LSP

You can set the next hop to an LSP as shown in the following.

```
device(config)# route-map pbrmap permit 10
device(config-routemap pbrmap)# match ip address 101
device(config-routemap pbrmap)# set next-hop-lsp t3
```

Syntax: [no] set next-hop-lsp lsp-name

This command allows you to forward matching traffic to an RSVP -signalled LSP that is specified by the *lsp-name* variable.

Setting next hop VLAN flooding

This feature supports the ability to use PBR to forward traffic to a VLAN through use of the "set" command. Using this feature, matched traffic can be flooded on all ports of the VLAN except the incoming physical port. Any PBR policy that contains the **set next-flood-vlan** statement applies to both routed and switched traffic. This means that if any instance in a PBR route-map contains the **set next-flood-vlan** statement, all instances of that route-map will be applied to both routed and switched traffic.

NOTE
Always use Transparent VLAN Flooding for the VLAN that is specified in the **set next-flood-vlan** configuration.

This feature supports IPv6 traffic. The behavioral differences when deployed on a Brocade NetIron CES Series or Brocade NetIron CER Series as compared to when deployed on a Brocade NetIron MLX Series or Brocade NetIron XMR Series are described in Table 54 .

The following example floods all traffic matched from ACL 101 on all ports of VLAN 10 except the incoming physical port.

```
device(config)# access-list 101 permit ip any any
device(config)# route-map calea permit 10
device(config-routemap
```

```
calea)# match ip address 101
device(config-routemap
calea)# set next-flood-vlan 10
device(config-routemap
calea)# exit
```

Syntax: [no] set next-flood-vlan vlan-id [**outgoing da mac-address**]

If the VLAN specified by the *vlan-id* variable is not configured, the PBR route-map set statement will fall through to the next configured set statement. If no valid next-hop is available, the packet is forwarded as per L2/L3 forwarding decision. If the VLAN specified by the *vlan-id* variable has no valid outgoing ports, (such as when all ports in the VLAN are down or when the VLAN is empty) matching packets will be dropped.

The **no set next-hop-flood-vlan** *vlan-id***outgoing-da** *mac-address* command deletes only the outgoing-da option from the set statement. It does not delete the set statement itself. To delete the set statement, the user would have to specify the **no set next-hop-flood-vlan***vlan-id* command.

In the case of traffic incoming on MPLS uplink, PBR to VLAN flooding is only supported for IPv4 traffic, and not for MPLS traffic.

TABLE 54 Behavioral differences of VLAN flooding per platform

Scenario	Brocade NetIron CES Series/ Brocade NetIron CER Series	Brocade NetIron MLX Series/Brocade NetIron XMR Series
Support for Switched Traffic	Not Supported	Supported
If the outgoing-da option is not set	Packet floods with original SA/DA	Switched traffic: Packet floods with original SA/DA. Routed traffic (L3 CAM miss): Packet floods with original SA/DA. Routed traffic (L3 CAM hit): Packet floods with original SA and DA as next-hop MAC.
If the outgoing-da option is set	DA: Configured outgoing-da SA: Outgoing port MAC	DA: Configured outgoing-da SA: Original SA
In set next-hop-flood-vlan, if the VLANID is configured but no ports are added or all ports are down	Go to next set statement.	Drop matching packets.

Enabling PBR

After you configure the ACLs and route map entries, you can enable PBR globally, on individual interfaces, or both as described in this section. To enable PBR, you apply a route map you have configured for PBR globally or locally.

Enabling PBR globally

To enable PBR globally, enter a command such as the following at the global CONFIG level.

```
device(config)# ip policy route-map test-route
```

This command applies a route map named "test-route" to all interfaces on the device for PBR.

Syntax: [no] ip policy route-map map-name

Enabling PBR locally

To enable PBR locally, enter commands such as the following.

```
device(config)# interface ve 1
device(config-vif-1)# ip policy route-map test-route
```

The commands in this example change the CLI to the Interface level for virtual interface 1, then apply the "test-route" route map to the interface. You can apply a PBR route map to Ethernet ports, or virtual interfaces.

Syntax: [no] ip policy route-map map-name

Enter the name of the route map you want to use for the **route-map***map-name* parameter.

Configuration examples

This section presents configuration examples for:

- Basic example on page 344
- Setting the next hop on page 345
- Setting the output interface to the null interface on page 346
- Selectively applying normal routing to packets on page 347

Basic example

The following commands configure and apply a PBR policy that routes HTTP traffic received on virtual routing interface 1 from the 10.10.10.x/24 network to 10.5.5.x/24 through next-hop IP address 10.1.1.1 or, if 10.1.1.x is unavailable, through 10.2.2.1.

```
device(config)# access-list 101 permit tcp 10.10.10.0 10.0.0.255 eq http
10.5.5.0 10.0.0.255
device(config)# route-map net10web permit 101
device(config-routemap
net10web)# match ip address 101
device(config-routemap
net10web)# set ip next-hop 10.1.1.1
device(config-routemap
net10web)# set ip next-hop 10.2.2.1
device(config-routemap
net10web)# exit
device(config)# vlan 10
device(config-vlan-10)# tagged ethernet 1/1 to 1/4
device(config-vlan-10)# router-interface ve 1
device(config)# interface ve 1
device(config-vif-1)# ip policy route-map net10web
```

Setting the next hop

The following commands configure the Brocade device to apply PBR to traffic from IP subnets 10.157.23.x, 10.157.24.x, and 10.157.25.x. In this example, route maps specify the next-hop gateway for packets from each of these subnets:

- Packets from 10.157.23.x are sent to 192.168.2.1.
- Packets from 10.157.24.x are sent to 192.168.2.2.
- Packets from 10.157.25.x are sent to 192.168.2.3.

The following commands configure three standard ACLs. Each ACL contains one of the ACLs listed above. Make sure you specify **permit** instead of deny in the ACLs, so that the Brocade device permits the traffic that matches the ACLs to be further evaluated by the route map. If you specify **deny**, the traffic that matches the **deny** statements are routed normally. Notice that these ACLs specify **any** for the destination address.

```
device(config)# access-list 50 permit 10.157.23.0 10.0.0.255
device(config)# access-list 51 permit 10.157.24.0 10.0.0.255
device(config)# access-list 52 permit 10.157.25.0 10.0.0.255
```

The following commands set an RSVP-signalled LSP as the next hop.

```
device(config)# access-list 101 permit tcp any any
device(config)# access-list 101 deny ip any any
device(config)# router mpls
device(config-mpls)# mpls-interface ethernet 6/1
device(config-mpls)# lsp t3
device(config-mpls-lsp-t3)# to 10.1.1.1
device(config-mpls-lsp-t3)# enable
device(config)# route-map pbrmap permit 10
device(config-routemap pbrmap)# match ip address 101
device(config-routemap pbrmap)# set next-hop-lsp t3
```

The following commands configure three entries in a route map called "test-route". The first entry (permit 50) matches on the IP address information in ACL 50 above. For IP traffic from subnet 10.157.23.0/24, this route map entry sets the next-hop IP address to 192.168.2.1.

```
device(config)# route-map test-route permit 50
device(config-routemap test-route)# match ip address 50
device(config-routemap test-route)# set ip next-hop 192.168.2.1
device(config-routemap test-route)# exit
```

The following commands configure the second entry in the route map. This entry (permit 51) matches on the IP address information in ACL 51 above. For IP traffic from subnet 10.157.24.0/24, this route map entry sets the next-hop IP address to 192.168.2.2.

```
device(config)# route-map test-route permit 51
device(config-routemap test-route)# match ip address 51
device(config-routemap test-route)# set ip next-hop 192.168.2.2
device(config-routemap test-route)# exit
```

The following commands configure the third entry in the test-route route map. This entry (permit 52) matches on the IP address information in ACL 52 above. For IP traffic from subnet 10.157.25.0/24, this route map entry sets the next-hop IP address to 192.168.2.3.

```
device(config)# route-map test-route permit 52
device(config-routemap test-route)# match ip address 52
device(config-routemap test-route)# set ip next-hop 192.168.2.3
device(config-routemap test-route)# exit
```

The following command enables PBR by globally applying the test-route route map to all interfaces.

```
device(config)# ip policy route-map test-route
```

Alternatively, you can enable PBR on specific interfaces, as shown in the following example. The commands in this example configure IP addresses in the three source subnets identified in ACLs 50, 51, and 52, then apply route map test-route the interface.

```
device(config)# interface ve 1
device(config-vif-1)# ip address 10.157.23.1/24
device(config-vif-1)# ip address 10.157.24.1/24
device(config-vif-1)# ip address 10.157.25.1/24
device(config-vif-1)# ip policy route-map test-route
```

Setting the next hop to a GRE tunnel

This section describes how to configure a Brocade device to apply PBR to traffic on port 1/4 from subnets 10.12.13.x and 10.15.16.x. Packets from these subnets are then sent to a next hop that is a GRE tunnel. In this configuration, two GRE tunnels are configured to provide redundancy. If the first tunnel in the configuration (Tunnel 1) is down, traffic will be routed to the second tunnel (Tunnel 2). In situations where both tunnels are down, traffic from the subnets will be routed as directed from the IP route table.

```
device(config)# interface ethernet 1/4
device(config-if-e1000-1/1)# ip policy route-map test1
device(config-if-e1000-1/1)# exit
device(config)# interface tunnel 1
device(config-tnif-1)# tunnel mode gre ip
device(config-tnif-1)# tunnel source ethernet 1/2
device(config-tnif-1)# tunnel destination 10.0.8.108
device(config-tnif-1)# ip address 10.10.3.2/24
device(config-tnif-1)# exit
device(config)# interface tunnel 2
device(config-tnif-2)# tunnel mode gre ip
device(config-tnif-2)# tunnel source ethernet 2/2
device(config-tnif-2)# tunnel destination 10.0.9.108
device(config-tnif-2)# ip address 10.10.4.2/24
device(config-tnif-2)# exit
device(config)# access-list 99 permit 10.12.13.0 10.0.0.255
device(config)# access-list 99 permit 10.15.16.0 10.0.0.255
device(config)# route-map test1 permit 5
device(config-routemap test1)# match ip address 99
device(config-routemap test1)# set next-hop-ip-tunnel 1
device(config-routemap test1)# set next-hop-ip-tunnel 2
```

Setting the output interface to the null interface

The following commands configure a PBR to send all traffic from 10.168.1.204 to the null interface, thus dropping the traffic instead of forwarding it.

```
device(config)# access-list 56 permit 10.168.1.204 0.0.0.0
```

The following commands configure an entry in a route map called "file-13". The first entry (permit 56) matches on the IP address information in ACL 56 above. For IP traffic from the host 10.168.1.204/32, this route map entry sends the traffic to the null interface instead of forwarding it, thus sparing the rest of the network the unwanted traffic.

```
device(config)# route-map file-13 permit 56
device(config-routemap
file-13)# match ip address 56
device(config-routemap
file-13)# set interface null0
device(config-routemap
file-13)# exit
```

The following command enables PBR by globally applying the route map to all interfaces.

```
device(config)# ip policy route-map file-13
```

Alternatively, you can enable the PBR on specific interfaces, as shown in the following example. The commands in this example configure IP addresses in the source subnet identified in ACL 56, then apply route map file-13 to the interface.

```
device(config)# interface ethernet 3/11
device(config-if-e10000-3/11)# ip address 192.168.1.204/32
device(config-if-e10000-3/11)# ip policy route-map file-13
```

Selectively applying normal routing to packets

This example demonstrates how to configure PBR to route all TCP traffic from a host normally while routing all other traffic from the same host through the PBR next hop. In this example, the IP address of the host is 192.168.2.2.

To route TCP traffic from 192.168.2.2 normally, configure a **deny** ACL clause and define it as a **permit route-map** entry as shown in the following.

```
device(config)# access-list 112 deny tcp host 192.168.2.2 any
device(config)# access-list 112 permit ip host 192.168.2.2 any
device(config)# route-map mymap2 permit 10
device(config-routemap mymap2)# match ip address 112
device(config-routemap mymap2)# set ip next-hop 10.1.1.2
```

Applying IPv6 PBR next hop VLAN flooding

```
This example demonstrates how to configure matched traffic to be flooded on all ports
of the VLAN except the incoming physical port.
```

ACL configuration

```
device(config)#ipv6 access-list ipv6acl-3
device(config-ipv6-access-list ipv6acl-3)#permit ipv6 2001:db8:1::/64 2001:db8:2::/64
device(config-ipv6-access-list ipv6acl-3)#deny ipv6 any 2001:db8:2::/64
device(config-ipv6-access-list ipv6acl-3)#exit
```

Route-map configuration

```
device(config)#route-map pbr-2 permit 1
device(config-routemap pbr-2)#match ipv6 address ipv6acl-3
device(config-routemap pbr-2)#set next-hop-flood-vlan 30
device(config-routemap pbr-2)#set next-hop-flood-vlan 40 outgoing-da 0000.1234.5678
device(config-routemap pbr-2)#set next-hop-flood-vlan 20
device(config-routemap pbr-2)#set ipv6 next-hop 2001:db8::2
```

Apply route-map

```
device(config)#interface ethernet 1/3
device(config-if-e1000-1/3)#ipv6 policy route-map pbr-2
device(config)#interface ve 10
device(config-vif-10)#ipv6 policy route-map pbr-2
```

LAG formation

When a LAG is formed, all ports must have the same PBR configuration before deployment, during deployment the configuration on the primary port is replicated to all ports and on undeployment each port inherits the same PBR configuration.

Policy based routing with the preserve VLAN option

When an IP packet matches the PBR policy with the **preserve-vlan** option, the Layer 2 and Layer 3 information is retained (for example, the VLAN information and the MAC address are retained). TTL is not decremented. A packet is sent to the configured next hop. IP packets not matching the PBR policy with **preserve-vlan** will be dropped. If none of the policy's direct routes or next hops is available, the packets are forwarded as per the routing table.

Configuring a physical interface to accept all VLAN packets for PBR

The **allow-all-vlan pbr** command configures a physical interface to accept all VLAN packets for the purpose of PBR. This command reduces configuration complexity since the physical interface does not have to be configured individually in multiple VLAN interfaces.

Syntax: allow-all-vlan pbr

NOTE
The **allow-all-vlan pbr** command cannot be applied to a VE.

Configuration considerations

- The command **allow-all-vlan pbr** cannot be configured when the physical port is configured with an IPv4 address, MPLS, VPLS, VLL, ICL, Layer 3 VPN; or when the port is part of other VLAN.
- The route map with **preserve-vlan** set policies cannot be configured globally.
- A route map used for PBR with a preserve VLAN policy must have the **preserve-vlan** keyword configured for each set policy.

Configuring policy based routing with the preserve VLAN option

The interface, on which PBR with**preserve-vlan** is configured, should be part of the VLANs through which packets are expected. A route map policy with set policies to preserve VLAN can be applied on a physical port or on a VE port.

Preserve VLAN option as part of a set policy

In a route map set policy configuration, the **preserve-vlan** keyword is used to preserve the packet.

Syntax: set ip next-hop *ip-address* **preserve-vlan**

Syntax: set next-hop-flood-vlan *vlan-id* **preserve-vlan**

Syntax: set interface ethernet *slot/port* **preserve-vlan**

Configuring a rule-name for a route-map

You can use the **rule-name** field in the route-map to organize and extract information about PBR configurations.

```
device(config-routemap test)# rule-name test permit 20
```

Syntax: [no] rule-name *rule_name*

The *rule_name* parameter is the name assigned to a specific instance in a route-map. The rule-name may be up to 127 characters in length.

The **no** version of the command removes the name assigned to this instance.

Output example

```
device(config)# route-map xGW_map permit 1
device(config-routemap xGW_map permit 1)# rule-name xGW_path1
device(config-routemap xGW_map permit 1)# match ip address xGW_Filter1
device(config-routemap xGW_map permit 1)# match ipv6 address xGW_Filter1
device(config-routemap xGW_map permit 1)# set next-hop-flood-vlan 2 preserve-vlan
device(config)#route-map xGW_map permit 2
device(config-routemap xGW_map permit 2)# rule-name xGW_path2
device(config-routemap xGW_map permit 2)# match ip address xGW_Filter2
device(config-routemap xGW_map permit 2)# set interface ethernet 15/1 preserve-vlan
```

Limitations

- The same rule names can be used in multiple route maps.
- For each route-map instance, there can be one rule-name configured.
- Within each route-map the rule name has to be unique.
- The show commands will show all the route-maps configured with the same rule name at the same time.
- The maximum number of rule names that can be configured in a system is 4096.

Configuration examples

This section presents the following configuration examples:

- Preserve VLAN IDs and forwarding to single destination on page 349
- Preserve VLAN IDs and replicate to multiple ports within a VLAN on page 350

Preserve VLAN IDs and forwarding to single destination

1. Configure the access list for IPv4.

```
device(config)# access-list 101 permit ip any any
```

2. Configure the route map with a set policy to preserve VLAN for IPv4 traffic.

```
device(config)# route-map map4 permit 10
device(config-routemap map4ve)# match ip address 101
device(config-routemap map4ve)# set ip next-hop 192.168.2.1 preserve-vlan
device(config-routemap map4ve)# exit
```

3. Apply route map to physical or VE interface.

```
device(config)# interface e1/1
device(config-if-e1000-1/1)# allow-all-vlan pbr
device(config-if-e1000-1/1)# ip policy route-map map4
```

```
device(config-if-e1000-1/1)# exit
device(config)# int ve 20
device(config-vif-20)# ip policy route-map map4
```

Preserve VLAN IDs and replicate to multiple ports within a VLAN

1. Configure the route map with set policies to preserve VLAN for IPv4 traffic.

```
device(config)# route-map test permit 100
device(config-routemap test)# match ip address 101
device(config-routemap test)# set next-hop-flood-vlan 200 preserve-vlan
device(config-routemap test)# route-map test permit 300
device(config-routemap test)# exit
```

2. Apply the route map to physical or VE interface.

```
device(config)# interface ethernet 3/1
device(config-if-e1000-3/1)# allow-all-vlan pbr
device(config-if-e1000-3/1)# ip policy route-map test
device(config-if-e1000-3/1)# exit
device(config-)# int ve 10
device(config-vif-10)# ip policy route-map test
```

Specify an interface as a PBR next-hop

Within a route-map instance, there can be only one match statement but multiple **set interface** statements. If there are multiple **set interface** statements configured, the first one in the configured order will be used to forward traffic. If the actively used interface is down, the next interface in the configuration order will take over. If the previously down interface comes back up, the traffic will be reverted to the first interface.

Configure the route map with set policies to preserve VLAN interface.

```
device(config)# route-map test permit 100
device(config-routemap test)# set interface ethernet 3/1 preserve vlan
```

Syntax: set interface ethernet *slot/port* **preserve-vlan**

The **no** version of the command removes the set command from the route-map.

Show telemetry command

To display information related to the telemetry configuration, use the**show telemetry [detail] rule-name** command.

```
device(config)# show telemetry rule-name
```

Syntax: show telemetry [**detail**] **rule-name** [*rule_name*]

The*rule_name* parameter is the rule name for which the information is to be displayed.

For detail mode output, the list of ports will be fully expanded and displayed if the ports are LAG or VLAN ports.

For the output VLAN and output ports information, the command will show the active ports or VLANs that are currently being used for traffic forwarding.

Output examples

The following example shows the output of the **show telemetry rule-name** command.

```
device(config#) show telemetry rule-name
Paths with leading * are configured but disabled, entries with + is for IPv6
                              Route-map    ACL         Output      Output
Rule Name         Input       Policy       Match       VLAN        Port(s)
*test-traffic     N/A         Testing      TestACL     N/A         N/A
*xGW_path1        N/A         other_map    xGW_Filter2 N/A         N/A
xGW_path1         10/1 10/2   xGW_map      xGW_Filter1 2           N/A
xGW_path1        +10/1        xGW_map      xGW_Filter1 2           N/A
xGW_path2         10/1 10/2   xGW_map      xGW_Filter2 None        4/1
xGW_path2        +10/1        xGW_map                  None        4/1
show telemetry rule-name test_traffic
Paths with leading * are configured but disabled, entries with + is for IPv6
                              Route-map    ACL         Output      Output
Rule Name         Input       Policy       Match       VLAN        Port(s)
*test-traffic     N/A         TestMap      TestACL     N/A         N/A
```

The following example shows the output of the **show telemetry detail rule-name** command.

```
device(config#) show telemetry detail rule-name
Rule name: test-traffic (disabled)
Input: None
Route-map Policy: Testing
IPv4 ACL match:  Test_filter1
Output: None
Rule Name: xGW_Path1
Input: IPv4 - 10/1 10/2 IPv6 - 10/1
Route-map Policy: xGW_map
IPv4 ACL match:  xGW_Filter1
IPv6 ACL match:  xGW_Filter1
Output: IPv4 - VLAN 2 IPv6 - VLAN 2
Rule Name: xGW_Path2
Input: IPv4 - 10/1, 10/2 IPv6 - 10/1
Route-map Policy: xGW_map
IPv4 ACL match: xGW_Filter2
Output: IPv4 - 2/2 4/1 IPv6 - 2/2 4/1
```

Policy-based routing support for preserve VLAN

NOTE
Policy-based routing support for preserve VLAN is supported only on Brocade NetIron XMR Series and Brocade NetIron MLX Series routers. This feature is now supported on the BR-MLX-10GX24 module.

Previously, PBR transparent VLAN flooding (TVF) replaced the ingress traffic's VLAN ID with the egress TVF VLAN ID, while flooding the egress TVF VLAN. Policy-based routing support for preserve VLAN allows the ingress packet VLAN header (VLAN ID and priority) to be preserved, while simultaneously flooding the PBR TVF VLAN.

The PBR TVF VLAN egress ports can be in strict tagged VLAN mode or dual VLAN mode. When PBR TVF VLAN egress ports are in strict tagged VLAN mode, the ingress tagged packets flood as "tagged" with the original VLAN ID and priority preserved. The ingress untagged packets flood as "tagged" with the default VLAN ID. When the PBR TVF VLAN egress ports are in dual VLAN mode, the ingress tagged packets flood as "tagged" with the original VLAN ID and priority preserved. The ingress untagged packets flood as "untagged if the egress port is the untagged member of the ingress port's default VLAN; otherwise, the ingress untagged packets flood as "tagged" with the original VLAN ID and priority preserved.

Configuration considerations

Consider the following when policy-based routing is supported for preserve VLAN:

- To preserve the ingress VLAN priority value, the ingress VLAN and the port QoS feature should not be configured at the same time.
- IPv4 and IPv6 ACL VLAN ID matches are supported for both ingress and egress ACLs.
- An egress ACL is supported to filter traffic.
- Policy-based routing support for the preserve VLAN option does not affect the feature implementation of policy-based routing support for the preserve VLAN.

Multi-VRF

- Overview of Multi-VRF 353
- Configuring Multi-VRF 358

Overview of Multi-VRF

Table 55 displays the individual Brocade devices and the Multi-VRF features they support.

TABLE 55 Supported Brocade Multi-VRF features

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
Multi-VRF IPv4	Yes	Yes	No	Yes	Yes	Yes	Yes
Multi-VRF IPv6	Yes	Yes	No	No	Yes	No	Yes
Multi-VRF for IPv4 Unicast - Static routing	Yes	Yes	No	Yes	Yes	Yes	Yes
Multi-VRF for IPv4 Unicast - RIP	Yes	Yes	No	Yes	Yes	Yes	Yes
Multi-VRF for IPv4 Unicast - OSPF	Yes	Yes	No	Yes	Yes	Yes	Yes
Multi-VRF for IPv4 Unicast - BGP	Yes	Yes	No	No	Yes	Yes	Yes
Multi-VRF for IBGP	Yes	Yes	No	No	Yes	Yes	Yes

Virtual Private Networks (VPNs) have been a key application in networking for a long time. Many possible solutions have been proposed over the last several years. Among the many requirements driving this need have been the need for secure transport of sensitive information and controlling information access to those who need it. In large enterprises, particularly those distributed across disparate locations, sensitivity to information pertinent to a department drives the requirement for an IT manager to logically demarcate information flows to be within that department. The need for privacy is another driver behind deployment of VPN solutions.

VPN technologies can be broadly classified into two types:

- secure VPNs
- trusted VPNs.

Secure VPNs require traffic to be encrypted and authenticated and are most important when communication occurs across an infrastructure that is not trusted (e.g. over the public Internet). The most commonly deployed types of secure VPNs are IPSec VPNs and SSL (Secure Sockets Layer) VPNs. Both offer encryption of data streams. While IPSec VPNs operate at the network layer and require special client software, SSL VPNs are more application centric and can generally work with any SSL-enabled browser.

Trusted VPNs ensure integrity and privacy of the data transfers but do not provide any encryption capabilities. Trusted VPNs are most useful when the goal is to leverage a shared infrastructure to allow virtual networks to be built. Examples of such "trusted VPN" technologies include IP or MPLS based Layer 2 VPNs (VPLS, VLL), BGP or MPLS VPNs, ATM or Frame Relay circuits, Layer 2 Tunneling Protocol (L2TP), etc. In short, all these technologies allow a shared infrastructure to be used without compromising the privacy needs of different users or user groups.

Central to Multi-VRF is the ability to maintain multiple "Virtual Routing and Forwarding" (VRF) tables on the same Provider Edge (PE) Router. Multi-VRF uses multiple instances of a routing protocol such as BGP or OSPF to exchange route information for a VPN among peer PE routers. The Multi-VRF capable PE router maps an input customer interface to a unique VPN instance. The router maintains a different VRF table for each VPN instance on that PE router. Multiple input interfaces may also be associated with the same VRF on the router, if they connect to sites belonging to the same VPN. This input interface can be a physical interface or a virtual Ethernet interface on a port.

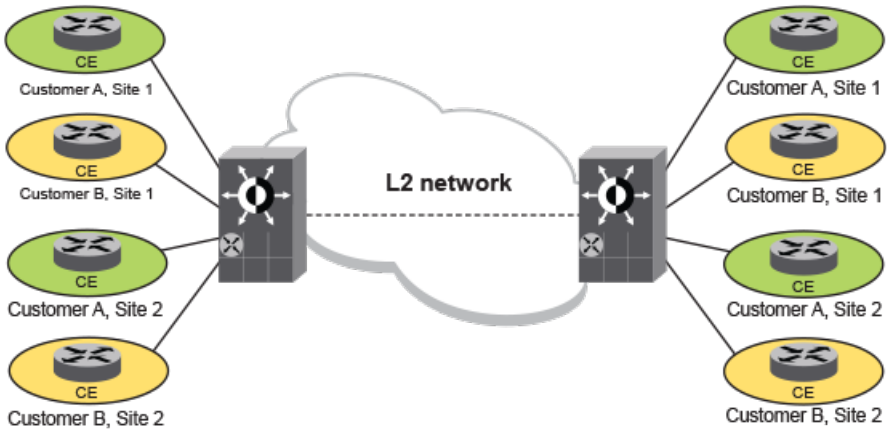
Multi-VRF routers communicate with one another by exchanging route information in the VRF table with the neighboring PE router. This exchange of information among the PE routers is done using BGP or OSPF. The PE routers that communicate with one another should be directly connected at Layer 3. Customers connect to PE routers in the network using Customer Edge (CE) routers as shown in Figure 22 .

Different routing protocols may be used for exchanging information between the PE-PE routers and between the adjacent PE-CE routers. Further, different PE-CE routing protocols may be used in a VPN to exchange customer routes with the various customer sites in that VPN. The routes learned from the PE-CE protocol are added to the corresponding VRF instance and redistributed through the PE-PE protocol to the peer router in the backbone network.

Figure 22 depicts a network using Multi-VRF to provide connectivity among sites that belong to multiple VPNs. To share the VPN route table information with remote PEs, each PE creates separate virtual interfaces and run different instances of the PE-PE routing protocol for each VRF.

NOTE
Some vendors also use the terminology of "Multi-VRF CE" or "VRF-Lite" for this technology.

FIGURE 22 A Network deploying Multi-VRF



Multi-VRF and BGP or MPLS VPNs share some common aspects. For instance, in both cases the edge router maintains a VRF for all directly connected sites that are part of the same VPN. Also in both cases, the PE and CE routers share customer route information using a variety of PE-CE routing protocols, such as OSPF, RIP, E-BGP or static routes. Overlapping address spaces among different VPNs are allowed for both.

There are however, several differences between the two VPN technologies. The fundamental difference between the two technologies is that Multi-VRF requires that peering PE routers be directly connected at Layer 3. A Layer 2 network however, can be present between these directly-connected PE routers. BGP or MPLS VPNs do not have this restriction. In BGP or MPLS VPNs, the MPLS network determines the path to the peer router. In order to distinguish between devices with overlapping IP addresses, route targets are used in BGP or MPLS VPNs. Multi-VRF uses the input interface to uniquely identify the associated VPN, which is why the two PE routers should be directly connected at Layer 3. Table 56 compares Multi-VRF and BGP or MPLS VPNs in more detail

TABLE 56 Comparison between Multi-VRF and BGP or MPLS VPNs

	Multi-VRF	BGP or MPLS VPN
PE-PE Routing Protocol	BGP, OSPF, RIP or Static routing	BGP
PE-CE Routing Protocol	BGP, OSPF, RIP or Static routing	BGP, OSPF, RIP or Static routing
PE-PE Routing Connectivity	PE Routers should be directly connected at Layer 3	PE Routers are interconnected through an IP or MPLS Network
Determination of VRF Instance	Based on input interface only	Based on route target (network interface) or input interface (CE)
Number of Routing Protocol Instances (PE to PE)	Unique routing protocol instance for each VRF instance	Single routing protocol instance
Controlling Advertisement of Routes	No need for route targets to be used. Advertisement on one VRF is independent of advertisement in other VRFs.	Route targets used to identify the customer VPN in advertised routes. The destination PE filters the routes advertised from a peer PE by comparing the route target with the VPNs maintained locally on that PE.
Number of VRF Instances	Unique VRF instance for each VPN	Unique VRF instance for each VPN

TABLE 56 Comparison between Multi-VRF and BGP or MPLS VPNs (Continued)

	Multi-VRF	BGP or MPLS VPN
Overlapping Private Addresses allowed over VPNs?	Yes	Yes
Scalability	Reasonably Scalable	Highly Scalable
MPLS Required	No	Yes

Benefits and applications of Multi-VRF

Multi-VRF provides a reliable mechanism for a network administrator to maintain multiple virtual routers on the same device. The goal of providing isolation among different VPN instances is accomplished without the overhead of heavyweight protocols used in secure VPN technologies or the administrative complexity of MPLS VPNs. It is particularly effective when operational staff has expertise in managing IP networks but may not have the same familiarity in managing MPLS networks. Overlapping address spaces can be maintained among the different VPN instances.

As the two examples in the following sections demonstrate, the simplicity of Multi-VRF allows for several interesting applications.

Example of Multi-VRF usage in an enterprise data center

Figure 23 displays an example of Multi-VRF in an enterprise data center. Each server farm is used for a dedicated application or set of applications. For security reasons, only specific servers in this farm may be allowed to communicate with other servers. Access in some cases may be completely prohibited whereas in other cases access may be allowed through the firewall. Each server is placed on a different subnet. To ensure optimal performance of the data center, trusted servers should be allowed to communicate directly whereas un-trusted servers should not be allowed to directly communicate at all. While Figure 23 shows a limited number of servers; in practice, the number of servers used for this application can run from the tens to the hundreds.

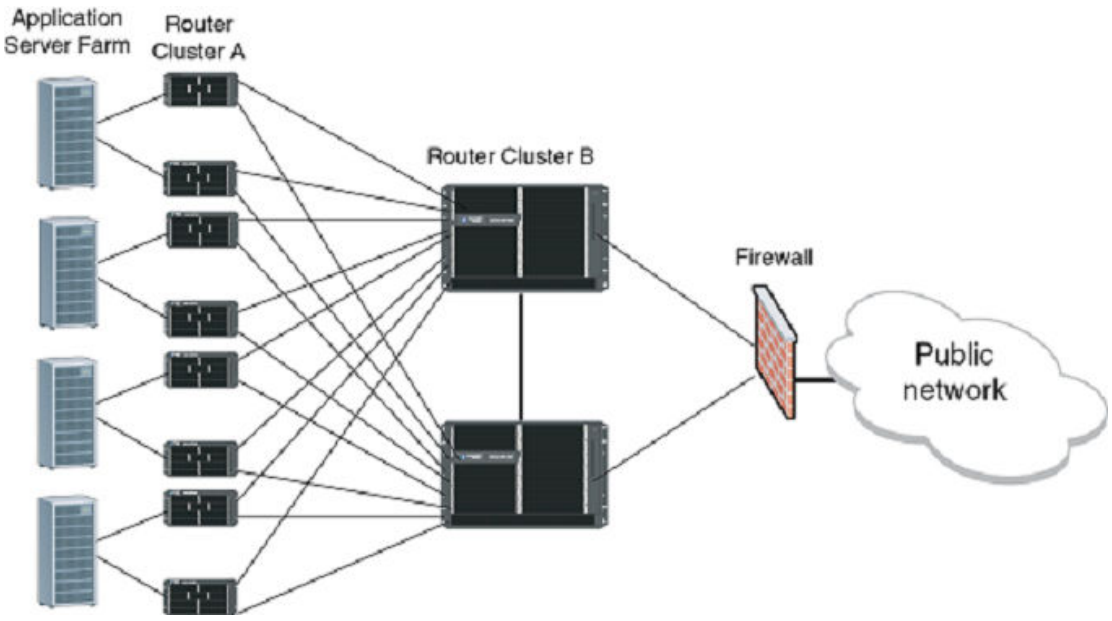
A common way to configure this example is by using Policy Based Routing (PBR). However, because PBR can become very difficult to administer and manage as the network begins to grow, it may require frequent configuration changes which is prone to introducing operator errors.

MPLS VPNs can also be used to configure this example. However, it may be too heavy-weight for what needs to be accomplished in this scenario. In addition, operational staff in enterprise data centers may not always be conversant with administering MPLS.

Secure VPN technologies like IP-Sec are not required here because the infrastructure is already secure. Therefore, the overhead of encryption is not needed.

Multi-VRF is an ideal solution for an application like this example. The servers that are allowed to communicate can be placed in the same Multi-VRF instance. If server access is to be controlled at a more granular level (e.g. at the application layer), then traffic from specific applications on that server can be sent on a specific tagged interface to the router in Cluster A. As shown in Figure 23 , a highly redundant cluster is achieved by ensuring that no single node becomes a point of failure within this network.

FIGURE 23 Example of Multi-VRF usage in an enterprise data center application



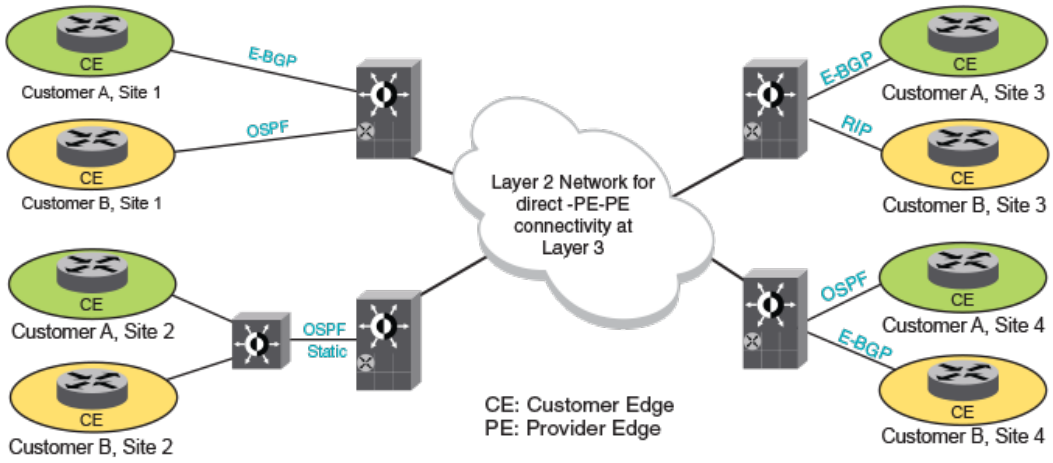
Example of Multi-VRF usage in a service provider network

The figure below depicts the use of Multi-VRF in a typical service provider application. This service provider owns a Layer 2 network connecting the PEs and offers managed VPN services to end users. As shown in the figure below, a host of PE-CE routing protocols can be used-E-BGP, OSPF, RIP or Static Routing.

It is also possible that a site (such as site 2) may have several customers in close geographical proximity as in a business park. This may warrant a dedicated MTU to be placed on-site, which is owned by the service provider. In such a scenario, the different customers may share the same MTU and still use overlapping private address spaces. The MTU is a switch that adds a unique VLAN tag for each connected customer. The PE router (labeled PE2) maps a Layer 3 tagged interface to a unique VRF. Thus, it could be sharing routes using OSPF with one CE and just using Static Routing with another CE (both of these may occur over different virtual interfaces on the same physical interface).

Layer 3 BGP or MPLS VPNs could also be used in a network such as the above. However, if one of the PE routers does not support MPLS or if the operational staff is not conversant with MPLS operations, Multi-VRF provides an alternative mechanism to achieve the same objective.

FIGURE 24 Multi-VRF in a service provider application



Summary

Multi-VRF provides a reliable mechanism for trusted virtual private networks to be built over a shared infrastructure. The ability to maintain multiple virtual routing or forwarding tables allows overlapping private IP addresses to be maintained across VPNs and accomplish goals very similar to that those of more complex VPN technologies such as BGP or MPLS VPNs.

Configuring Multi-VRF

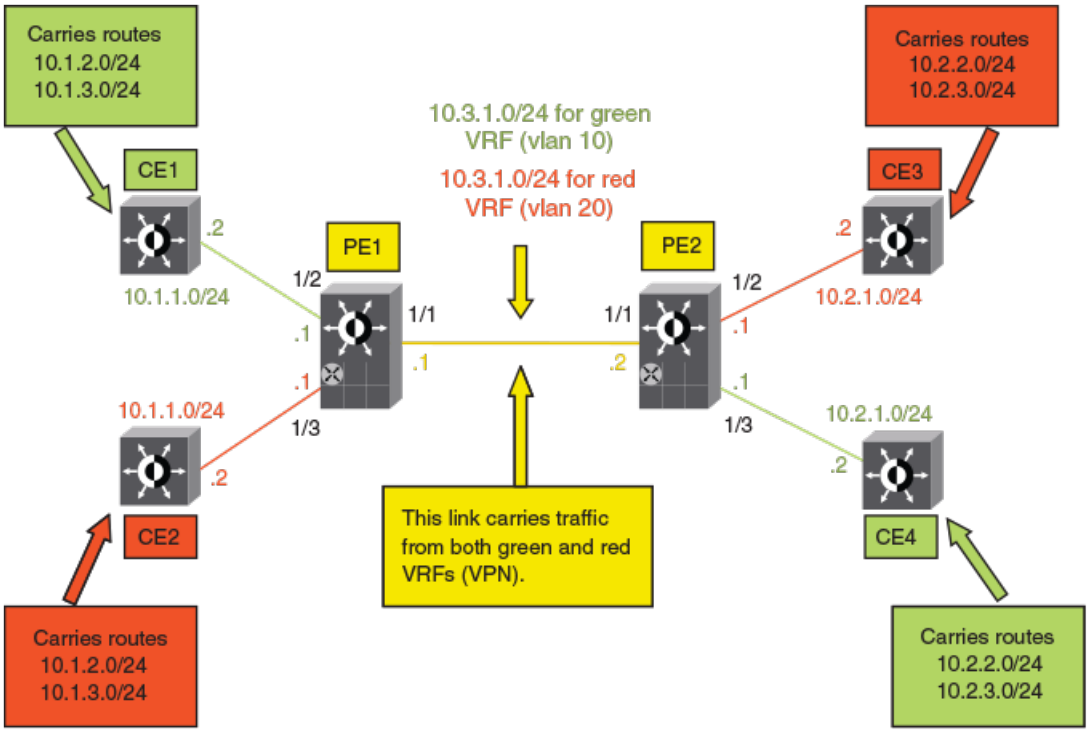
Configuration of the Multi-VRF feature uses the following commands that are defined in Configuring BGP VPNs on a PE in the *Multi-Service IronWare MPLS Configuration Guide*

- Defining a VRF routing instance - This procedure describes how to define a VRF using the**vrf** command.
- Assigning a Route Distinguisher to a VRF - This procedure describes how to define a Route Distinguisher (RD). The RD sets a unique identity to an instance of a VRF. As such, it allows the same IP address to be used in different VPNs without creating any conflict.
- Assigning a VRF routing instance to an interface - This procedure describes how to assign a VRF to one or more virtual or physical interfaces.
- Assigning a VRF routing instance to a LAG interface - This procedure describes how to assign a VRF to a LAG interface.

The main difference between configurations described in Configuring BGP VPNs on a PE and Multi-VRF is that there is no MPLS configuration required for Multi-VRF. This section provides a common Multi-VRF configuration with two possible methods to achieve that configuration.

The diagram in figure below shows a typical network utilizing the multi-VRF feature to implement layer 3 VPNs across 2 directly connected (at layer3) PE routers.

FIGURE 25 Example network topology with both RED and GREEN VPNs



In the diagram in the figure above, CE1 and CE4 are customer edge (CE) routers for the "green" VPN, while CE2 and CE3 belong to "red" VPN. These CE routers can be any routers or layer 3 switches that are capable of running one or many dynamic routing protocols such as BGP, OSPF or RIP or even simple static routing.

The two PEs routers have to be routers that are capable of supporting VRF routing (either with or without MPLS support). In this example, we use two BrocadeBrocade NetIron XMR Series devices. They connect all four CE routers together with a single link between the two of them. Note that this single link between the 2 PEs could also be replaced by a layer-2 switched network if direct physical connection between the PEs is not possible. The only requirement for the connections is that the 2 PEs have to be "directly connected" at layer 3.

Both the customer RED and customer GREEN networks (or VPN) consist of internal routes with overlapping IP address ranges. Thus, traffic communication within each customers' VPN across the 2 PE routers, i.e. between CE1 and CE4, and between CE2 and CE3, must be separated using VRFs.

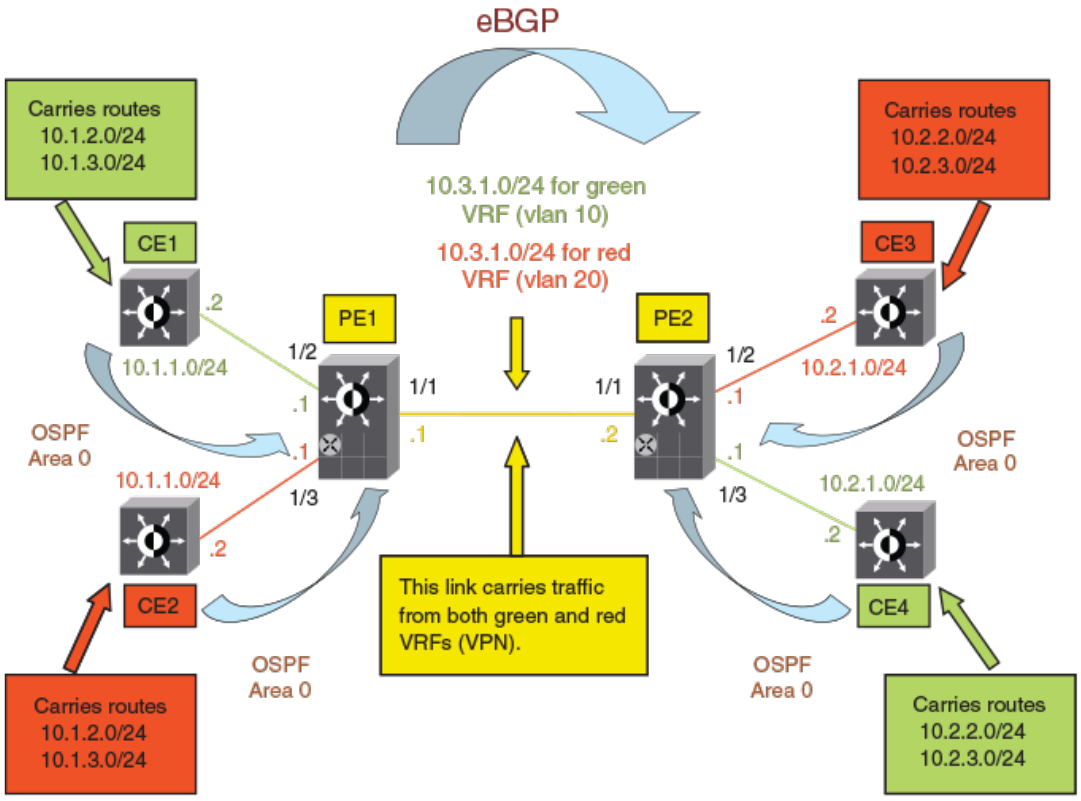
The following sections provide two examples of how to set-up the network shown in the figure above using different routing protocol configurations:

- **Configuration 1** - eBGP Configured between PE1 and PE2 with OSPF (Area 0) Configured between PEs and CEs
- **Configuration 2** - OSPF (Area 0) Configured between PE1 and PE2 with OSPF (Area 1 and Area 2) Configured between PEs and CEs

Configuration 1

As shown in the figure below, eBGP is configured between PE1 and PE2 and OSPF (Area 0) is configured between PEs and CEs.

FIGURE 26 eBGP configured between PE1 and PE2 with OSPF (Area 0) configured between PEs and CEs



The following configuration examples for PE1, PE2, CE1, CE2, CE3, and CE4 describe how to create he example shown in the figure above.

PE1 configuration

In this configuration, VLANs 10 and 20 are created as a link on a tagged port (e 1/10) between PE1 and PE2. Two VRFs ("RED" and "GREEN") are then defined with each having a unique Route Distinguisher (RD). VRF "Green" is assigned an RD value of 10:10, and VRF "Red" is assigned an RD value of 20:20.

In the BGP configuration, PE1 is defined in Local AS1. VRFs "Green" and "Red" are configured and both "Green" and "Red" have the same IP network address assigned (10.3.1.2/24). This is possible because each of the BGP VRF instances have their own separate BGP tables. This is also the same IP network address that will be assigned to VRFs "Green" and "Red" on PE2 within Local AS 2. Redistribution of OSPF routes from PE1's CE peers is enabled to all for their advertisement to PE2.

Both VRFs are configured in Area "0" and directed to redistribute their routes to BGP. The physical interfaces (e 1/2 and e 1/3) to the CEs are assigned to the correct VRF and are configured with the same IP address (10.1.1.1/24) and OSPF Area "0".

The virtual Interfaces (ve10 and ve20) are configured with the same IP address (10.3.1.1/24) and for VRF forwarding in the appropriate VRF (Green or Red).

```
device(config)# vlan 10
device(config-vlan-10)# tagged e 1/1
device(config-vlan-10)# router-interface ve 10
device(config-vlan-10)# vlan 20
```

```
device(config-vlan-20)# tagged e 1/1
device(config-vlan-20)# router-interface ve 20
device(config-vlan-20)# exit
device(config)# vrf green
device(config-vrf-green) rd 10:10
device(config-vrf-green) vrf red
device(config-vrf-red) rd 20:20
device(config-vrf-red) exit-vrf
device(config)# router bgp
device(config-bgp)# local-as 1
device(config-bgp)# address-family ipv4 unicast vrf green
device(config-bgp-ipv4u-vrf)# neighbor 10.3.1.2 remote-as 2
device(config-bgp-ipv4u-vrf)# network 10.3.1.0/24
device(config-bgp-ipv4u-vrf)# redistribute ospf match internal
device(config-bgp-ipv4u-vrf)# redistribute ospf match external1
device(config-bgp-ipv4u-vrf)# redistribute ospf match external2
device(config-bgp-ipv4u-vrf)# exit
device(config)# router bgp
device(config-bgp)# address-family ipv4 unicast vrf red
device(config-bgp-ipv4u-vrf)# neighbor 10.3.1.2 remote-as 2
device(config-bgp-ipv4u-vrf)# network 10.3.1.0/24
device(config-bgp-ipv4u-vrf)# redistribute ospf match internal
device(config-bgp-ipv4u-vrf)# redistribute ospf match external1
device(config-bgp-ipv4u-vrf)# redistribute ospf match external2
device(config-bgp-ipv4u-vrf)# exit
device(config)# router ospf vrf green
device(config-ospf-router-vrf-green)# area 0
device(config-ospf-router-vrf-green)# redistrbution bgp
device(config-ospf-router-vrf-green)# exit
device(config)# router ospf vrf red
device(config-ospf-router-vrf-red)# area 0
device(config-ospf-router-vrf-red)# redistrbution bgp
device(config-ospf-router-vrf-red)# exit
device(config)# interface ethernet 1/2
device(config-if-e1000-1/2)# vrf forwarding green
device(config-if-e1000-1/2)# ip ospf area 0
device(config-if-e1000-1/2)# ip address 10.1.1.1/24
device(config-if-e1000-1/2)# interface ethernet 1/3
device(config-if-e1000-1/3)# vrf forwarding red
device(config-if-e1000-1/3)# ip ospf area 0
device(config-if-e1000-1/3)# ip address 10.1.1.1/24
device(config-if-e1000-1/3)# exit
device(config)# interface ve 10
device(config-vif-10)# vrf forwarding green
device(config-vif-10)# ip address 10.3.1.1/24
device(config-vif-10)# Interface ve 20
device(config-vif-10)# vrf forwarding red
device(config-vif-10)# ip address 10.3.1.1/24
```

PE2 Configuration:

The PE2 configuration is a mirror image of the PE1 configuration. The only difference is that the BGP neighbor is port 1/1 on PE1 which has an IP address of 10.3.1.1. This is used in the BGP configuration.

```
device(config)# vlan 10
device(config-vlan-10)# tagged e 1/1
device(config-vlan-10)# router-interface ve 10
device(config-vlan-10)# vlan 20
device(config-vlan-20)# tagged e 1/1
device(config-vlan-20)# router-interface ve 20
device(config-vlan-20)# exit-vrf
device(config)# vrf green
device(config-vrf-green) rd 10:10
device(config-vrf-green) vrf red
device(config-vrf-red) rd 20:20
device(config-vrf-red) exit
device(config)# router bgp
device(config-bgp)# local-as 1
device(config-bgp)# address-family ipv4 unicast vrf green
device(config-bgp-ipv4u-vrf)# neighbor 10.3.1.1 remote-as 2
device(config-bgp-ipv4u-vrf)# network 10.3.1.0/24
device(config-bgp-ipv4u-vrf)# redistribute ospf match internal
device(config-bgp-ipv4u-vrf)# redistribute ospf match external1
device(config-bgp-ipv4u-vrf)# redistribute ospf match external2
device(config-bgp-ipv4u-vrf)# exit
```

```
device(config)# router bgp
device(config-bgp)# address-family ipv4 unicast vrf red
device(config-bgp-ipv4u-vrf)# neighbor 10.3.1.1 remote-as 2
device(config-bgp-ipv4u-vrf)# network 10.3.1.0/24
device(config-bgp-ipv4u-vrf)# redistribute ospf match internal
device(config-bgp-ipv4u-vrf)# redistribute ospf match external1
device(config-bgp-ipv4u-vrf)# redistribute ospf match external2
device(config-bgp-ipv4u-vrf)# exit
device(config)# router ospf vrf green
device(config-ospf-router-vrf-green)# area 0
device(config-ospf-router-vrf-green)# redistrbution bgp
device(config-ospf-router-vrf-green)# exit
device(config)# router ospf vrf red
device(config-ospf-router-vrf-red)# area 0
device(config-ospf-router-vrf-red)# redistrbution bgp
device(config-ospf-router-vrf-red)# exit
device(config)# interface ethernet 1/2
device(config-if-e1000-1/2)# vrf forwarding green
device(config-if-e1000-1/2)# ip ospf area 0
device(config-if-e1000-1/2)# ip address 10.1.1.1/24
device(config-if-e1000-1/2)# interface ethernet 1/3
device(config-if-e1000-1/3)# vrf forwarding red
device(config-if-e1000-1/3)# ip ospf area 0
device(config-if-e1000-1/3)# ip address 10.1.1.1/24
device(config-if-e1000-1/3)# exit
device(config)# interface ve 10
device(config-vif-10)# vrf forwarding green
device(config-vif-10)# ip address 10.3.1.1/24
device(config-vif-10)# Interface ve 20
device(config-vif-10)# vrf forwarding red
device(config-vif-10)# ip address 10.3.1.1/24
```

CE 1 and CE 2 configurations

The CE1 and CE2 router configurations are exactly the same. Both are configured in OSPF Area 0 with route redistribution enabled. The IP addresses: 10.1.2.1/32 and 10.1.3.1/32 are configured for the Loopback1 interface allowing them to carry routes from these networks.

```
device(config)# router ospf
device(config-ospf-router)# area 0
device(config-ospf-router)# redistribution connected
device(config-ospf-router)# exit
device(config)# interface loopback 1
device(config-lbif-1)# ip address 10.1.2.1/32
device(config-lbif-1)# ip address 10.1.3.1/32
device(config-lbif-1)# interface ethernet 1/1
device(config-if-e1000-1/1)# ip ospf area 0
device(config-if-e1000-1/1)# ip address 10.1.1.2/24
```

CE 3 and CE 4 configurations

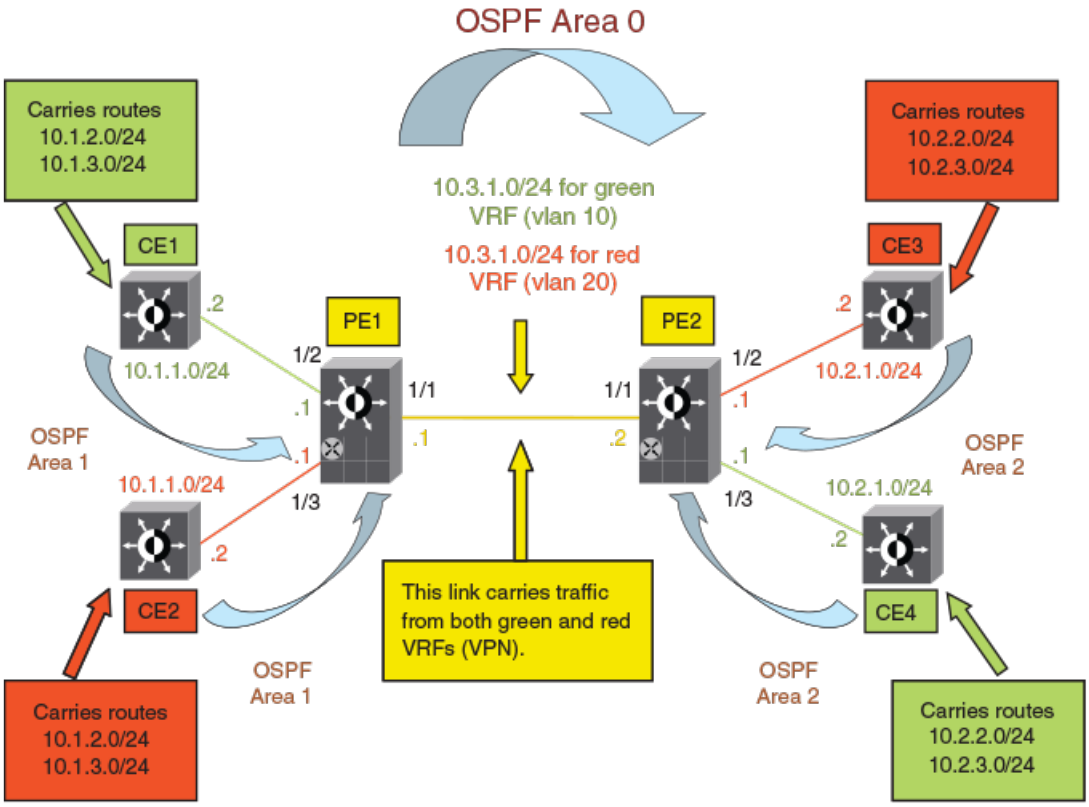
The CE3 and CE4 router configurations are exactly the same. Both are configured in OSPF Area 0 with route redistribution enabled. The IP addresses: 10.2.2.1/32 and 10.2.3.1/32 are configured for the Loopback1 interface allowing them to carry routes from these networks.

```
device(config)# router ospf
device(config-ospf-router)# area 0
device(config-ospf-router)# redistribution connected
device(config-ospf-router)# exit
device(config)# interface loopback 1
device(config-lbif-1)# ip address 10.2.2.1/32
device(config-lbif-1)# ip address 10.2.3.1/32
device(config-lbif-1)# interface ethernet 1/1
device(config-if-e1000-1/1)# ip ospf area 0
device(config-if-e1000-1/1)# ip address 10.2.1.2/24
```

Configuration 2

As shown in the figure below, OSPF (Area 0) is configured between PE1 and PE2 and OSPF (Area 1 and Area 2) is configured between PEs and CEs. The biggest difference between this configuration and Configuration 1 is that OSPF is used between the 2 PEs instead of eBGP. Otherwise most of the configuration is the same as Configuration 1.

FIGURE 27 OSPF (Area 0) configured between PE1 and PE2 with OSPF (Area 1 and Area 2) configured between PEs and CEs



The following configuration examples for PE1, PE2, CE1, CE2, CE3, and CE4 describe how to create the example shown in the figure above.

PE1 configuration:

In this configuration, VLANs 10 and 20 are created as a link on a tagged port (e 1/10) between PE1 and PE2. Two VRFs ("RED" and "GREEN") are then defined with each having a unique Route Distinguisher (RD). VRF "Green" is assigned an RD value of 10:10, and VRF "Red" is assigned an RD value of 20:20.

Because OSPF is the only routing protocol used in this set-up, multiple OSPF areas are used. Area 0 is configured between the two PEs. Area 1 is configured PE1 and CE's 1 and 2. Area 0 is configured between the two PEs. Area 2 is configured PE2 and CE's 3 and 4.

The virtual Interfaces (ve10 and ve20) are configured with the same IP address (10.3.1.1/24) and for VRF forwarding in the appropriate VRF (Green or Red). Both are also configured in OSPF Area 0.

```
device(config)# vlan 10
device(config-vlan-10)# tagged e 1/1
device(config-vlan-10)# router-interface ve 10
```

```
device(config-vlan-10)# vlan 20
device(config-vlan-20)# tagged e 1/1
device(config-vlan-20)# router-interface ve 20
device(config-vlan-20)# exit-vrf
device(config)# vrf green
device(config-vrf-green) rd 10:10
device(config-vrf-green) vrf red
device(config-vrf-red) rd 20:20
device(config-vrf-red) exit-vrf
device(config)# router ospf vrf green
device(config-ospf-router-vrf-green)# area 0
device(config-ospf-router-vrf-green)# area 1
device(config-ospf-router-vrf-green)# exit
device(config)# router ospf vrf red
device(config-ospf-router-vrf-red)# area 0
device(config-ospf-router-vrf-red)# area 1
device(config-ospf-router-vrf-red)# exit
device(config)# interface ethernet 1/2
device(config-if-e1000-1/2)# vrf forwarding green
device(config-if-e1000-1/2)# ip ospf area 1
device(config-if-e1000-1/2)# ip address 10.1.1.1/24
device(config-if-e1000-1/2)# interface ethernet 1/3
device(config-if-e1000-1/3)# vrf forwarding red
device(config-if-e1000-1/3)# ip ospf area 1
device(config-if-e1000-1/3)# ip address 10.1.1.1/24
device(config-if-e1000-1/3)# exit
device(config)# interface ve 10
device(config-vif-10)# vrf forwarding green
device(config-vif-10)# ip ospf area 0
device(config-vif-10)# ip address 10.3.1.1/24
device(config-vif-10)# Interface ve 20
device(config-vif-20)# vrf forwarding red
device(config-vif-20)# ip ospf area 0
device(config-vif-20)# ip address 10.3.1.1/24
```

PE2 configuration:

The PE2 configuration is a mirror image of the PE1 configuration. The only difference is that PE2 connects to CE3 and CE 4 in OSPF Area 2.

```
device(config)# vlan 10
device(config-vlan-10)# tagged e 1/1
device(config-vlan-10)# router-interface ve 10
device(config-vlan-10)# vlan 20
device(config-vlan-20)# vlan 20
device(config-vlan-20)# tagged e 1/1
device(config-vlan-20)# router-interface ve 20
device(config-vlan-20)# exit
device(config)# vrf green
device(config-vrf-green) rd 10:10
device(config-vrf-green) vrf red
device(config-vrf-red) rd 20:20
device(config-vrf-red) exit
device(config)# router ospf vrf green
device(config-ospf-router-vrf-green)# area 0
device(config-ospf-router-vrf-green)# area 2
device(config-ospf-router-vrf-green)# exit
device(config)# router ospf vrf red
device(config-ospf-router-vrf-red)# area 0
device(config-ospf-router-vrf-red)# area 2
device(config-ospf-router-vrf-red)# exit
device(config)# interface ethernet 1/2
device(config-if-e1000-1/2)# vrf forwarding red
device(config-if-e1000-1/2)# ip ospf area 2
device(config-if-e1000-1/2)# ip address 10.2.1.1/24
device(config-if-e1000-1/2)# interface ethernet 1/3
device(config-if-e1000-1/3)# vrf forwarding green
device(config-if-e1000-1/3)# ip ospf area 2
device(config-if-e1000-1/3)# ip address 10.2.1.1/24
device(config-if-e1000-1/3)# exit
device(config)# interface ve 10
device(config-vif-10)# vrf forwarding green
device(config-vif-10)# ip ospf area 0
device(config-vif-10)# ip address 10.3.1.1/24
device(config-vif-10)# Interface ve 20
```

```
device(config-vif-20)# vrf forwarding red
device(config-vif-20)# ip ospf area 0
device(config-vif-20)# ip address 10.3.1.1/24
```

CE 1 and CE 2 configurations

The CE1 and CE2 router configurations are exactly the same. Both are configured in OSPF Area 1 with route redistribution enabled. The IP addresses: 10.1.2.1/24 and 10.1.3.1/24 are configured for the Loopback1 interface allowing them to carry routes from these networks.

```
device(config)# router ospf
device(config-ospf-router)# area 1
device(config-ospf-router)# redistribution connected
device(config-ospf-router)# exit
device(config)# interface loopback 1
device(config-lbif-1)# ip address 10.1.2.1/24
device(config-lbif-1)# ip address 10.1.3.1/24
device(config-lbif-1)# interface ethernet 1/1
device(config-if-e1000-1/1)# ip ospf area 1
device(config-if-e1000-1/1)# ip address 10.1.1.2/24
```

CE 3 and CE 4 configurations

The CE3 and CE4 router configurations are exactly the same. Both are configured in OSPF Area 2 with route redistribution enabled. The IP addresses: 10.2.2.1/24 and 10.2.3.1/24 are configured for the Loopback1 interface allowing them to carry routes from these networks

```
device(config)# router ospf
device(config-ospf-router)# area 2
device(config-ospf-router)# redistribution connected
device(config-ospf-router)# exit
device(config)# interface loopback 1
device(config-lbif-1)# ip address 10.1.2.1/24
device(config-lbif-1)# ip address 10.1.3.1/24
device(config-lbif-1)# interface ethernet 1/1
device(config-if-e1000-1/1)# ip ospf area 2
device(config-if-e1000-1/1)# ip address
```

Inter-VRF Routing

- Features and benefits..... 367
- Configuration considerations.....369
- Maximum route limitations..... 369
- Configuring Inter-VRF routing..... 370
- Clearing IP routes..... 375
- Configuring the number of VRFs for IPv4 and IPv6.....377
- Modified CLI commands..... 377

Table 57 displays the individual Brocade devices and the Inter-VRF Routing features they support.

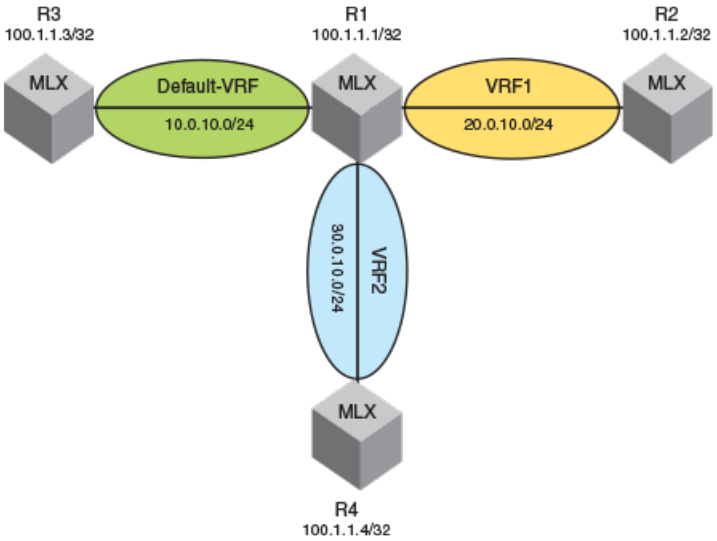
TABLE 57 Supported Brocade Inter-VRF features

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
Inter-VRF Routing IPv4	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Inter-VRF Routing IPv6	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Inter-VRF routing overview

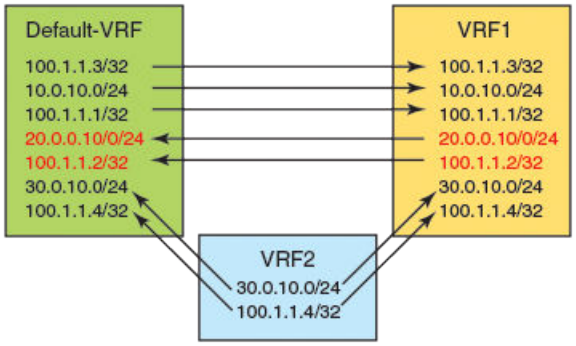
Inter-VRF routing feature permits routes from one VRF to import into other VRFs. This feature is useful in cases where all the VRFs share the same path to reach the external domain, but each VRF can still keep its internal routing information limited to its own VRF. Currently, Brocade NetIron series routers permit static routes to be configured across VRFs. User can configure to import routes from one VRF to other VRFs through configuration. Figure 28 depicts a network using inter-VRF to provide connectivity among sites that belong to multiple VPNs. To share the VPN routing table information with remote PEs, each PE creates separate virtual interfaces and runs different instances of the PE-PE routing protocol for each VRF.

FIGURE 28 A Network deploying Inter-VRF



Following are the route entries and routing tables in Router R1.

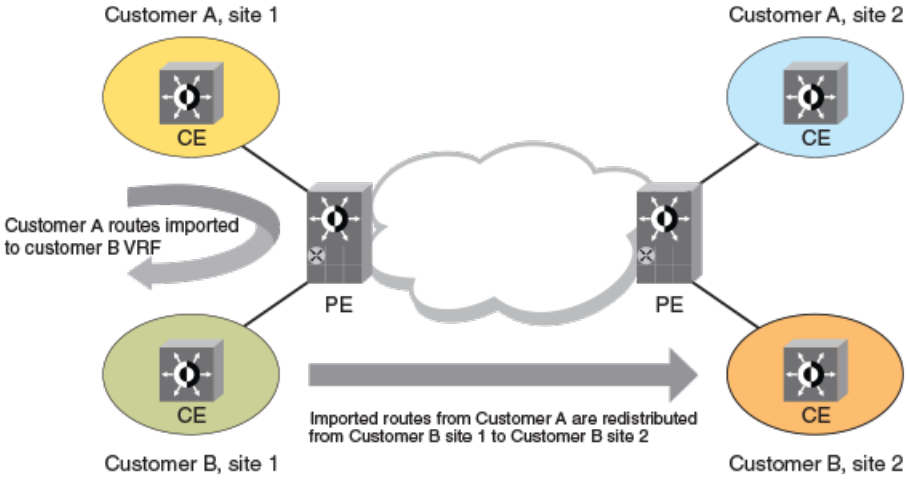
FIGURE 29 Router R1 route-entries and route-tables



Features and benefits

Inter-VRF routing feature allows customers to selectively access each other's networks through configuration. It allows all VRFs to share the same path to the external domain while keeping internal routing information separate.

FIGURE 30 Inter-VRF routing topology



To import routes from multiple VRFs, multiple import commands need to be defined. The filtering criteria for routes to be imported are specified using route-maps by the user. Routes can be filtered based on BGP attributes, interfaces, IP addresses, next hops, metrics, metric types, protocols, route types and tags which are supported by our existing route-map infrastructure.

Routes from multiple (default 50) VRFs can be selectively imported into a target destination VRF. Imported routes can be redistributed using routing protocols.

Configuring IPv6 inter-VRF routing is very similar to configuring inter-VRF routing for IPv4. The behavior and CLI syntax of route-maps is the same between IPv4 and IPv6 address families.

The route-map attributes that are used for filtering the routes are:

TABLE 58 IPv4 Route-map handling

Attributes used for filtering routes	Attributes set using a route map
IP address (Prefix list, Access list)	
Next hop (Prefix list, Access list)	Metric value
Metric value	Nexthop
Tag type	Distance
Route type	Tag
BGP attributes (AS path, Community, Ext community access list)	
Interface type	
Protocol type	

TABLE 59 IPv6 Route-map handling

Attributes used for filtering routes	Attributes set using a route map
IP address (Prefix list)	
Next hop (Prefix list)	Metric value
Metric value	Nexthop
Tag type	Distance
Route type	Tag
BGP attributes (AS path, Community, Ext community access list)	
Interface type	
Protocol type	

Configuration considerations

These are the things to consider while configuring the Brocade device:

- Import configuration commands allow specifying a non-existing source VRF. Routes will be imported dynamically when the source VRF is created.
- If the route-map is empty, then no course of actions are applied while importing the routes.
- If you change the configuration of a route-map, then all the VRFs which are configured to use this route-map will be processed again.

Tie breaker rules

The rules in the sequence below apply to break the tie when the same routes are imported from multiple VRFs including the local route:

- If routes are originated from different protocols, then the protocol with the best administrative distance will be used to break the tie.
- If the routes' origin (protocol) are the same, then the metric value will be used to break the tie.
- If the metric value is the same, then routes learned in local VRF will be used to break the tie.
- If the metric value is the same, and a local VRF route is not available, then the lowest nexthop address will be used to break the tie.
- If the nexthop address is the same, then the oldest route will be used to break the tie.

Maximum route limitations

When importing routes from other VRFs, there may be a chance that routes are not added due to a limitation on the number of routes that the destination VRF can support. This may happen in the following cases:

1. The source VRF is importing more routes than the destination VRF can support.
2. The destination VRF is configured to limit the number of routes with a configuration command such as address-family **ipv4 max-route** or **address-family ipv6 max-route** .
3. While processing the route-map changes, you exceed the number of routes that the VRF can support because you process the new set of routes before deleting the old set of routes.

   For any of the above situations, execute the **clear ip route VRF dest-vrf-name** followed by the **import***src-vrf-name* command to recover.

Configuring Inter-VRF routing

The following configuration steps allow the VRF VPN to import IPv4 routes from the **default-vrf brcd-sj**.

```
device(config)#vrf vpn
device(config-vrf-vpn)#address-family ipv4
device(config-vrf-vpn-ipv4)#import routes vrf default-vrf route-map brcd-sj
```

The following configuration allows the default-vrf to import IPv4 routes from the non-default VRF VPN after satisfying conditions specified in the route-map brcd-sj.

```
device(config)#ip import routes vrf vpn route-map brcd-sj
```

From non-default VRF, user can configure the command in address-family mode.

Syntax: import routes vrf *vrf-name* **route-map** *route-map-name*

This command imports the IPv4 routes from src-vrf to dest-vrf. The route-map import-map is applied while importing the routes.

Syntax: import routes vrf *src-vrf* **route-map** *import-map*

From default VRF the commands for IPv4 and IPv6 are as below.

These commands import the IPv4 and IPv6 routes from src-vrf to default-vrf using the route-map import-map.

```
device(config)#ip import routes vrf src-vrf route-map import-map>
device(config)#ipv6 import routes vrf src-vrf route-map import-map
```

TABLE 60 Route-map for non-default and default VRF

This field...	Displays
src-vrf	The source VRF from where the routes have been imported to the destination VRF. The "-" in the src-vrf column output denotes the route is local route.
import-map	Route-map which has clauses to filter the routes coming from src-vrf.
Defaults	By default no routes will be imported in to dest-vrf.
Range	User can configure a maximum of 50 import commands for a given VRF per address-family. If user tries to configure more than 50 commands then the configuration will be rejected and an error message will be generated.

TABLE 60 Route-map for non-default and default VRF (Continued)

This field...	Displays
No	The no command removes the configuration and all the routes imported from src-vrf will be removed in the dest-vrf.

NOTE
Warning message will be displayed when import route is issued for non-existing VRF as follows: "VRF *vrf-name* is not created yet"

NOTE
Warning message will be displayed when VRF is getting deleted by the user and exports routes to other VRFs: "All IPv4 and IPv6 export routes in VRF one have been removed"

Show commands

Use the following commands to display the IPv4 and IPv6 routing configuration on the device and to include the maximum allowed import VRFs.

```
device# show ip
```

This command will display the maximum IPv4 allowed import VRFs and list of import VRFs to default-vrf. The following is an example of this enhancement to **show ip** .

```
device(config)#show ip
Global Settings
IP CAM Mode: static IPVPN CAM Mode: static
  ttl: 64, arp-age: 10, bootp-relay-max-hops: 4, icmp-error-rate: 400
  IP Router-Id: 10.0.0.1   load-sharing path: 4
  enabled : UDP-Broadcast-Forwarding  ICMP-Redirect  ICMP-MPLS-Response  Source-
Route  Load-Sharing
 RARP  RIP  BGP4  IS-IS  OSPF  VRRP
  disabled: Directed-Broadcast-Forwarding  drop-arp-pending-packets  IRDP  Proxy-ARP
RPF-Check  RPF
-Exclude-Default  VRRP-Extended  VSRP
Configured Static Routes: 15
Maximum allowed import VRFs: 2048
```

device **# show ipv6**

This command will display the maximum IPv6 allowed import VRFs and list of IPv6 import VRFs to default-vrf.

Displaying the IP route table for a specified VRF

To display the IP routes for a specified VRF, enter the following command at any CLI level for IPv4 and IPv6 respectively.

```
device# show ip route vrf one
device# show ipv6 route vrf one
```

Syntax: show ip route vrf vrf-name [*num*] | [*ip-addr*] | [**bgp**] | [**connected**] | [**isis**] | [**ospf**] | [**rip**] | [**static**] | [**tags**]

Syntax: show ipv6 route vrf vrf-name [*num*] | [*ip-addr*] | [**bgp**] | [**connected**] | [**isis**] | [**ospf**] | [**rip**] | [**static**] | [**tags**] | **nexthop** *nexthop_id* | **ref-routes**

The *vrf-name* parameter specifies the VRF for which you want to display the IP routes.

The **nexthop** option displays the next-hop information for all next hops in the routing table or for a specific entry. The *nexthop_id* parameter is a specific nexthop entry from the next hop table.

The **ref-routes** option allows you to display IPv6 routes in the forwarding table that refer to the specified nexthop entry.

The following table lists the information displayed by the **show ip/ipv6 route vrf** command.

TABLE 61 CLI display of IP route-table

This field...	Displays
Total number of IP routes	The total number of IP routes that are in the specified VRP routing-table.
Destination	The destination network of the route.
NetMask	The network mask of the destination address.
Gateway	The next-hop router.
Port	The port through which this Brocade device sends packets to reach the route's destination.
Cost	The route's cost.
Type	The route type, which can be one of the following: • B - The route was learned from BGP. • D - The destination is directly connected to this Brocade device. • R - The route was learned from RIP. • S - The route is a static route. • * - The route is a candidate default route. • O - The route is an OSPF route. Unless you use the **ospf** option to display the route table, "O" is used for all OSPF routes. If you do use the **ospf** option, the following type codes are used: - O - OSPF intra area route (within the same area). - IA - The route is an OSPF inter area route (a route that passes from one area into another). - E1 - The route is an OSPF external type 1 route. - E2 - The route is an OSPF external type 2 route.

TABLE 61 CLI display of IP route-table (Continued)

This field...	Displays
Uptime	The amount of time since the route was last modified. The format of this display parameter may change depending upon the age of the route to include the seconds (s), minutes (m), hours (h), and days (d), as described in the following: 400d - Only days (d) displayed 20d23h - days (d) and hours (h) displayed 14h33m - hours (h) and minutes (m) displayed 10m59s - minutes (m) and seconds (s) displayed
src-vrf	The source VRF from where the routes have been imported to the destination VRF. The "-" in the src-vrf column output denotes the route is local route.

Displaying the IP route table for a specified VRF import, local or summary

To display the IP routes for a specified VRF import or from local VRF or all VRFs summary, enter the following command at any CLI level for IPv4 and IPv6 respectively.

```
device# show ip route vrf one import/local/summary
device# show ipv6 route vrf one import/local/summary
```

Syntax: show ip route vrf [**import** | **local** | **summary**] **vrf-name** [*num*] | [*ip-addr*] | [**bgp**] | [**connected**] | [**isis**] | [**ospf**] | [**rip**] | [**static**] | [**tags**]

Syntax: show ipv6 route vrf [**import** | **local** | **summary**] **vrf-name** [*num*] | [*ip-addr*] | [**bgp**] | [**connected**] | [**isis**] | [**ospf**] | [**rip**] | [**static**] | [**tags**] | **nexthop** *nexthop_id* | **ref-routes**

The *vrf-name* parameter specifies the VRF for which you want to display the IP routes.

The **nexthop** option displays the next-hop information for all next hops in the routing table or for a specific entry. The *nexthop_id* parameter is a specific nexthop entry from the next hop table.

The **ref-routes** option allows you to display IPv6 routes in the forwarding table that refer to the specified nexthop entry.

Displaying the IP route table for a specified VRF on page 371 lists the information displayed by these commands.

Displaying IPv4 routes in VRF one

To display IP information for a specified VRF, enter the following command at any level of the CLI.

```
device(config)#show ip route vrf one
Total number of IP routes: 4
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
BGP  Codes - i:iBGP e:eBGP
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
        Destination        Gateway         Port          Cost          Type Uptime
src-vrf
1       10.0.0.0/8         10.25.104.1     eth1/1          20          O
4d1h       c
2       10.1.0.0/16        10.25.103.1     eth2/1          20          O
```

```
3d1h       b
3       10.20.0.0/8        10.25.104.1     eth1/1          20          O
4d1h       -
4       10.40.0.0/8        10.25.105.1     eth2/1          20          O
3d1h       default
```

Displaying IPv4 routes in VRF one import

To display IP information for a specified VRF import, enter the following command at any level of the CLI.

```
device(config)#show ip route vrf one import
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
BGP  Codes - i:iBGP e:eBGP
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
        Destination        Gateway         Port          Cost          Type Uptime
src-vrf
1       10.0.0.0/8         10.25.104.1     eth1/1          20          O
4d1h       c
2       10.1.0.0/16        10.25.103.1     eth2/1          20          O
3d1h       b
3       10.40.0.0/8        10.25.105.1     eth2/1          20          O
3d1h       default
```

Displaying outputs routed from other VRF

To display all the other VRFs from where the routes will be imported into this VRF, enter the following command at any level of the CLI.

Syntax: show ip vrf *vrf-name*

```
device#show ip vrf one
VRF one, default RD 1001:11, Table ID 2 IFL ID 131070
Label: 500000,  Label-Switched Mode: OFF
IP Router-Id: 10.1.1.2
  Interfaces:
    e2/8
  No Export VPN route-target communities
  No Import VPN route-target communities
  No import route-map
  No export route-map
  Address Family IPv4
    Max Routes: 5120
    Imports routes from VRF: a, b, c, d
    No Export VPN route-target communities
    No Import VPN route-target communities
 Address Family IPv6
    Max Routes: 128
    Imports routes from VRF: a, b
    No Export VPN route-target communities
    No Import VPN route-target communities
```

Configuring routes from multiple VRFs

This example shows the sequence of commands in configuring routes from multiple VRFs.

```
device# vrf a
device# rd 1111:11
device# address-family ipv4
device# import routes vrf b route-map import-map
device# import routes vrf c route-map import-map
device# exit-address-family
device# address-family ipv6
device# import routes vrf b route-map import-v6map
device# exit-address-family
device# exit-vrf
```

```
device# route-map import-map permit 10
device# match ip address prefix-list export
device# route-map import-map permit 15
device# match ip address prefix-list loop
```

NOTE
If the configuration of a route-map is changed, then the VRFs which are configured to use the respective route-map will be processed again.

Displaying IPv6 routes in VRF one from local

To display IP information for a specified VRF for IPv6 routes from the local VRF, enter the following command at any level of the CLI.

```
device(config)#show ipv6 route vrf one local
Type Codes - B:BGP C:Connected I:ISIS L:Local O:OSPF R:RIP S:Static
BGP  Codes - i:iBGP e:eBGP
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2
Destination       Gateway                   Port   Cost   Type   Uptime   src-vrf
   3000:/8  fe80::768e:f8ff:fe2a:d063 eth1/3   20     O      4d1h       -
```

Displaying IPv6 imported routes summary

To display IP information for all VRFs summary for IPv6 routes, enter the following command at any level of the CLI.

```
device(config)#show ipv6 route vrf one import vrf summary
IPv6 Routing Table - 10 entries:
  0 connected, 0 static, 0 RIP, 10 OSPF, 0 BGP, 0 ISIS
  Number of prefixes:
  /64:10
```

NOTE
An error will be displayed when an attempt to match the source VRF name with the import VRF name.

Displaying IPv6 routes in VRF one imported from another VRF

To display IP information for a specified VRF import routes from VRF two, enter the following command at any level of the CLI.

```
device(config)#show ipv6 route vrf one import vrf b
Type Codes - B:BGP C:Connected I:ISIS L:Local O:OSPF R:RIP S:Static
BGP  Codes - i:iBGP e:eBGP
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2
Destination       Gateway                   Port   Cost   Type   Uptime   src-vrf
        2000:/8  fe80::768e:f8ff:fe2a:d062 eth1/2 20     O      4d1h       -
```

Clearing IP routes

You can clear the entire routing-table or specific individual routes as needed.

To clear all routes from the IPv4 routing-table, enter the following command at any level of the CLI.

```
device# clear ip route
```

To clear route 10.157.22.0/24 from the IPv4 routing table, enter:

```
device# clear ip route 10.157.22.0/24
```

Syntax: clear [**ip** | **ipv6**] **route** [*ip-addrip-mask* | *ip-addr/mask-bits*] [**import** | **local** | **import vrf** *vrf-name*] | **nexthop** *nexthop_id*

The following examples illustrate the use of the **clear route** command:

```
device# clear ip route vrf one [<IP address
> <Mask
>] import
```

Clears the imported IPv4 routes from all other VRFs. When this command is issued with an IP address and mask, then the imported routes matching the address and mask from other VRFs are cleared.

```
device# clear ip route vrf one [<IP address
> <Mask
> local]
```

Clears the IPv4 routes from specific VRF. When this command is issued with an IP address and mask only, then the local routes matching the address and mask are cleared, otherwise this option is not available.

```
device# clear ip route vrf one [<IP address
> <Mask
>] import vrf two
```

Clears the imported IPv4 routes from VRF two. When this command is issued with an IP address and mask, then imported routes matching the address and mask from VRF two are cleared.

```
device# clear ip route vrf one [<IP address
> <Mask
>] import vrf default-vrf
```

Clears the imported IPv4 routes from the default-vrf. When this command is issued with an IP address and mask, then imported routes matching the address and mask from the default VRF are cleared.

```
device# clear ipv6 route
```

Clears all routes from the IPv6 routing-table.

```
device# clear ipv6 route 10.157.22.0/24
```

Clears route 10.157.22.0/24 from the IPv6 routing-table.

```
device# clear ipv6 route vrf one [IPv6 addr/Prefix length
] import vrf two
```

Clears the imported IPv6 routes from VRF two. When this command is issued with an IPv6 address and prefix length, then the imported routes matching the IPv6 address and prefix length from VRF two are removed.

```
device# clear ipv6 route vrf one [IPv6 addr/Prefix length
] import vrf default-vrf
```

Clears the imported IPv6 routes from the default-vrf. When this command is issued with an IPv6 address and prefix length, then the imported routes matching the IPv6 address and prefix length from the default VRF are removed.

```
device# clear ipv6 route nexthop nexthop_ID
```

Clears the imported IPv6 routes for the specified nexthop ID on the interface module (LP).

Configuring the number of VRFs for IPv4 and IPv6

To limit the number of imported IPv4 or IPv6 routes into any VRF including the default VRF, the following command is available in the global configuration mode and not available in any individual VRF mode. Changes in the value in the global configuration mode will be effective in all VRFs.

```
device(config)# [ip|ipv6] max-import-vrfs 1-2048
```

The **no** command will set the value to the default value, which is 50.

If you configure **ip max-import-vrfs** to a number which is less than the currently imported routes in the IPv4 or IPv6 address family for any VRF, then the following error will be displayed and the configuration will not be accepted.

```
device(config)# [ip|ipv6
] max-import-vrfs 2
Error: VRF one has 3 import commands configured in ipv4/ipv6 address families
```

The configured non-default value of **ip/ipv6 max-import-vrfs** may be displayed using the **show ip** or **show ip vrf-name** commands.

NOTE
A system maximum of 1000 import commands (including all VRFs, IPv4 and IPv6 address families) can be defined.

NOTE
Using the **system-max ip-vrf-route** command, the number of IPv4 routes per VRF instance is limited to 1024. Using the **system-max ipv6-vrf-route** command, the number of IPv6 routes per VRF instance is limited to 8192.

Modified CLI commands

The Inter-vrf routing feature makes it possible to import OSPF routes from one VRF to another VRF. There may be a need to advertise the imported OSPF routes back to the OSPF domain as external routes. This requires redistribution of OSPF into OSPF again, which was not supported In prior releases. With the introduction of redistribution, the following configuration is supported:

```
device(config-ospf-router)#redistribute ospf route-map <route-map-name>
  bgp         Border Gateway Protocol (BGP)
  connected   Connected
  isis        Intermediate System to Intermediate System (IS-IS)
  rip         Routing Information Protocol (RIP)
  static      Static routes
```

ospf OSPF routes (new addition)

This command is applicable to OSPF, RIP and BGP. Currently IS-IS does not support VRFs and it is not possible to have IS-IS running in multiple VRFs.

If you configure to import the same protocol routes into the same protocol, then RTM will send back protocol routes belonging to other VRFs.

Redistribution of a protocol into itself is supported for the following protocols:

- IPv4

1. OSPF->OSPF
   a) route-map option
2. BGP->BGP
   a) route-map option
   b) Metric option
3. RIP->RIP
   a) route-map option
   b) Metric option

- IPv6

Management VRF

- Management VRF overview.....379
- Configuring a global management VRF.....383
- Displaying the management VRF information.....384

Table 62 displays the individual devices and the management Virtual Routing and Forwarding (VRF) features they support.

TABLE 62 Supported NetIron management Virtual Routing and Forwarding (VRF) features

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
IPv4 management VRF	Yes	Yes	Yes	Yes	Yes	Yes	Yes
IPv6 management VRF	Yes	Yes	No	No	No	No	No

Management VRF overview

The management VRF is used to provide secure management access to the device by sending inbound and outbound management traffic through the VRF specified as a global management VRF and through the out-of-band management port, thereby isolating management traffic from the network data traffic.

By default, the inbound traffic is unaware of VRF and allows incoming packets from any VRF, including the default VRF. The outbound traffic is only through the default VRF. The default VRF consists of out-of-band management port and all the LP ports that do not belong to any other VRFs.

Any VRF, except the default VRF, can be configured as a management VRF. When a management VRF is configured, the management traffic is allowed through the ports belonging to the specified VRF and the out-of-band management port. The management traffic through the ports belonging to the other VRFs and the default VRF are dropped and the rejection statistics are incremented.

If the management VRF is not configured, the management applications will follow the default behavior. The management VRF configuration is applicable for both IPv4 and IPv6 management traffic.

NOTE
The IPv6 management VRF is not supported on Brocade NetIron CES Series and Brocade NetIron CER Series devices.

The management VRF is supported by the following management applications:

- SNMP server
- SNMP trap generator
- Telnet server
- SSH server
- Telnet client
- RADIUS client
- TACACS+ client
- TFTP
- SCP
- Syslog

NOTE
The management VRF is not applicable to inbound and outbound traffic of the **ping** and **traceroute** commands. These commands use the VRF specified in the command or the default VRF, if no VRF is specified.

Source interface and management VRF compatibility

There is a source interface configuration associated with the management applications. When a source interface is configured, the management applications use the lowest configured IP address of the specified interface as source IP address in all the outgoing packets. If the configured interface is not part of the management VRF, the response packet will not reach the destination. If the compatibility check fails while configuring either the management VRF or the source interface, the following warning message will be displayed. However, the configuration command will be accepted.

```
The source-interface for Telnet, TFTP is not part of the management-vrf
```

Supported management applications

This section explains the management VRF support provided by the management applications.

SNMP server

When the management VRF is configured, the SNMP server receives SNMP requests and sends SNMP responses only through the ports belonging to the management VRF and through the out-of-band management port.

Any change in the management VRF configuration becomes immediately effective for the SNMP server.

SNMP trap generator

When the management VRF is configured, the SNMP trap generator sends traps to trap hosts through the ports belonging to the management VRF and through the out-of-band management port.

Any change in the management VRF configuration becomes immediately effective for the SNMP trap generator.

NOTE
The SNMP source interface configuration command **snmp-server trap-source** must be compatible with the management VRF configuration. Refer to Source interface and management VRF compatibility on page 380.

Telnet server

When the management VRF is configured, the incoming Telnet connection requests are allowed only from the ports belonging to the management VRF and from the out-of-band management port. Management VRF enforcement is only done during the establishment of a connection. Once the connection is established, no further management VRF enforcement is done.

To allow the incoming Telnet connection requests only from the management VRF and not from the out-of-band management port, enter the following command.

```
device(config)# telnet strict-management-vrf
```

The previous command is applicable only when the management VRF is configured. If not, the command issues the following warning message.

```
Warning - Management-vrf is not configured.
```

For the Telnet server, changes in the management VRF configuration or configuring the **telnet strict-management-vrf** command will not affect the existing Telnet connections and the changes will be applied only to the new incoming connection requests.

SSH server

When the management VRF is configured, the incoming SSH connection requests are allowed only from the ports belonging to the management VRF and from the out-of-band management port. Management VRF enforcement is only done during the establishment of a connection. Once the connection is established, no further management VRF enforcement is done.

To allow the incoming SSH connection requests only from the management VRF and not from the out-of-band management port, enter the following command.

```
device(config)# ip ssh strict-management-vrf
```

The previous command is applicable only when the management VRF is configured. If not, the command issues the following warning message.

```
Warning - Management-vrf is not configured.
```

For the SSH server, changes in the management VRF configuration or configuring the **ip ssh strict-management-vrf** command will not affect the existing SSH connections and the changes will be applied only to the new incoming connection requests.

Telnet client

When the VRF name is specified in the **telnet vrf** command, the Telnet client initiates Telnet requests only from the ports belonging to the specified VRF.

To configure the VRF name in outbound Telnet sessions, enter the following command at the privileged EXEC level:

```
device(config)# telnet vrf red 10.157.22.39
```

Syntax: telnet vrf *vrf-name IPv4address* | **ipv6** *IPv6address*

The *vrf-name* variable specifies the name of the pre-configured VRF.

NOTE
The IPv6 management VRF is not supported on Brocade NetIron CES Series and Brocade NetIron CER Series devices.

RADIUS client

When the management VRF is configured, the RADIUS client will sends RADIUS requests or receives responses only through the ports belonging to the management VRF and through the out-of-band management port.

Any change in the management VRF configuration will be immediately effective for the RADIUS client.

NOTE
The RADIUS source interface configuration command**ip radius source-interface** must be compatible with the management VRF configuration. Refer to Source interface and management VRF compatibility on page 380.

TACACS+ client

When the management VRF is configured, the TACACS+ client establishes connections with TACACS + servers only through the ports belonging to the management VRF and the out-of-band management port.

For the TACACS+ client, any change in the management VRF configuration will not affect the existing TACACS+ connections and the changes will be applied only to the new TACACS+ connections.

NOTE
The TACACS+ source interface configuration command **ip tacacs source-interface** must be compatible with the management VRF configuration. Refer to Source interface and management VRF compatibility on page 380.

TFTP

When the management VRF is configured, TFTP will send or receive the data and acknowledgements only through the ports belonging to the management VRF and through the out-of-band management port.

Any change in the management VRF configuration will be immediately effective for TFTP. You cannot change in the management VRF configuration while TFTP is in progress.

NOTE
The TFTP source interface configuration command **ip tftp source-interface** must be compatible with the management VRF configuration. Refer to Source interface and management VRF compatibility on page 380.

SCP

SCP uses SSH as underlying transport. The behavior of SCP is similar to the SSH server. For more information, refer to SSH server on page 381.

Syslog

When the management VRF is configured, the Syslog module sends log messages only through the ports belonging to the management VRF and the out-of-band management port.

Any change in the management VRF configuration will be immediately effective for Syslog.

NOTE
The Syslog source interface configuration command **ip syslog source-interface** must be compatible with the management VRF configuration. Refer to Source interface and management VRF compatibility on page 380.

Configuring a global management VRF

To configure a VRF as a global management VRF, enter the following command.

```
device(config)# management-vrf mvrf
```

Syntax: [no] management-vrf *vrf-name*

The *vrf-name* parameter specifies the name of the pre-configured VRF. If the VRF is not pre-configured, the command execution fails and displays the following error message.

```
Error - VRF <vrf-name>
 doesn't exist
```

When the management VRF is configured, the software generates the following Syslog message.

```
SYSLOG: VRF <vrf-name>
 has been configured as management-vrf
```

Enter the **no** form of the command to remove the management VRF. When the management VRF is deleted, the software generates the following Syslog message.

```
SYSLOG: VRF <vrf-name>
 has been un-configured as management-vrf
```

Configuration notes

Consider the following configuration notes:

- If there is a management VRF already configured, you must remove the existing management VRF configuration before configuring a new one. If not, the system displays the following error message.

```
device(config)# management-vrf  red
Error - VRF mvrf already configured as management-vrf
```

- If you try to delete a management VRF that was not configured, the system displays the following error message.

```
device(config)# no management-vrf red
Error - VRF red is not the current management-vrf
```

- The deletion or modification of the VRF will fail if the specified VRF is currently configured as the management VRF. Attempting to do so causes the system to return the following error message.

```
device(config)# no vrf mvrf
Error - Cannot modify/delete a VRF which is configured as management-vrf
```

Displaying the management VRF information

To display IP Information for a specified VRF, enter the following command at any level of the CLI.

```
device(config)# show vrf mvrf
Total number of VRFs configured: 1
Status Codes - A:active, D:pending deletion, I:inactive
Name                 Default RD                                        IFL ID  vrf|
v4|v6                Routes Interfaces
a                              1:1                          131071   A | A|
A                                    14
Total number of IPv4 unicast route for all non-default VRF is 12
Total number of IPv6 unicast route for all non-default VRF is 2
device#show vrf a
VRF a, default RD 1:1, Table ID 1 IFL ID 131071
Label: (Not Allocated),  Label-Switched Mode: OFF
Configured as management-vrf
IP Router-Id: 10.2.2.2
No interfaces
  No Export VPN route-target communities
  No Import VPN route-target communities
  No import route-map
  No export route-map
  Address Family IPv4
    Max Routes: 5120
    Number of Unicast Routes: 12
    No Export VPN route-target communities
    No Import VPN route-target communities
Address Family IPv6
    Max Routes: 128
    Number of Unicast Routes: 2
    No Export VPN route-target communities
    No Import VPN route-target communities
```

Syntax: show vrf *vrf-name*

The *vrf-name* parameter specifies the VRF for which you want to display IP information.

Table 63 displays a description of the output from the **show vrf** command.

TABLE 63 Output from the **show vrf** command

This field...	Displays...
VRF *vrf-name*	The name of the VRF.
default RD	The default route distinguisher for the VRF.
Table ID	The table ID for the VRF.
Routes	The total number of IPv4 and IPv6 Unicast routes configured on this VRF.
IFL ID	The Internal Forwarding Lookup Identifier (IFL-ID) for ports in the VRF instance.
Label	The unique VRF label that has been assigned to the specified VRF.
Label-Switched Mode	Indicates whether Label-Switched Mode is ON or OFF.
Configured as management-vrf	Indicates that the specified VRF is configured as a management VRF.
IP Router-Id	The 32-bit number that uniquely identifies the router.
Number of Unicast Routes	The number of Unicast routes configured on this VRF.
import route-map	The name of the import route-map, if any, that is configured for this management VRF.
export route-map	The name of the export route-map if a route-map has been configured for this management VRF.

The `show who` command displays information about the management VRF from which the Telnet and SSH connection has been established.

```
device(config)# show who
Console connections:
        established, monitor enabled, privilege super-user, in config mode
        1 minutes 47 seconds in idle
Telnet server status: Enabled
Telnet connections (inbound):
 1      established, client ip address 10.53.1.181, user is lab, privilege super-user
        using vrf default-vrf.
        2 minutes 46 seconds in idle
 2      established, client ip address 10.20.20.2, user is lab, privilege super-user
        using vrf mvrf.
        16 seconds in idle
 3      closed
 4      closed
 5      closed
Telnet connections (outbound):
 6      established, server ip address 10.20.20.2, from Telnet session 2, , privilege
super-user
        using vrf mvrf.
        12 seconds in idle
 7      closed
 8      closed
 9      closed
 10     closed
SSH server status: Enabled
SSH connections:
```

```
1       established, client ip address 10.53.1.181, privilege super-user
        using vrf default-vrf.
        you are connecting to this session
        3 seconds in idle
2       established, client ip address 10.20.20.2, privilege super-user
        using vrf mvrf.
        48 seconds in idle
3       closed
4       closed
5       closed
6       closed
7       closed
8       closed
9       closed
10      closed
11      closed
12      closed
13      closed
14      closed
15      closed
16      closed
```

Syntax: show who

To display the packets and sessions rejection statistics due to failure in management VRF validation, enter the following command.

```
device(config)# show management-vrf
Management VRF name : mvrf
Management Application Rx Drop Pkts Tx Drop Pkts
SNMP Engine                              36             0
RADIUS Client                             0             8
TFTP Client                               0             4
SNMP Notifications                                -             55
SysLogs                             -             78
TCP Connection rejects:
Telnet                    : 1
SSH                  : 1
TACACS+ Client                    : 8
```

Syntax: show management-vrf

Table 64 displays a description of the output from the **show management-vrf** command.

TABLE 64 Output from the **show management-vrf** command

This field...	Displays...
Management VRF name	Displays the configured management VRF name.
Management Application	Displays the management application names.
Rx Drop Pkts	Displays the number of packets dropped in the inbound traffic.
Tx Drop Pkts	Displays the number of packets dropped in the outbound traffic.
TCP Connection rejects	Displays the number of TCP connections per application rejected due to management VRF validation.

Make sure that the management VRF is configured before executing the **show management-vrf** command. If not, the system will display the following error message.

```
Error - Management VRF is not configured.
```

To clear the management VRF rejection statistics, enter the following command.

```
device(config)# clear management-vrf-stats
```

Syntax: clear management-vrf-stats

Basic IPv6 Connectivity

- Enabling IPv6 routing....390
- Configuring IPv6 on each interface....391
- Configuring the management port for an IPv6 automatic address configuration....394
- IPv6 host support....395
- IPv6 Non stop routing and graceful restart....399
- Configuring IPv4 and IPv6 protocol stacks....407
- Configuring IPv6 Domain Name Server (DNS) resolver....408
- IPv6 Non-Stop Routing support....409
- ECMP load sharing for IPv6....410
- DHCP relay agent for IPv6....411
- Configuring IPv6 ICMP....419
- Configuring IPv6 neighbor discovery....423
- IPv6 source routing security enhancements....430
- Changing the IPv6 MTU....434
- Configuring static neighbor entries....435
- Limiting the number of hops an IPv6 packet can traverse....436
- QoS for IPv6 traffic....436
- Configuring the rate limit for IPv6 subnet traffic....437
- Clearing global IPv6 information....437
- Displaying global IPv6 information....439
- Commands....461

Table 65 displays the individual Brocade devices and the basic IPv6 Connectivity features they support.

TABLE 65 Supported basic IPv6 connectivity features

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
IPv6 Routing	Yes	Yes	No	Yes	Yes	Yes	Yes
IPv6 Anycast Addresses	Yes	Yes	No	No	No	No	No
IPv6 Host Support	Yes	Yes	No	Yes	Yes	Yes	Yes
IPv6 Non stop routing and graceful restart	Yes	Yes	Yes	No	No	No	No

TABLE 65 Supported basic IPv6 connectivity features (Continued)

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
Restricting SNMP Access to an IPv6 Node	Yes	Yes	No	Yes	Yes	Yes	Yes
IPv6 Domain Name Server (DNS) Resolver	Yes	Yes	No	Yes	Yes	Yes	Yes
ECMP Load Sharing for IPv6	Yes	Yes	No	Yes	Yes	Yes	Yes
IPv6 ICMP	Yes	Yes	No	Yes	Yes	Yes	Yes
DHCPv6	Yes	Yes	No	Yes	Yes	Yes	Yes
IPv6 Neighbor Discovery	Yes	Yes	No	Yes	Yes	Yes	Yes
IPv6 Source Routing Security	Yes	Yes	No	Yes	Yes	Yes	Yes
IPv6 MTU	Yes	Yes	No	Yes	Yes	Yes	Yes
Static Neighbor Entries	Yes	Yes	No	Yes	Yes	Yes	Yes
Limiting the Number of Hops an IPv6 Packet Can Traverse	Yes	Yes	No	Yes	Yes	Yes	Yes
QoS for IPv6 Traffic	Yes	Yes	Yes	Yes	Yes	Yes	Yes
IPv6 Non-Stop Routing (NSR) Support	Yes	Yes	No	No	No	No	No
DNS Queries of IPv4 and IPv6 DNS Servers	Yes	Yes	No	Yes	Yes	Yes	Yes
Displaying IPv6 Traffic Statistics	Yes	Yes	No	No	No	No	No
Displaying the IPv6 Route Table	Yes	Yes	No	Yes	Yes	Yes	Yes

TABLE 65 Supported basic IPv6 connectivity features (Continued)

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
Rate Limit for IPv6 Subnet Traffic	Yes	Yes	No	No	No	No	No
DHCPv6 Relay Agent Prefix Delegation Notification	Yes	Yes	No	Yes	Yes	Yes	Yes
IPv6 Router Advertisement Options for DNS Configuration	Yes	Yes	No	Yes	Yes	Yes	Yes
127-bit IPv6 address	Yes	Yes	Yes	Yes	Yes	Yes	Yes

This chapter explains how to get a Brocade device up and running with IPv6. To configure basic IPv6 connectivity, you must do the following:

- Enable IPv6 routing globally.
- Configure an IPv6 address or explicitly enable IPv6 on each Brocade device interface over which you plan to forward IPv6 traffic.
- Configure IPv4 and IPv6 protocol stacks. (This step is mandatory only if you want a Brocade device interface to send and receive both IPv4 and IPv6 traffic.)

The following configuration tasks are optional:

- Configure IPv6 Domain Name Server (DNS) resolver.
- Configure equal-cost mulitpath (ECMP) routing Load Sharing for IPv6.
- Configure IPv6 Internet Control Message Protocol (ICMP).
- Configure the IPv6 neighbor discovery feature.
- Change the IPv6 maximum transmission unit (MTU).
- Configure static neighbor entries.
- Limit the hop count of an IPv6 packet.
- Configure Quality of Service (QoS) for IPv6 traffic.

Enabling IPv6 routing

By default, IPv6 routing is enabled. If forwarding of IPv6 traffic globally on the device has been disabled, you can enable it by entering the following command.

```
device(config)# ipv6 unicast-routing
```

Syntax: [no] ipv6 unicast-routing

To disable the forwarding of IPv6 traffic globally on the device, enter the **no** form of this command.

NOTE
Downgrading from release 04.1.00 to an earlier release of the software can impact IPv6 routing. In earlier versions of the Multi-Service IronWare software, IPv6 routing was disabled by default. As of release 04.1.00, IPv6 routing is enabled by default and therefore does not appear in the configuration. If you are downgrading from 04.1.00 to an earlier version of the software and want IPv6 routing to be enabled, you must add the line "ipv6 unicast-routing" to the configuration.

Configuring IPv6 on each interface

To forward IPv6 traffic on an interface, the interface must have an IPv6 address, or IPv6 must be explicitly enabled. By default, an IPv6 address is not configured on an interface.

If you choose to configure a global or unique local IPv6 unicast address (ULA) for an interface, IPv6 is also enabled on the interface. Further, when you configure a global or unique local IPv6 unicast address, you must decide on one of the following in the low-order 64 bits:

- A manually configured interface ID.
- An automatically computed EUI-64 interface ID.

If you prefer to assign a link-local IPv6 address to the interface, you must explicitly enable IPv6, which causes a link-local address to be automatically computed for the interface. If preferred, you can override the automatically configured link-local address with an address that you manually configure.

This section provides the following information:

- Configuring a global or unique local IPv6 unicast address with a manually configured or automatically computed interface ID for an interface.
- Automatically or manually configuring a link-local address for an interface.
- Configuring IPv6 anycast addresses

NOTE
On Brocade NetIron XMR Series and Brocade NetIron MLX Series devices, the IPv6 packet received with the DA MAC as the router's MAC is subjected to an IPv6 route lookup irrespective of IPv6 routing enabled on the interface.

Configuring a global or unique local IPv6 unicast address

Configuring a global or unique local IPv6 unicast address on an interface does the following:

- Automatically configures an interface ID (a link-local address), if specified.
- Enables IPv6 on that interface.

Additionally, the configured interface automatically joins the following required multicast groups for that link:

- Solicited-node multicast group FF02:0:0:0:0:1:FF00::/104 for each unicast address assigned to the interface.
- All-nodes link-local multicast group FF02::1
- All-routers link-local multicast group FF02::2

The neighbor discovery feature sends messages to these multicast groups. For more information, refer to Configuring IPv6 neighbor discovery on page 423.

Configuring a global or unique loal IPv6 unicast address with a manually configured interface ID

To configure a global or unique local IPv6 unicast address, including a manually configured interface ID, for an interface, enter commands such as the following.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 address 2001:DB8:12D:1300:240:D0FF:
FE48:4672/64
```

These commands configure the global prefix 2001:DB8:12d:1300::/64 and the interface ID:: 240:D0FF:FE48:4672, and enable IPv6 on Ethernet interface 3/1.

Syntax: ipv6 address *ipv6-prefix/prefix-length*

You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

Configuring a global or unique local IPv6 unicast address with an automatically computed EUI-64 interface ID

To configure a global or unique local IPv6 unicast address with an automatically computed EUI-64 interface ID in the low-order 64-bits, enter commands such as the following.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 address 2001:DB8:12D:1300::/64 eui-64
```

These commands configure the global prefix 2001:DB8:12d:1300::/64 and an interface ID, and enable IPv6 on Ethernet interface 3/1.

Syntax: [no] ipv6 address *ipv6-prefix/prefix-length* **eui-64**

You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The **eui-64** keyword configures the global or unique local unicast address with an EUI-64 interface ID in the low-order 64 bits.The interface ID is automatically constructed in IEEE EUI-64 format using the interface's MAC address.

Configuring a link-local IPv6 address

To explicitly enable IPv6 on an interface without configuring a global or unique local unicast address for the interface, enter commands such as the following.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 enable
```

These commands enable IPv6 on Ethernet interface 3/1 and specify that the interface is assigned an automatically computed link-local address.

Syntax: [no] ipv6 enable

NOTE
When configuring VLANs that share a common tagged interface with a Virtual Ethernet (VE) interface, it is recommended that you override the automatically computed link-local address with a manually configured unique address for the interface. If the interface uses the automatically computed address, which in the case of VE interfaces is derived from a global MAC address, all VE interfaces will have the same MAC address.

To override a link-local address that is automatically computed for an interface with a manually configured address, enter commands such as the following.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 address FE80::240:D0FF:FE48:4672 link-local
```

These commands explicitly configure the link-local address FE80::240:D0FF:FE48:4672 for Ethernet interface 3/1.

Syntax: [no] ipv6 address *ipv6-address* **link-local**

You must specify the *ipv6-address* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **link-local** keyword indicates that the Brocade device interface should use the manually configured link-local address instead of the automatically computed link-local address.

Configuring IPv6 anycast addresses

In IPv6, an **anycast** address is an address for a set of interfaces that belong to different nodes. Sending a packet to an anycast address results in the delivery of the packet to the closest interface that has an anycast address.

An anycast address looks similar to a unicast address, because it is allocated from the unicast address space. If you assign an IPv6 unicast address to multiple interfaces, it is an anycast address. On the device, you configure an interface assigned an anycast address to recognize the address as an anycast address.

For example, the following commands configure an anycast address on interface 2/1.

```
device(config)# int e 2/1
device(config-if-e100-2/1)# ipv6 address 2001:db8::6/64 anycast
```

Syntax: [no] ipv6 address *ipv6-prefix* | *prefix-length* [**anycast**]

IPv6 anycast addresses are described in detail in RFC 1884. See RFC 2461 for a description of how the IPv6 Neighbor Discovery mechanism handles anycast addresses.

IPv6 anycast filtering

By default all IPv6 packets with anycast address as destination will be processed. The following command provides options to selectively enable protocols or disable all protocols.

```
Brocade(config)# ipv6 anycast-no-response allow tcp
```

Syntax:[no] ipv6 anycast-no-response [allow tcp|udp|icmp]

The **allow tcp | udp | icmp** specifies the protocol to allow for processing.

NOTE

1. The **allow** options can also be used as standalone commands. If **ipv6 anycast-no-response** is already configured, it is modified based on the specified filters.
2. User can enable generation of TCP resets for incoming TCP packets with destination set as anycast address, by configuring **ip tcp enable-reset** command. However, if **ipv6 anycast-no-response** command is also enabled, this command becomes void since all anycast packets are blocked. If user requires reset to be sent for incoming anycast TCP packets, user has to configure **ipv6 anycast-no-response allow tcp** to unblock the incoming TCP packets.

Configuring IPv6 127 bit mask address

With 127 bit mask we will have 127 bits in the network part of the address, and 1 bit in the host part of the address. With 1 bit in the host part, we can have only two IPv6 addresses, one for each host. With 127 bit mask we consider 0 and 1 as host address and eliminates subnet-anycast for the configured network from that link.

NOTE
The 127 bit mask address supports only inter-router Point-to-Point links.

Benefits of using 127 bit mask:

- Eliminates the Ping-pong issue
- Reduces the impact of Denial of Service (DOS) attacks
- Saves IPv6 address space

For example, the following commands configure an 127 bit mask IPv6 address:

```
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)#enable
device(config-if-e10000-1/1)#vrf forwarding green
device(config-if-e10000-1/1)#ipv6 address 10:1:1::1/127
device(config-if-e10000-1/1)#ipv6 enable
device(config-if-e10000-1/1)#ipv6 ospf area 0
device(config)# interface ethernet 1/2
device(config-if-e10000-1/2)#enable
device(config-if-e10000-1/2)#ip address 8.8.8.1/24
device(config-if-e10000-1/2)#ipv6 address 1:1:1:1:1::/127
device(config-if-e10000-1/2)#ipv6 enable
```

Configuring the management port for an IPv6 automatic address configuration

You can configure the management port to automatically obtain an IPv6 address. The process is the same for all ports and is described in detail in the Configuring a global or unique local IPv6 unicast address with an automatically computed EUI-64 interface ID on page 392

IPv6 host support

You can configure the device to be an IPv6 host. An IPv6 host has interfaces with IPv6 addresses, but does not have IPv6 routing enabled.

This section lists supported and unsupported IPv6 host features.

IPv6 host supported features

The following IPv6 host features are supported:

- Automatic address configuration

NOTE
Automatic IPv6 address configuration is supported, however, automatic configuration of an IPv6 *global* address is supported only if there is an IPv6 router present on the network. Manual IPv6 address configuration is not supported.

- HTTP/HTTPS over IPv6
- IPv6 ping
- Telnet using an IPv6 address
- TFTP using an IPv6 address
- Trace route using an IPv6 address
- Name to IPv6 address resolution using IPv6 DNS Server
- IPv6 access lists
- IPv6 debugging
- SSH version 1 over IPv6
- SNMP over IPv6
- Logging (Syslog) over IPv6
- MLD version 1 and version 2

See Basic IPv6 Connectivity on page 388 for additional support information

Restricting SNMP access to an IPv6 node

You can restrict SNMP access (which includes Brocade Network Advisor access) to a specified IPv6 host. Enter a command such as the following.

```
device(config)# snmp-client ipv6 2001:DB8:efff:89::23
```

Syntax: [no] snmp-client ipv6 *ipv6-address*

The *ipv6-address* must be in hexadecimal format using 16-bit values between colons, as documented in RFC 2373.

NOTE
You cannot use the following IPv6 addresses with the **snmp-client ipv6** *ipv6-address* command: :: (unspecified address), ff02::01 (all nodes address), and ff02:02 (all routers address)).

Specifying an IPv6 SNMP trap receiver

You can specify an IPv6 host to be a trap receiver so that all SNMP traps are sent to the same SNMP trap receiver or set of receivers, typically one or more host devices on the network. Enter a command such as the following.

```
device(config)# snmp-server host ipv6 2001:DB8:89::13
```

Syntax: [no] snmp-server host ipv6 *ipv6-address*

The *ipv6-address* must be in hexadecimal format using 16-bit values between colons, as documented in RFC 2373.

Restricting Telnet access by specifying an IPv6 ACL

You can specify an IPv6 ACL to restrict Telnet access to management functions on the device. Enter commands similar to the following.

```
(config)# ipv6 access-list acl1
(config-ipv6-access-list acl1)# deny ipv6 host 2000:2382::e0bb:2 any
(config-ipv6-access-list acl1)# deny ipv6 2001:DB8::ff89/128 any
(config-ipv6-access-list acl1)# permit ipv6 any any
(config-ipv6-access-list acl1)# exit
(config)# telnet access-group ipv6 acl1
```

This example configures and applies an IPv6 ACL named "acl1", which denies Telnet access to the device from the specified IPv6 addresses, but allows access from any other IPv6 address.

```
(config)# ipv6 access-list acl2
(config-ipv6-access-list acl2)# permit ipv6 host 2000:2382::e0bb:2 any
(config-ipv6-access-list acl2)# deny ipv6 any any
(config-ipv6-access-list acl2)# exit
```

This example configures and applies an IPv6 ACL named "acl2", which allows Telnet access to the device only from the specified IPv6 address, and denies access from any other IPv6 address.

Syntax: telnet access-group ipv6 *ipv6-acl-name*

The *ipv6-acl-name* is a valid IPv6 ACL.

Restricting SSH access by specifying an IPv6 ACL

You can configure an IPv6 ACL to restrict SSH access to management functions on the device. Enter commands such as the following.

```
(config)# ipv6 access-list acl1
(config-ipv6-access-list acl1)# deny ipv6 host 2000:2382::e0bb:2 any
(config-ipv6-access-list acl1)# deny ipv6 2001:DB8::ff89/128 any
(config-ipv6-access-list acl1)# permit ipv6 any any
(config-ipv6-access-list acl1)# exit
(config)# ssh access-group ipv6 acl1
```

This example configures and applies an IPv6 ACL named "acl1", which denies SSH access to the device from the specified IPv6 addresses, but allows access from any other IPv6 address.

```
(config)# ipv6 access-list acl2
(config-ipv6-access-list acl2)# permit ipv6 host 2000:2382::e0bb:2 any
(config-ipv6-access-list acl2)# deny ipv6 any any
(config-ipv6-access-list acl2)# exit
(config)# ssh access-group ipv6 acl2
```

This example configures and applies an IPv6 ACL named "acl2", which allows SSH access to the device only from the specified IPv6 address, and denies access from any other IPv6 address.

Syntax: [no] ssh access-group ipv6 *ipv6-acl-name*

The *ipv6-acl-name* is a valid IPv6 ACL.

Restricting Web management access by specifying an IPv6 ACL

You can configure an IPv6 ACL to restrict Web management access to management functions on the device. Enter commands such as the following.

```
(config)# ipv6 access-list acl1
(config-ipv6-access-list acl1)# deny ipv6 host 2000:2382::e0bb:2 any
(config-ipv6-access-list acl1)# deny ipv6 2001:DB8::ff89/128 any
(config-ipv6-access-list acl1)# permit ipv6 any any
(config-ipv6-access-list acl1)# exit
(config)# web access-group ipv6 acl1
```

This example configures and applies an IPv6 ACL named "acl1", which denies Web management access to the device from the specified IPv6 addresses, but allows access from any other IPv6 address.

```
(config)# ipv6 access-list acl2
(config-ipv6-access-list acl2)# permit ipv6 host 2000:2382::e0bb:2 any
(config-ipv6-access-list acl2)# deny ipv6 any any
(config-ipv6-access-list acl2)# exit
```

This example configures and applies an IPv6 ACL named "acl2", which allows Web management access to the device only from the specified IPv6 address, and denies access from any other IPv6 address.

Syntax: web access-group ipv6 *ipv6-acl-name*

The *ipv6-acl-name* variable is a valid IPv6 ACL.

Restricting SNMP access by specifying an IPv6 ACL

You can configure an IPv6 ACL to restrict Web management access to management functions on the device.

NOTE
The syntax for configuring ACLs for SNMP access differs from the syntax for controlling Telnet, SSH, and Web management access using ACLs.

```
device(config)# ipv6 access-list aclro
device(config-ipv6-access-list aclro)# deny ipv6 host 2000:2382::e0bb:2 any
device(config-ipv6-access-list aclro)# deny ipv6 2001:DB8::ff89/128 any
device(config-ipv6-access-list aclro)# permit ipv6 any any
device(config-ipv6-access-list aclro)# exit
device(config)# ipv6 access-list aclrw
device(config-ipv6-access-list aclrw)# permit ipv6 host 2000:2382::e0bb:2 any
device(config-ipv6-access-list aclrw)# deny ipv6 any any
device(config-ipv6-access-list aclrw)# exit
device(config)# snmp-server community public ro ipv6 aclro
device(config)# snmp-server community private rw ipv6 aclrw
device(config)# write memory
```

These commands configure IPv6 ACLs *aclro* and *aclrw*, then apply these ACLs to community strings. ACL *aclro* controls read-only access using the "public" community string. ACL *aclrw* controls read-write access using the "private" community string.

Syntax: [no] snmp-server community *string* { **ro** | **rw** } **ipv6** *ipv6-acl-name*

The *string* specifies the SNMP community string you must enter for SNMP access.

The **ro** parameter indicates that the community string is for read-only ("get") access. The **rw** parameter indicates the community string is for read-write ("set") access.

The **ipv6** parameter indicates that you are applying an IPv6 access list.

The *ipv6-acl-name* variable specifies the IPv6 access list name.

NOTE
When **snmp-server community** is configured, all incoming SNMP packets are validated first by their community strings and then by their bound ACLs. Packets are permitted if no filters are configured for an ACL.

Restricting Web management access to your device to a specific IPv6 host

You can restrict Web management access to your device to a specific IPv6 host only. Enter commands such as the following.

```
device(config)# web client ipv6 2001:db8:e0bb::2
```

Syntax: [no] web client ipv6 *ipv6-address*

The *ipv6-address* must be in hexadecimal format using 16-bit values between colons, as documented in RFC 2373.

Specifying an IPv6 Syslog server

To specify an IPv6 Syslog server, enter a command such as the following.

```
device(config)# log host ipv6 2001:db8:e0bb::4
```

Syntax: [no] log host ipv6 ipv6-address [**udp-port-num**]

The *ipv6-address* must be in hexadecimal using 16-bit values between colons, as documented in RFC 2373.

The *udp-port-num* optional parameter specifies the UDP application port used for the Syslog facility.

Viewing IPv6 SNMP server addresses

Many **show** commands display IPv6 addresses for IPv6 SNMP servers. This example shows output for the **show snmp server** command.

```
device# show snmp server

      Contact:
     Location:
Community(ro): .....

Traps
              Warm/Cold start: Enable
                     Link up: Enable
                   Link down: Enable
                Authentication: Enable
     Locked address violation: Enable
         Power supply failure: Enable
```

```
                Fan failure: Enable
          Temperature warning: Enable
                STP new root: Enable
          STP topology change: Enable
                         vsrp: Enable

 Total Trap-Receiver Entries: 4

Trap-Receiver IP-Address                Port-Number Community

     1        10.147.201.100
               162     .....

     2        2001:db8:4000::200
            162     .....

     3        10.147.202.100
               162     .....

     4        2001:db8:3000::200
            162     .....
```

Disabling router advertisement and solicitation messages

Router advertisement and solicitation messages enable a device to discover other devices on the same link. By default, router advertisement and solicitation message generation is enabled. To disable this feature, configure an IPv6 access list that denies them. Enter commands such as the following.

```
device(config)# ipv6 access-list rtradvert
device(config-ipv6-access-list rtradvert)# deny icmp any any router-advertisement
device(config-ipv6-access-list rtradvert)# deny icmp any any router-solicitation
device(config-ipv6-access-list rtradvert)# permit ipv6 any any
```

IPv6 Non stop routing and graceful restart

At times, routers may need to restart or may undergo failover. Traditionally during a restart or failover, sessions with the restarting devices are tore down and re-established. Traffic is disrupted due to route deletion and addition in the forwarding plane. Graceful Restart (GR) and Non Stop Routing (NSR) are two different mechanisms to prevent routing protocol re-convergence during a processor switchover.

When Graceful Restart is used, peer networking devices are informed, via protocol extensions that the router is undergoing a restart condition. Peer devices, known as "helper" devices, will continue to forward to the restarting router until a "grace period", within which the adjacency is re-established.

When Non Stop Routing is used, peer networking devices have no knowledge of any event on the router that is switching over. All information needed to continue the routing protocol peering state is transferred to the standby processor so it can continue immediately upon a switchover. Since NSR does not require the help of neighboring routers during restart, NSR capable routers can be deployed independently in an existing network.

Limitations

- Configuration events that occur at the same time as the switchover may get lost are lost due to the CLI synchronization.
- Neighbor, interface, or NSSA translation state changes 'close' to and during the switchover will not be handled.

  - Due to the core-reset of the LP, dead-timers below 40 seconds are not supported.
  - Number of neighbors supported may be limited depending on how many packets LP can send upon completion of the core-reset, due to competition with LP-sync-updates to get OSPF neighbor packets sent out.
- Traffic counters will not be synced. Neighbor and LSA DB counters will be recalculated on Standby during sync.
- There may be a slowdown of LSA acking due to the wait for the ack from Standby before acking the received LSAs.
- OSPF Database Overflow condition for External LSAs - depending on the sequence of redistribution or new LSAs (from neighbors), the LSAs accepted within the limits of the database may change upon switchover.
- The NSR hitless failover event may not be completely transparent to the network as after switchover additional flooding related protocol traffic will be generated to the directly connected neighbors.
- OSPF Startup Timers will not be applied upon NSR switchover.

Supported protocols

The following protocols support both failover and Hitless Operating system Switchover (HLOS) for each protocol.

TABLE 66 IPv6 Supported protocols for non-stop routing and graceful restart

Protocol	Mechanism
OSPFv3	Non-stop routing, Graceful restart helper
IS-IS IPv6	Non-stop routing
BGP IPv6	Graceful restart

Restart global timers

Restart contains two global timers, **max-hold-timer** and the **protocols-converge-timer,** that:

- Limit the amount of time used for re-syncing routes between the backup Management module and Interface modules (LPs) within the same chassis
- Allow a buffer time for protocols to converge and solve dependencies among each other

If the protocol-based restart features are configured when a Management module (MP) performs a switchover to its backup, routes are maintained on the LPs through the protocol-based restart processes for a specified period of time while the new MP learns the network routes. Once the MP learns all of its routes, the routes from the MP are synced with the routes on the LPs.

Graceful-restart IPv6 max-hold-timer

The **graceful-restart ipv6 max-hold-timer** command defines the time that a Brocade device waits before sync up forwarding information is sent to the LP.

Use the **graceful-restart ipv6 max-hold-timer** command to set the max-hold-timer value.

```
device(config)# graceful-restart ipv6 max-hold-timer 300
```

Syntax: [no] graceful-restart ipv6 max-hold-timer *hold-interval*

The *hold-time* variable is the maximum hold time in seconds before sync up forwarding information is sent to the LP. The acceptable range is 30 to 3600 seconds. The default is 300 seconds.

Graceful-restart IPv6 protocols-converge-timer

The **graceful-restart ipv6 protocols-converge-timer** command defines the time that a Brocade device waits for restarting protocols to converge at the final step in the restart process. In a heavily loaded system where BGP/OSPF/GRE/Static protocols can have a dependency on each other, their restart procedures may also depend on each other. This timer allows protocols to solve inter-dependencies after individual restart processes and before routing modules sync up new forwarding information to the interface module. The default value of 5 seconds will work in most cases, but if a system is heavily loaded and has protocols that depend on each other, it is recommended to increase this value.

Use the **graceful-restart ipv6 protocols-converge-timer** command to set the timer value.

```
device(config)# graceful-restart ipv6 protocols-converge-timer 20
```

Syntax: [no] graceful-restart ipv6 protocols-converge-timer convergence-interval

The *hold-time* variable is the maximum hold time in seconds before management routing modules sync up new forwarding information to interface modules during restart. The range of permissible values is 0 to 1200 seconds. The default value is 5 seconds.

Configuring NSR and graceful restart on OSPFv3

OSPFv3 supports nonstop routing and graceful-restart helper mode. Nonstop routing and graceful-restart helper mode can be configured both in legacy router mode or VRF mode. The following commands are used to configure NSR and graceful-restart helper mode.

```
device(config-ospf6-router)#nononstop-routing
```

Syntax: [no] nonstop-routing

The**nonstop routing** command enables nonstop routing in OSPFv3.

NSR OSPFv3 is only supported on Brocade NetIron MLX Series and Brocade NetIron XMR Series devices. Graceful restart helper mode is supported on Brocade NetIron MLX Series and Brocade NetIron XMR Series devices and Brocade NetIron CER Series and Brocade NetIron CES Series devices.

Use the**graceful-restart helper** command to configure or disable helper mode.

```
device(config-ospf6-router)#graceful-restart helper
```

Syntax: [no] graceful-restart helper [**disable** | **strict-lsa-checking**]

The **graceful-restart helper disable** command disables the graceful-restart helper capability. By default it is enabled.

The**strict-las-checking** command exits helper mode upon a change in topology during a graceful restart.

Show commands

Show running-configuration

This command shows the running configuration.

```
device#show running-config
...
ip router-id 10.1.1.1
!
ipv6 router ospf
 area 0
 nonstop-routing
!
...
!
ipv6 router ospf vrf red
 graceful-restart helper strict-lsa-checking
!
...
ipv6 router ospf vrf blue
 area 0
 graceful-restart helper disable
!
```

Syntax: show running-config

Show ipv6 ospf

This command shows the IPv6 OSPF configuration.

```
device#show ipv6 ospf
OSPFv3 Process number 0 with Router ID 0x10010101(10.1.1.1)
 Running 0 days 3 hours 11 minutes 42 seconds
 Number of AS scoped LSAs is 9
 Sum of AS scoped LSAs Checksum is 00006cc6
 External LSA Limit is 250000
 Route calculation executed 1 times
 Pending outgoing LSA count 0
 Authentication key rollover interval 300 seconds
 Number of areas in this router is 1
 High Priority Message Queue Full count: 0
 BFD is disabled
 Graceful restart helper is enabled, strict lsa checking is disabled
 Nonstop Routing is enabled
```

Syntax: show ipv6 ospf

Show ipv6 ospf vrf *vrf name*

This command shows the IPv6 OSPF configuration on a specific VRF.

```
device#show ipv6 ospf vrf red
 OSPFv3 Process number 0 with Router ID 0x10020202(10.2.2.2)
 Running 0 days 8 hours 32 minutes 14 seconds
 Number of AS scoped LSAs is 4
 Sum of AS scoped LSAs Checksum is 00007d93
 External LSA Limit is 250000
 Route calculation executed 1 times
 Pending outgoing LSA count 0
 Authentication key rollover interval 300 seconds
 Number of areas in this router is 1
 High Priority Message Queue Full count: 0
 BFD is disabled
 Graceful restart helper is enabled, strict lsa checking is enabled
 Nonstop Routing is disabled
device#show ipv6 ospf vrf blue
 OSPFv3 Process number 0 with Router ID 0x10020202(10.2.2.2)
 Running 0 days 8 hours 32 minutes 14 seconds
 Number of AS scoped LSAs is 4
 Sum of AS scoped LSAs Checksum is 00007d93
 External LSA Limit is 250000
```

```
 Route calculation executed 1 times
 Pending outgoing LSA count 0
 Authentication key rollover interval 300 seconds
 Number of areas in this router is 1
 High Priority Message Queue Full count: 0
 BFD is disabled
 Graceful restart helper is disabled, strict lsa checking is disabled
 Nonstop Routing is disabled
```

Syntax: show ipv6 ospf vrf *vrfname*

Show ipv6 ospf database

```
This command shows the IPv6 OSPF database configuration.
device#show ipv6 ospf database
 LSA Key - Rtr:Router Net:Network Inap:InterPrefix Inar:InterRouter
          Extn:ASExternal Grp:GroupMembership Typ7:Type7 Link:Link
          Iap:IntraPrefix Grc:Grace
Area ID         Type LSID       Adv Rtr         Seq(Hex) Age  Cksum Len   Sync
0               Rtr  0          10.1.1.1         800004cb 264  e06e  40    Yes
0               Iap  0          10.1.1.1         800004dc 264  9de4  52    Yes
0               Grc  1          10.2.2.2         80000001 17   a8a6  32    Yes
```

Syntax: show ipv6 ospf database

Show ipv6 ospf data summary

```
This command displays the IPv6 OSPF data summary.
device(config-ospf6-router)#show ipv6 ospf data summary
 AS scope:
                       Active     MaxAge
 ASExternal                 0          0
 Area 0 scope:
                       Active     MaxAge
 Router                     1          0
 Network                    0          0
 InterPrefix                0          0
 InterRouter                0          0
 Type7                      0          0
 IntraPrefix                1          0
 Other                      0          0
 Total                      0          0
  Interface scope (over 1 interfaces):
                        Active     MaxAge
  Link                       0          0
  Grace                      1          0
  Other                      0          0
  Total                      1          0
 Total: 3 LSAs, 3 Active LSAs, 0 MaxAge LSAs
```

Syntax: show ipv6 ospf data summary

Show ipv6 ospf database grace

```
This command shows the IPv6 OSPF LSA timer grace period configuration.
device#show ipv6 ospf database grace
  LSA Key - Rtr:Router Net:Network Inap:InterPrefix Inar:InterRouter
          Extn:ASExternal Grp:GroupMembership Typ7:Type7 Link:Link
          Iap:IntraPrefix Grc: Grace
 Area ID         Type LSID       Adv Rtr         Seq(Hex) Age  Cksum Len
0               Grc  1          10.4.4.4         80000001 17   a8a6  36
    Restart duration: 150
    Restart Reason: Software Reload
```

Syntax: show ipv6 ospf database grace

Configuring Non Stop Routing on IS-IS

NOTE
IPv6 IS-IS NSR is not supported on the Brocade NetIron CES Series and Brocade NetIron CER Series platforms.

IS-IS IPv6 supports nonstop routing. The following command is used to configure NSR. Further configuration details are available in Chapter 54.

```
device(config-isis-router)#nononstop-routing
```

Syntax: [no] nonstop-routing

The**nonstop routing** command enables nonstop routing in IS-IS IPv6.

Show commands

Show isis

This command shows the IS-IS configuration.

```
device#show isis
IS-IS Routing Protocol Operation State: Enabled
 IS-Type: Level-1-2
 ...
 Global Hello Padding For Point to Point Circuits: Enabled
 Ptpt Three Way HandShake Mechanism: Enabled
 BGP Ipv4 Converged: FALSE, Ipv6 Converged: FALSE
 IS-IS Traffic Engineering Support: Disabled
 No ISIS Shortcuts Configured
 BFD: Disabled
 NSR: Enabled
    NSR State: Normal
    Standby MP: Active
    Sync State: Enabled
 Interfaces with IPv4 IS-IS configured:
    None
 ...
```

Configuring BGP graceful restart

BGP IPv6 supports graceful restart.

- BGP informs Graceful Restart capability to its peer.
- BGP peers retains BGP routing information and help Graceful Restart process.

The following command is used to configure graceful restart.

```
device(config-bgp-router)#graceful-restart
```

Syntax: [no] graceful-restart [**purge-time**] [**restart-time**] [**stale-routes-time**]

The**graceful-restart** command enables graceful restart for the address-family. The **purge-time** command is used to configure the maximum time in seconds before stale routes are purged. The **purge-time** cannot be less than the time set for the **stale-routes-time** .

The **restart-time** command is used to configure the maximum restart time advertised to neighbors in seconds. The**stale-routes-time** command is used to configure the maximum wait time in seconds for BGP EOR marker.

Show commands

Show running-configuration

```
This command shows the running configuration.
device#show running-config
Current BGP configuration:
router bgp
 local-as 200
 neighbor 2001:DB8:22::6 remote-as 100

 address-family ipv4 unicast
 graceful-restart stale-routes-time 100
 graceful-restart purge-time 100
 graceful-restart
 no neighbor 2001:DB8:22::6 activate
 exit-address-family

 address-family ipv4 multicast
 exit-address-family

 address-family ipv6 unicast
 graceful-restart restart-time 160
 graceful-restart stale-routes-time 120
 graceful-restart purge-time 120
 graceful-restart
 neighbor 2001:DB8:22::6 activate
 exit-address-family

 address-family ipv6 multicast
 exit-address-family

 address-family l2vpn vpls
 exit-address-family
end of BGP configuration
```

Show ipv6 bgp neighbors *IP address*

```
This command shows the running configuration.
device#show ipv6 bgp neighbors 2001:DB8:22::6
1   IP Address: 2001:DB8:22::6, AS: 100 (EBGP), RouterID: 10.6.6.6, VRF: default-vrf
    State: ESTABLISHED, Time: 2h24m36s, KeepAliveTime: 60, HoldTime: 180
       KeepAliveTimer Expire in 16 seconds, HoldTimer Expire in 142 seconds
    Minimal Route Advertisement Interval: 0 seconds
       RefreshCapability: Received
       GracefulRestartCapability: Sent
           Restart Time 160 sec, Restart bit 0
           afi/safi 2/1, Forwarding bit 0
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
       Sent    : 1       1       164       0            0
       Received: 1       0       164       0            0
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                  Tx: ---        ---           Rx: ---        ---
    Last Connection Reset Reason:Unknown
    Notification Sent:     Unspecified
    Notification Received: Unspecified
    Neighbor NLRI Negotiation:
      Peer Negotiated IPV6 unicast  capability
      Peer configured for IPV6 unicast  Routes
    Neighbor ipv6 MPLS Label Capability Negotiation:
    Neighbor AS4 Capability Negotiation:
    Outbound Policy Group:
       ID: 2, Use Count: 1
    BFD:Disabled
    TCP Connection state: ESTABLISHED, flags:00000044 (0,0)
     Maximum segment size: 1440
```

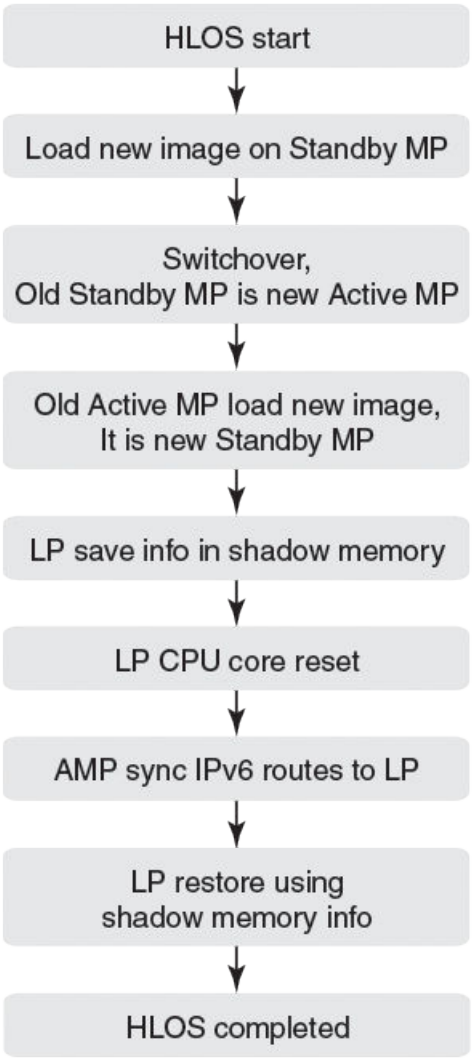
Syntax: Show ipv6 bgp neighbors *IPaddress*

IPv6 Hitless OS upgrade

OSPFv3, IS-IS IPv6, and BGP IPv6 support both failover and Hitless Operating System Switchover (HLOS). HLOS provides a platform support mechanism to upgrade image without disrupting routing and forwarding service.

The process of syncing routes between a new MP and its LPs using the new timers are illustrated in Figure 31 and described in the following steps.

FIGURE 31 IPv6 HLOS operation



1. HLOS starts and the Standby MP is rebooted with a new image.
2. System switches over and the Standby MP takes role of the Active MP.
3. The old Active MP is rebooted with the new image and it takes the role of the Standby MP.
4. Once the Active and Standby MP are in sync, the LP backs up the necessary IPv6 route information.
5. The LP CPU core resets, once the core reset is complete the LP receives IPv6 route information from Active MP.

6. The LP restores the complete IPv6 routes using the information synced from the Active MP to the LP and the backed up information on the LP.
7. HLOS complete.

Configuring IPv4 and IPv6 protocol stacks

If a device is deployed as an endpoint for an IPv6 over IPv4 tunnel, you must configure the device to support IPv4 and IPv6 protocol stacks. Each interface that sends and receives IPv4 and IPv6 traffic must be configured with an IPv4 address and an IPv6 address. You can also explicitly enable IPv6 using the **ipv6 enable** command. Refer to Configuring a link-local IPv6 address on page 392.)

To configure an interface to support both IPv4 and IPv6 protocol stacks, enter commands such as the following.

```
device(config)# ipv6 unicast-routing
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ip address 192.168.1.1 255.255.255.0
device(config-if-e100-3/1)# ipv6 address 2001:DB8:12d:1300::/64 eui-64
```

These commands globally enable IPv6 routing on the device, and configure an IPv4 address and an IPv6 address for Ethernet interface 3/1.

Syntax: [no] ipv6 unicast-routing

To disable IPv6 traffic globally on the Brocade device, enter the **no** form of this command.

Syntax: [no] ip address *ip-address sub-net-mask* [**secondary**]

You must specify the *ip-address* parameter using 8-bit values in dotted decimal notation.

You can specify the *sub-net-mask* parameter in either dotted decimal notation or as a decimal value preceded by a slash mark (/).

The **secondary** keyword specifies that the configured address is a secondary IPv4 address.

To remove the IPv4 address from the interface, enter the **no** form of this command.

Syntax: [no] ipv6 address *ipv6-prefix/prefix-length* [**eui-64**]

This syntax specifies a global or unique local IPv6 unicast address. For information about configuring a link-local IPv6 address, refer to Configuring a link-local IPv6 address on page 392.

You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The **eui-64** keyword configures the global or unique local unicast address with an EUI-64 interface ID in the low-order 64 bits. The interface ID is automatically constructed in IEEE EUI-64 format using the MAC address of the interface. If you do not specify the **eui-64** keyword, you must manually configure the 64-bit interface ID as well as the 64-bit network prefix. For more information about manually configuring an interface ID, refer to Configuring a global or unique local IPv6 unicast address on page 391.

Configuring IPv6 Domain Name Server (DNS) resolver

The Domain Name Server (DNS) resolver feature lets you use a host name to perform Telnet, ping, and traceroute commands. You can also define a DNS domain on a device to recognize all hosts within that domain. After you define a domain name, the device automatically appends the appropriate domain to the host and forwards it to the domain name server.

For example, if the domain "example.com" is defined on a device, and you want to initiate a ping to host "EXC01" on that domain, you only need to reference the host name instead of the host name and the domain name. For example, enter either of the following commands to initiate the ping.

```
device# ping exc01
device# ping exc01.example.com
```

Defining a DNS entry

You can define up to four DNS servers for each DNS entry. The first entry serves as the primary default address. If a query to the primary address is not resolved after three attempts, the next gateway address is queried (up to three times). This process continues for each defined gateway address until the query is resolved. The order in which the default gateway addresses are polled is the same as the order in which you enter them.

To define the domain name *example.com* on a device and then define four possible default DNS gateway addresses, using IPv4 addressing, enter the following commands.

```
device(config)# ip dns domain-name example.com
device(config)# ip dns server-address 10.157.22.199 10.96.7.15 10.95.7.25 10.98.7.15
```

Syntax: [no] ip dns server-address *ip-addr* [*ip-addr*] [*ip-addr*] [*ip-addr*]

In this example, the first IP address in the command becomes the primary gateway address and all others are secondary addresses. Because IP address 10.98.7.15 is the last address listed, it is also the last address consulted to resolve a query.

Defining an IPv6 DNS entry

IPv6 defines new DNS record types to resolve queries for domain names to IPv6 addresses, as well as IPv6 addresses to domain names. Devices running IPv6 software support AAAA DNS records, which are defined in RFC 1886.

AAAA DNS records are analogous to the A DNS records used with IPv4. A complete IPv6 address is stored in each record. AAAA records have a type value of 28.

To establish an IPv6 DNS entry for the device, enter the following command.

```
device(config)# ipv6 dns domain-name example.com
```

Syntax: [no] ipv6 dns domain-name *domainname*

To define an IPv6 DNS server address, enter the following command.

```
device(config)# ipv6 dns server-address 2001:DB8::1
```

Syntax: [no] ipv6 dns server-address *ipv6-addr* [*ipv6-addr*] [*ipv6-addr*] [*ipv6-addr*]

For example, in a configuration where *ftp6.example.com* is a server with an IPv6 protocol stack, when a user pings *ftp6.example.com* , the device attempts to resolve the AAAA DNS record. In addition, if

the DNS server does not have an IPv6 address, as long as it is able to resolve AAAA records, it can still respond to DNS queries.

DNS queries of IPv4 and IPv6 DNS servers

IPv4 and IPv6 DNS record queries search through IPv4 and IPv6 DNS servers are described here.

For IPv4 DNS record queries:

- Loop through all configured IPv4 DNS servers.
- If no IPv4 DNS servers are configured, then loop through all configured IPv6 DNS servers (if any).

For IPv6 DNS record queries:

- Loop through all configured IPv6 DNS servers.
- If no IPv6 DNS servers are configured, then loop through all configured IPv4 DNS servers (if any).

IPv6 Non-Stop Routing support

When IPv6 Non-Stop-Routing (NSR) is used, peer networking devices do not have knowledge of any event on the switching over router. All information needed to continue the routing protocol peering state is transferred to the standby processor so it can pick up immediately upon a switchover. As NSR does not need the help of neighboring routers during restart, the NSR-capable routers can be deployed independently in an existing network.

This section describes support for IPv6 Non-Stop-Routing (NSR) on Brocade NetIron MLX Series and Brocade NetIron XMR Series devices. The scope of this section is for IPv6 unicast routing only.

Limitations

- Configuration events that occur closer to switchover may get lost due to CLI synchronization issues.
- Neighbor, interface, or NSSA translation state changes that occur close to and during the switchover will not be handled.
- Counters - Traffic counters will not be synchronized. Neighbor and LSA DB counters will be recalculated on Standby during sync and thus not synchronized.
- OSPF Database Overflow condition for External LSAs - depending on the sequence of redistribution or new LSAs (from neighbors) the LSAs accepted within the limits of the database may change upon switchover.
- The NSR hitless failover event may not be completely transparent to the network as after switchover additional flooding related protocol traffic will be generated to the directly connected neighbors.
- OSPF Startup Timers - will not be applied upon NSR switchover.

Configuring IPv6 NSR support

Use the following commands to configure IPv6 Non-Stop Routing support.

The **graceful-restart ipv6 max-hold-timer** sets the hold interval.

```
device(config)#graceful-restart ipv6 max-hold-timer 100
```

Syntax: [no] graceful-restart ipv6 max-hold-timer *hold-interval*

The acceptable range for the maximum hold time before sync up forwarding information is 30 to 3600 seconds. The default is 300 seconds.

The **graceful-restart ipv6 protocols-converge-timer** sets the convergence interval. The default setting is 5 seconds.

```
device(config)#no graceful-restart ipv6 protocol-convergence-timer 50
```

Syntax: [no] graceful-restart ipv6 protocols-converge-timer *convergence-interval*

The acceptable range for the maximum time for protocols to converge after a graceful restart is 0 to 1200 seconds. The default protocol convergence time is 5seconds.

ECMP load sharing for IPv6

IPv6 ECMP load sharing is hardware-managed. If there is more than one path to a given destination, a hash is calculated based on the source MAC address, destination MAC address, source IPv6 address, destination IPv6 address, and TCP/UDP source port and destination port (if the packet is also a TCP and UDP packet). This hash is used to select one of the paths.

Disabling or re-enabling ECMP load sharing for IPv6

ECMP load sharing for IPv6 is enabled by default. To disable the feature, enter the following command.

```
device(config)# no ipv6 load-sharing
```

To re-enable the feature after disabling it, enter the following command.

```
device(config)# ipv6 load-sharing 4
```

Syntax: [no] ipv6 load-sharing *number*

The *number* parameter specifies the number of ECMP load sharing paths. Enter a value between 2 and 32 for *number* to set the maximum number of paths. The default value is 4.

NOTE
The maximum number of paths supported by the BR-MLX-10Gx24-DM module is 16.

Changing the maximum number of load sharing paths for IPv6

By default, IPv6 ECMP load sharing balances traffic across up to four equal paths. You can change the maximum number of paths to a value between 2 and 32.

To change the number of ECMP load sharing paths for IPv6, enter the following command:

```
device(config)# ipv6 load-sharing 8
```

Syntax: [no] ipv6 load-sharing *num* **ber**

The *number* parameter specifies the number of ECMP load sharing paths. Enter a value between 2 and 32 for *number* to set the maximum number of paths. The default value is 4.

NOTE
The maximum number of paths supported by the BR-MLX-10Gx24-DM module is 16.

DHCP relay agent for IPv6

A client locates a DHCP server using a reserved, link-scoped multicast address. Direct communication between client and the server requires that they be attached by the same link. In some situations where ease-of-management, economy, and scalability are concerns, you can allow a DHCPv6 client to send a message to a DHCP server using a DHCPv6 relay agent. A DHCPv6 relay agent, which may reside on the client link, but is transparent to the client, relays messages between the client and the server.

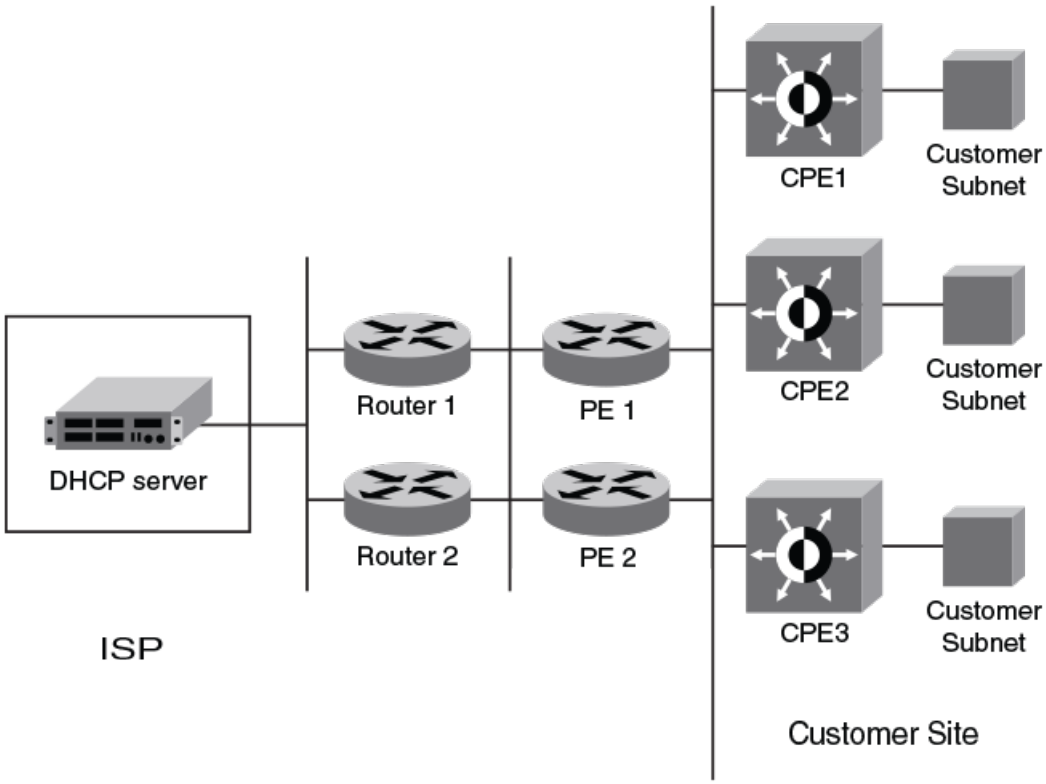
When the relay agent receives a message, it creates a new relay-forward message, inserts the original DHCPv6 message, and sends the relay-forward message as the DHCP server.

DHCPv6 Relay Agent Prefix Delegation Notification

DHCPv6 Relay Agent Prefix Delegation Notification feature allows a DHCPv6 server to dynamically delegate IPv6 prefixes to a DHCPv6 client using the DHCPv6 Prefix Delegation (PD) option. DHCPv6 prefix delegation enables an Internet service provider (ISP) to automate the process of assigning prefixes to a customer premises equipment (CPE) network. The CPE then assigns IPv6 subnets from the delegated IPv6 prefix to its downstream customer interfaces.

This feature description is shown in Figure 32 .

FIGURE 32 DHCPv6 Relay Agent Prefix Delegation Notification



A route is added to the IPv6 route table on the provider edge router (PE) for the delegated prefix to be delegated to requesting routers. The delegating router chooses a prefix for delegation and responds with it to the requesting router. to the external network and to enable the correct forwarding of the IPv6 packets for the delegated IPv6 prefix. Adding the delegated prefix to the IPv6 route table ensures that the unicast Reverse Path Forwarding (uRPF) works correctly.

Since the PE is also a DHCPv6 relay agent (it relays DHCPv6 messages between the CPE and the DHCP server), it examines all DHCPv6 messages relayed between the CPE and the DHCP server and gathers information about a delegated prefix and then manages the advertisement of this delegated prefix to the external network.

DHCPv6 Relay Agent Prefix Delegation Notification limitations

The following limitations apply to the DHCPv6 Relay Agent Prefix Delegation Notification.

- The PD notification fails when the DHCPv6 messages between a DHCPv6 server and a DHCPv6 client containing the PD option are not relayed via the DHCPv6 relay agent.
- If the delegated prefix is released or renewed by the client at the time when the DHCPv6 relay agent is down or rebooting, then this release or renewal of the delegated prefix will not be detected by the relay agent.
- If there is no sufficient disk space on a flash disk, then the system may not store all the delegated prefixes in the IPv6 route table.

Upgrade and downgrade considerations

- When a router is upgraded to the version of software that supports this feature, the saved information about delegated prefixes will be examined and if the delegated prefix lifetime is not expired, then the prefix will be added to the IPv6 static route table.
- When a router is downgraded to the version of software that does not support this feature, the saved information about delegated prefixes is retained and it cannot be used.

Configuring DHCP for IPv6 relay agent

You can enable the DHCP for IPv6 relay agent function and specify the relay destination address (i.e. the DHCP server) on an interface by entering this command at the interface level.

```
device(config)# interface ethernet 2/3
device(config-if-e10000-2/3)#ipv6 dhcp-relay destination 2001:DB8::2
```

Syntax: [no] ipv6 dhcp-relay destination *ipv6-address*

Specify the *ipv6-address* as a destination address to which client messages are forwarded and which enables DHCP for IPv6 relay service on the interface. A maximum of 16 relay destination addresses may be entered.

Configuring DHCPv6 Relay Agent Prefix Delegation Notification

To set the number of delegated prefixes that can be learned at the global system level, use the **ipv6 dhcp-relay maximum-delegated-prefixes** command.

By default, the DHCPv6 Relay Agent Prefix Delegation Notification is enabled when the DHCPv6 relay agent feature is enabled on an interface. User can disable the DHCPv6 Relay Agent Prefix Delegation Notification at system or interface level by setting **ipv6 dhcp-relay maximum-delegated-prefixes** to 0 at system level or interface level as required.

NOTE
There should be a minimum free space of 7 MB in the flash memory to save information about delegated prefixes in flash on both the Active and Standby management processor

```
device(config)# ipv6 dhcp-relay maximum-delegated-prefixes 5000
```

Syntax: [no] ipv6 dhcp-relay maximum-delegated-prefixes *value*

The *value* parameter is used to limit the maximum number of prefixes that can be learnt at the global level. The range is from 0 to 100000. The default value varies for different platforms.

Use the **no ipv6 dhcp-relay maximum-delegated-prefixes** command to set the parameter to the default value of the specified platform. See Enabling DHCPv6 Relay Agent Prefix Delegation notification on an interface on page 413 for more information.

Enabling DHCPv6 Relay Agent Prefix Delegation notification on an interface

To set the number of delegated prefixes that can be learned at the interface level, use the **ipv6 dhcp-relay maximum-delegated-prefixes** command. This command limits the maximum number of prefixes that can be learnt on the interface.

```
device(config-if-eth2/1)# ipv6 dhcp-relay maximum-delegated-prefixes 4000
```

Syntax: [no] ipv6 dhcp-relay maximum-delegated-prefixes *value*

The *value* parameter is used to limit the maximum number of prefixes that can be delegated. The range is from 0 to 20000. The default value is 20000.

Use the **no ipv6 dhcp-relay maximum-delegated-prefixes** command to set the parameter to the default value of the specified platform. See Table 67 for more information.

Table 67 lists the default and maximum prefix values for different platforms.

TABLE 67 Default and maximum values for different platforms

Platform	Default Maximum System Prefixes supported	Maximum System Prefixes supported	Default Maximum Interface Prefixes supported	Maximum interface Prefixes supported	Default Maximum IPv6 Route	Maximum IPv6 routes that can be supported
Brocade NetIron CES Series	1000	8000	250	2000	1024	8192
Brocade NetIron CER Series	8000	100000	2000	20000	8192	131072
Brocade MLX and MLXe series	32000	100000	8000	20000	32768	114688
Brocade NetIron XMR Series	60000	100000	20000	20000	65536	245760

Assigning the administrative distance to DHCPv6 static routes

To assign the administrative distance to DHCPv6 static routes installed in IPv6 route table for the delegated prefixes on the interface, use the **ipv6 dhcp-relay distance** command at the interface level. The administrative distance value has to be set so that it does not replace the same IPv6 static route configured by the user.

```
device(config-if-eth2/1)# ipv6 dhcp-relay distance 25
```

Syntax: [no] ipv6 dhcp-relay distance *value*

The *value* parameter is used to assign the administrative distance to DHCPv6 static routes on the interface. The range is from 1 to 255. The default value is 10. If the value is set to 255, then the delegated prefixes for this interface will not be installed in the IPv6 static route table.

Use the **no ipv6 dhcp-relay distance** command to set the parameter to a default value of 10.

Displaying the DHCPv6 Relay Agent Prefix Delegation Notification information

Enter the **show ipv6 dhcp-relay delegated-prefixes** command to display information about the delegated prefixes.

```
device#show ipv6 dhcp-relay delegated-prefixes vrf red
IPv6 DHCP Relay Delegated Prefixes Table - 2 entries VRF: red
IPv6 Prefix           Client              Interface     ExpireTime
2001:db8:aaa::/48     2001:db8:103:10:1::8        eth 1/3       3h24m10s
2001:db8:bbb::/48     2001:db8:104:10:1::6        eth 1/4       0m28s
device#
```

Syntax: show ipv6 dhcp-relay delegated-prefixes vrf *vrf-name* } { *X:X::X:X/M* | **client-id** *client ipv6address* | **interface** *interface-id* }

The **vrf***vrf-name* parameter is used to display the DHCPv6 delegated prefixes for a specific VRF.

The *X:X::X:X/M* parameter is used to display the specified delegated prefix information.

The **client-id***client ipv6 address* parameter is used to display the delegated prefix for the specific client.

The **interface***interface-id* parameter is used to display delegated prefixes for the specified outgoing interface.

Table 68 describes the fields from the output of **show ipv6 dhcp-relay delegated-prefixes** command.

TABLE 68 Output from the show ipv6 dhcp-relay delegated-prefixes command

Field	Description
IPv6 Prefix	The IPv6 prefix delegated to the client.
Client	The IPv6 address of the client.
Interface	The interface on which the DHCPv6 messages are relayed to the client.
ExpireTime	The remaining lifetime of the delegated prefix.

Displaying the DHCPv6 Relay configured destinations

Enter the **show ipv6 dhcp-relay destinations** command to display information about the delegated prefixes' configured destinations for a specific interface.

```
device#show ipv6 dhcp-relay destinations
DHCPv6 Relay Destinations:
Interface ve 100:
   Destination                  OutgoingInterface
   2001:db8:1::39                  NA
Interface ve 101:
   Destination                  OutgoingInterface
   2001:db8:1::39                  NA
Interface ve 102:
   Destination                  OutgoingInterface
   2001:db8:1::39                  NA
```

Syntax: show ipv6 dhcp-relay destinations

Table 69 describes the fields from the output of **show ipv6 dhcp-relay destinations** command.

TABLE 69 Output from the show ipv6 dhcp-relay destinations command

Field	Description
Destination	The configured destination IPv6 address.
OutgoingInterface	The interface on which packets will be relayed if the destination relay address is local link or multicast.

Displaying the DHCPv6 Relay Agent options

Enter the **show ipv6 dhcp-relay options** command to display information about the relay options available to the prefixed delegates for a specific interface.

```
device# show ipv6 dhcp-relay options
DHCPv6 Relay Options Information:
Interface       Interface-Id     Remote-Id
ve 100          No               No
ve 101          Yes              No
ve 102          No               Yes
```

Syntax: show ipv6 dhcp-relay options

Table 70 describes the fields from the output of **show ipv6 dhcp-relay options** command.

TABLE 70 Output from the show ipv6 dhcp-relay options command

Field	Description
Interface	The interface name.
Interface-Id	The interface ID option. Yes or No indicates if the option is used or not.
Remote-Id	The remote ID option. Yes or No indicates if the option is used or not.

Displaying the DHCPv6 Relay prefix delegation information

Enter the **show ipv6 dhcp-relay prefix-delegation-information** command to display additional information about the DHCPv6 prefix delegation.

```
device#show ipv6 dhcp prefix-delegation-information
DHCPv6 Relay Prefix Delegation Notification Information:
  Interface  Current    Maximum    AdminDistance
  ve 100     20         20000      10
  ve 101     4000       20000      10
  ve 102     0          20000      10
  ve 103     0          20000      10
  ve 104     0          20000      10
  ve 105     0          20000      10
```

Syntax: show ipv6 dhcp-relay prefix-delegation-information

Table 71 describes the fields from the output of the **show ipv6 dhcp-relay prefix-delegation-information** command.

TABLE 71 Output from the show ipv6 dhcp-relay prefix-delegation-information command

Field	Description
Interface	The interface name.
Current	The number of delegated prefixes currently learned on the interface.
Maximum	The maximum number of delegated prefixes that can be learned on the interface.
AdminDistance	The current administrative distance used for prefixes learned on this interface when added to the IPv6 static route table.

Displaying the DHCPv6 Relay information for an interface

Enter the **show ipv6 dhcp-relay interface** command to display DHCPv6 relay information for a specific interface.

```
device#show ipv6 dhcp-relay interface ve 100
DHCPv6 Relay Information for interface ve 100:
Destinations:
 Destination                    OutgoingInterface
 2001:db8:1::39                    NA
Options:
 Interface-Id: No     Remote-Id:No
Prefix Delegation Notification:
 Current:0 Maximum:20000 AdminDistance:10
```

Syntax: show ipv6 dhcp-relay interface *interfacetype*

The *interface type* is interface type such as ethernet, POS, or VE and the specific port number.

Table 72 describes the fields from the output of the **show ipv6 dhcp-relay interface** command.

TABLE 72 Output from the show ipv6 dhcp-relay interface command

Field	Description
Destinations	The DHCPv6 relay destination configured on the interface. • **Destination** : The configured destination IPv6 address. • **OutgoingInterface** : The interface on which packet will be relayed if the destination relay address is link local or multicast.
Options	The current information about DHCPv6 relay options for the interface. • **Interface-Id** : The interface ID option indicating if the option is used or not. • **Remote-Id** : The remote ID option indicating if the option is used or not.
Prefix Delegation Notification	This current information about the DHCPv6 Prefix Delegation for the interface. • **Interface:** The name of the interface. • **Current** - The number of delegated prefixes currently learned on the interface. • **Maximum** - The maximum number of delegated prefixes that can be learned on the interface. • **AdminDistance** - The administrative distance used for prefixes learned on the specific interface when added to IPv6 Static Route table.

Clearing the DHCPv6 delegated prefixes

To clear the DHCPv6 delegated prefixes for specific VRFs, use the **clear ipv6 dhcp-relay delegated-prefixes** command at the privilege level.

```
device# clear ipv6 dhcp-relay delegated-prefixes vrf VRF1
```

Syntax: clear ipv6 dhcp-relay delegated-prefixes { **vrf** *vrf-name* } { **X:X::X:X/M** | **all** | **interface** *interface-id* }

The **vrf***vrf-name* parameter is used to clear the DHCPv6 delegated prefixes for a specific VRF. If this parameter is not provided, then the information for the default VRF is cleared.

The *X:X::X:X/M* parameter is used to clear the specified delegated prefix and remove the corresponding route permanently from the router.

The **all** parameter is used to clear all the delegated prefixes and remove the corresponding routes permanently from the router for the VRF.

The **interface***interface-id* parameter is used to clear all the delegated prefixes and remove the corresponding routes permanently from the router for the specified outgoing interface.

Clearing the DHCPv6 packet counters

To clear all DHCPv6 packet counters, use the **clear ipv6 dhcp-relay statistics** command at the privilege level.

```
device# clear ipv6 dhcp-relay statistics
```

Syntax: clear ipv6 dhcp-relay statistics

Enabling support for network-based ECMP load sharing for IPv6

If network-based ECMP load sharing is configured, traffic is distributed across equal-cost paths based on the destination network address. Routes to each network are stored in CAM and accessed when a path to a network is required. Because multiple hosts are likely to reside on a network, this method uses fewer CAM entries than load sharing by host. When you configure network-based ECMP load sharing, you can choose either of the following CAM modes:

- **Dynamic mode** - In the dynamic mode, routes are entered into the CAM dynamically using a flow-based scheme, where routes are only added to the CAM as they are required. Once routes are added to the CAM, they can be aged-out when they are not in use. Because this mode conserves CAM, it is useful for situations where CAM resources are stressed or limited.
- **Static mode** - In the static mode, routes are entered into the CAM whenever they are discovered. Routes are not aged once routes are added to the CAM and can be aged-out when they are not in use.

IPv6 VPN CAM supports ECMP load sharing, which is created for IPv6 VPN routes.

Configuring the CAM mode to support network-based ECMP load sharing for IPv6

To configure the CAM mode to support network-based ECMP load sharing for IPv6, enter a command such as the following at the Global Configuration level.

```
device(config)# cam-mode ipv6 dynamic
```

Syntax: [no] cam-mode ipv6 [**dynamic** | **static** | **host**]

The **dynamic** parameter configures the device for network-based ECMP load sharing using the dynamic CAM mode.

The**static** parameter configures the device for network-based ECMP load sharing using the static CAM mode.

The **host** parameter configures the device for host-based ECMP load sharing using the dynamic CAM mode.

You must restart the device for this command to take effect.

Displaying ECMP load-sharing information for IPv6

To display the status of ECMP load sharing for IPv6, enter the following command.

```
device# show ipv6
Global Settings
  unicast-routing enabled, ipv6 allowed to run, hop-limit 64
  reverse-path-check disabled
  urpf-exclude-default disabled
  session-logging-age 5
  No Inbound Access List Set
  No Outbound Access List Set
  Prefix-based IPv6 Load-sharing is Enabled, Number of load share paths: 4
  source-route disabled, forward-source-route disabled
Configured Static Routes: 66
Configured Static Mroutes: 66
RIP: enabled
OSPF (default VRF): enabled
BGP: enabled, 1 active neighbor(s) configured
```

Syntax: show ipv6

You can display the entries in the IPv6 forwarding cache by entering the **show ipv6 cache** command.

```
device# show ipv6 cache
Total number of IPv6 and IPv6 VPN cache entries: 3
       IPv6 Address                        Next Hop                      Interface
1      6000::                              LOCAL                         ve 60
2      6000::2                             LOCAL                         ve 60
3      fe80::768e:f8ff:fe2a:6200           LOCAL                         ve 60
```

Syntax: show ipv6 cache [*index-number* | *ipv6-prefix/prefix-length* | *ipv6-address* | **ethernet** *port* | **ve** *number* | **tunnel** *number*]

Configuring IPv6 ICMP

ICMP for IPv6 provides error and informational messages. The stateless auto-configuration, neighbor discovery, and path MTU discovery features use ICMP messages.

This section explains how to configure the following IPv6 ICMP options:

- ICMP rate limiting
- ICMP redirects
- ICMP unreachable address or route messages
- ICMP error messages for source-routed IPv6 packets
- ICMP error messages for an unreachable address
- ICMP messages for an unreachable route
- ICMP error messages for IPv6 packets with hop-limit 0
- ICMP error messages for CES/CER devices

Configuring ICMP rate limiting

You can limit the rate at which IPv6 ICMP error messages are sent out on a network. For this rate-limiting implementation, IPv6 ICMP uses a token bucket algorithm.

The algorithm works using a *virtual bucket* that contains a number of tokens, where each token represents the ability to send one ICMP error message. Tokens are placed in the bucket at a specified interval until the maximum allowed number of tokens is reached. For each error message ICMP sends, a token is removed from the bucket. ICMP generates a series of error messages until the bucket is empty. When the bucket is empty, further error messages cannot be sent until a new token is placed in the bucket.

You can adjust the following elements related to the token bucket algorithm:

- The interval at which tokens are added to the bucket. The default is 100 milliseconds.
- The maximum number of tokens in the bucket. The default is 10 tokens.

For example, to adjust the interval to 1000 milliseconds and the number of tokens to 100 tokens, enter the following command.

```
device(config)# ipv6 icmp error-interval 1000 100
```

Syntax: [no] ipv6 icmp error-interval *interval* [*number-of-tokens*]

The interval at which tokens are placed in the bucket has a range of 0 - 2147483647 milliseconds.

NOTE
If you keep the default interval (100 milliseconds), output from the **show run** command does not show the setting of the **ipv6 icmp error-interval** command. In addition, if you configure the interval value to a number that does not evenly divide into 100000 (100 milliseconds), the system rounds the value up to the next higher value that does divide evenly. For example, if you specify an interval value of 150, the system rounds it to 200.

ICMP rate limiting is enabled by default. To disable ICMP rate limiting, set the interval to 0.

Enabling ICMP redirect messages

To enable ICMP redirect messages, you need to configure icmp redirect at both global level and the interface level. You can enable or disable a device to transmit ICMP redirect messages from a global level and the interface level.

To enable the ICMP redirect messages from global level, enter the following commands.

Syntax: [no] ipv6 icmp redirects

By default, IPv6 redirect is disabled and the device does not send an ICMP redirect message to a neighboring host to inform it of a better first-hop device on a path to a destination. (For more information about how ICMP redirect messages are implemented for IPv6, refer to Configuring IPv6 neighbor discovery on page 423.)

To enable the sending of ICMP redirect messages on interface 3/1, enter the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 redirects
```

To disable the ICMP redirect messages from Ethernet interface 3/1, enter the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# no ipv6 redirects
```

Syntax: [no] ipv6 redirects

Use the **show ipv6 interface***interface port-number* command to verify that the sending of ICMP redirect messages is enabled on a particular interface.

Disabling or re-enabling ICMP redirect messages

You can disable or re-enable a device to transmit ICMP redirect messages from an interface. By default, a device sends an ICMP redirect message to a neighboring host to inform it of a better first-hop device on a path to a destination. No further configuration is required to enable the sending of ICMP redirect messages. (For more information about how ICMP redirect messages are implemented for IPv6, refer to Configuring IPv6 neighbor discovery on page 423.)

For example, to disable the ICMP redirect messages from Ethernet interface 3/1, enter the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# no ipv6 redirects
```

Syntax: [no] ipv6 redirects

To re-enable the sending of ICMP redirect messages on Ethernet interface 3/1, enter the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 redirects
```

Use the **show ipv6 interface***interface port-number* command to verify that the sending of ICMP redirect messages is enabled on a particular interface.

Disabling ICMP error messages for source-routed IPv6 packets

By default, ICMP error messages are transmitted to announce discarded IPv6 source-routed packets that were addressed to one of the IPv6 addresses of a device. By default, these packets are discarded in software, as described in Software filtering of IPv6 source-routed packets on page 430.

You can disable or re-enable the sending of ICMP error messages for discarded, IPv6 source-routed packets by using the **ipv6 icmp source-route** command. Use the **no** form of this command to disable the transmission of these error messages. The following example illustrates the disabling operation.

```
device(config)# no ipv6 icmp source-route
```

Syntax: [no] ipv6 icmp source-route

Enabling ICMP error messages for an unreachable address

By default, the ICMPv6 destination unreachable messages with the code for an unreachable address are not sent for a discarded IPv6 packet. You can enable the sending of these messages by using the **ipv6 icmp unreachable address** command. This command applies globally.

For example, to enable ICMPv6 error messages for unreachable address on the current device, enter the following command.

```
device(config)# ipv6 icmp unreachable address
```

Syntax: [no] ipv6 icmp unreachable address

Use the **no** parameter in front of the **ipv6 icmp unreachable address** command to disable the sending of ICMPv6 destination unreachable messages with the code is address unreachable.

Enabling ICMP messages for an unreachable route

By default, the ICMPv6 destination unreachable messages with the code for an unreachable route are not sent for a discarded IPv6 packet. You can enable the sending of these messages by using the **ipv6 icmp unreachable route** command.

For example, to enable ICMPv6 error messages for unreachable route on the current device, enter the following command.

```
device(config)# ipv6 icmp unreachable route
```

Syntax: [no] ipv6 icmp unreachable route

Use the **no** parameter in front of the **ipv6 icmp unreachable route** command to disable the sending of ICMPv6 destination unreachable messages with the code for destination unreachable.

Enabling ICMP error messages for IPv6 packets with hop-limit 0

By default, a Brocade MLX\XMR series box does not respond to an IPv6 packet with hop-limit 0, and drops it at the hardware. You can enable or disable a device to respond to such packets with a proper ICMPv6 error message using the **ipv6 icmp hop-limit-zero** command from the global config mode.

NOTE
This command is available only on Brocade MLX\XMR series routers.

For example, to enable ICMPv6 error messages for IPv6 routed packets with hop-limit 0, enter the following command:

```
device(config)#ipv6 icmp hop-limit-zero
```

Syntax: [no] ipv6 icmp hop-limit-zero

Use the **show running-configuration** command to see if this is enabled or disabled. Use the no parameter in front of the**ipv6 icmp hop-limit-zero** command to disable the sending of ICMP error messages for IPv6 Routed packet with hop-limit 0.

Enabling ICMP error messages for multicast Too Big packets

By default, the brocade box will not send an ICMPv6 Packet Too Big error message for the multicast packets. You can enable or disable (default behavior) a device to send the ICMPv6 Packet Too Big error messages for the IPv6 packets sent to multicast address destination using the **ipv6 icmp packet-too-big-for-multicast** command from the global config mode.

NOTE
This command is available on Brocade MLX\XMR and CES\CER series boxes.

For example, to enable a device to send the ICMPv6 Packet Too Big error messages for the IPv6 packets sent to multicast address destination, enter the following command:

```
device(config)#ipv6 icmp packet-too-big-for-multicast
```

Syntax: [no] ipv6 icmp packet-too-big-for-multicast

Use the **show running-configuration** command to see if this is enabled or disabled. Use the no parameter in front of the **ipv6 icmp packet-too-big-for-multicast** command to disable the sending of ICMPv6 Packet Too Big error message for the multicast packets.

Enabling ICMP error messages for CES/CER devices

By default, the Brocade CES/CER devices do not generate error message for many of the ICMPv6 error cases. You can enable or disable a device to generate error message in all the error conditions using the **ipv6 icmp generate-error-message** command from the global config mode.

NOTE
Enabling this command enables all the IPv6 packets will be sent to the CPU. This command is available only on Brocade CES/CER series devices.

For example, to enable a device to generate error message in all the error conditions, enter the following command:

```
device(config)#ipv6 icmp generate-error-message
```

Syntax: [no] ipv6 icmp generate-error-message

Use the **show running-configuration** command to see if this is enabled or disabled. Use the no parameter in front of the **ipv6 icmp generate-error-message** command to disable device to generate error message in all the error conditions.

Configuring IPv6 neighbor discovery

The neighbor discovery feature for IPv6 uses IPv6 ICMP messages to do the following:

- Determine the link-layer address of a neighbor on the same link.
- Verify that a neighbor is reachable.
- Track neighbor devices.

An IPv6 host is required to listen for and recognize the following addresses, which identify this host:

- Link-local address.
- Assigned unicast address.
- Loopback address.
- All-nodes multicast address.
- Solicited-node multicast address.
- Multicast address to all other groups to which it belongs.

You can adjust the following IPv6 neighbor discovery features:

- Neighbor solicitation messages for duplicate address detection.
- Router advertisement messages:
  - Interval between router advertisement messages.
  - Value that indicates a device is advertised as a default device (for use by all nodes on a given link).

  - Prefixes advertised in router advertisement messages.
  - Flags for host stateful autoconfiguration.
- The time that an IPv6 node considers a remote node reachable (for use by all nodes on a given link).

The default maximum value for IPv6 neighbor discovery (ND) entries is 4096 for Brocade NetIron XMR Series and Brocade NetIron MLX Series devices.

The memory is allocated for IPv4 and IPv6 separately. The maximum IPv4 ARP and IPv6 ND entries can be supported together.

Neighbor solicitation and advertisement messages

Neighbor solicitation and advertisement messages enable a node to determine the link-layer address of another node (neighbor) on the same link. (This function is similar to the function provided by the Address Resolution Protocol [ARP] in IPv4.) For example, node 1 on a link wants to determine the link-layer address of node 2 on the same link. To do so, node 1, the source node, multicasts a neighbor solicitation message. The neighbor solicitation message, which has a value of 135 in the Type field of the ICMP packet header, contains the following information:

- **Source address:** IPv6 address of node 1 interface that sends the message.
- **Destination address:** solicited-node multicast address (FF02:0:0:0:0:1:FF00::/104) that corresponds the IPv6 address of node 2.
- Link-layer address of node 1.
- A query for the link-layer address of node 2.

After receiving the neighbor solicitation message from node 1, node 2 replies by sending a neighbor advertisement message, which has a value of 136 in the Type field of the ICMP packet header. The neighbor solicitation message contains the following information:

- **Source address:** IPv6 address of the node 2 interface that sends the message.
- **Destination address:** IPv6 address of node 1.
- Link-layer address of node 2.

After node 1 receives the neighbor advertisement message from node 2, nodes 1 and 2 can now exchange packets on the link.

After the link-layer address of node 2 is determined, node 1 can send neighbor solicitation messages to node 2 to verify that it is reachable. Also, nodes 1, 2, or any other node on the same link can send a neighbor advertisement message to the all-nodes multicast address (FF02::1) if there is a change in their link-layer address.

Router advertisement and solicitation messages

Router advertisement and solicitation messages enable a node on a link to discover the devices on the same link.

Each configured interface on a link sends out a router advertisement message, which has a value of 134 in the Type field of the ICMP packet header, periodically to the all-nodes link-local multicast address (FF02::1).

A configured interface can also send a router advertisement message in response to a router solicitation message from a node on the same link. This message is sent to the unicast IPv6 address of the node that sent the router solicitation message.

At system startup, a host on a link sends a router solicitation message to the all-routers multicast address (FF01). Sending a router solicitation message, which has a value of 133 in the Type field of the ICMP packet header, enables the host to automatically configure its IPv6 address immediately instead of awaiting the next periodic router advertisement message.

Because a host at system startup typically does not have a unicast IPv6 address, the source address in the router solicitation message is usually the unspecified IPv6 address (0:0:0:0:0:0:0:0). If the host has a unicast IPv6 address, the source address is the unicast IPv6 address of the host interface sending the router solicitation message.

Entering the **ipv6 unicast-routing** command automatically enables the sending of router advertisement messages on all configured interfaces. You can configure several router advertisement message parameters. For information about disabling router advertisement messages and the router advertisement parameters you can configure, refer to Configuring reachable time for remote IPv6 nodes on page 429 and Setting IPv6 router advertisement parameters on page 426.

Neighbor redirect messages

After forwarding a packet, by default, a device can send a neighbor redirect message to a host to inform it of a better first-hop device. The host receiving the neighbor redirect message will then readdress the packet to the better device.

A device sends a neighbor redirect message only for unicast packets, only to the originating node, and to be processed by the node.

A neighbor redirect message has a value of 137 in the Type field of the ICMP packet header.

Setting neighbor solicitation parameters for duplicate address detection

Although the stateless autoconfiguration feature assigns the 64-bit interface ID portion of an IPv6 address using the MAC address of the host's NIC, duplicate MAC addresses can occur. Therefore, the duplicate address detection feature verifies that a unicast IPv6 address is unique before it is assigned to a host interface by the stateless autoconfiguration feature. Duplicate address detection verifies that a unicast IPv6 address is unique.

If duplicate address detection identifies a duplicate unicast IPv6 address, the address is not used. If the duplicate address is the link-local address of the host interface, the interface stops processing IPv6 packets.

You can configure the following neighbor solicitation message parameters that affect duplicate address detection while it verifies that a tentative unicast IPv6 address is unique:

- The number of consecutive neighbor solicitation messages that duplicate address detection sends on an interface. By default, duplicate address detection sends three neighbor solicitation messages without any follow-up messages.
- The interval in seconds at which duplicate address detection sends a neighbor solicitation message on an interface. By default, duplicate address detection sends a neighbor solicitation message every 1 second.

NOTE
For the interval at which duplicate address detection sends a neighbor solicitation message on an interface, the device uses seconds as the unit of measure instead of milliseconds.

For example, to change the number of neighbor solicitation messages sent on Ethernet interface 3/1 to two and the interval between the transmission of the two messages to 9 seconds, enter the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 nd dad attempt 2
device(config-if-e100-3/1)# ipv6 nd ns-interval 9
```

Syntax: [no] ipv6 nd dad attempt number

Syntax: [no] ipv6 nd ns-interval number

For the number of neighbor solicitation messages, you can specify between 0-255 attempts. Configuring a value of 0 disables duplicate address detection processing on the specified interface. To restore the number of messages to the default value, use the **no** form of this command.

For the interval between neighbor solicitation messages, you can specify between 0 and 4294967 seconds. Not recommended for very short intervals in normal IPv6 operation. When a non-default value is configured, the configured time is both advertised and used by the device itself. To restore the default interval, use the **no** form of this command.

Setting IPv6 router advertisement parameters

You can adjust the following parameters for router advertisement messages:

- The interval (in seconds) at which an interface sends router advertisement messages. By default, an interface sends a router advertisement message every 200 seconds.
- The "router lifetime" value, which is included in router advertisements sent from a particular interface. The value (in seconds) indicates if the device is advertised as a default device on this interface. If you set the value of this parameter to 0, the device is not advertised as a default device on an interface. If you set this parameter to a value that is not 0, the device is advertised as a default device on this interface. By default, the device lifetime value included in device advertisement messages sent from an interface is 1800 seconds.

When adjusting these parameter settings, it is recommended that the interval between device advertisement transmission be less than or equal to the device lifetime value if the device is advertised as a default device. For example, to adjust the interval of device advertisements to 300 seconds and the device lifetime value to 1900 seconds on Ethernet interface 3/1, enter the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 nd ra-interval 300
device(config-if-e100-3/1)# ipv6 nd ra-lifetime 1800
```

Syntax: [no] ipv6 nd ra-interval number

Syntax: [no] ipv6 nd ra-lifetime number

The *number* parameter in both commands indicates any numerical value.

Possible range value for **ipv6 nd ra-interval***number* **is 3 to 1800 seconds.**

Possible range value for **ipv6 nd ra-lifetime***number* **is 3 to 1800 seconds.**

To restore the default interval or device lifetime value, use the **no** form of the respective command.

Controlling prefixes advertised in IPv6 router advertisement messages

By default, router advertisement messages include prefixes configured as addresses on interfaces using the **ipv6 address** command. You can use the **ipv6 nd prefix-advertisement** command to control exactly which prefixes are included in router advertisement messages. Along with which prefixes the router advertisement messages contain, you can also specify the following parameters:

- Valid lifetime -- (Mandatory) The time interval (in seconds) in which the specified prefix is advertised as valid. The default is 2592000 seconds (30 days). When the timer expires, the prefix is no longer considered to be valid.
- Preferred lifetime -- (Mandatory) The time interval (in seconds) in which the specified prefix is advertised as preferred. The default is 604800 seconds (7 days). When the timer expires, the prefix is no longer considered to be preferred.

- Onlink flag -- (Optional) If this flag is set, the specified prefix is assigned to the link upon which it is advertised. Nodes sending traffic to addresses that contain the specified prefix consider the destination to be reachable on the local link.
- Autoconfiguration flag -- (Optional) If this flag is set, the stateless auto configuration feature can use the specified prefix in the automatic configuration of 128-bit IPv6 addresses for hosts on the local link.

For example, to advertise the prefix 2001:DB8:a487:7365::/64 in router advertisement messages sent out on Ethernet interface 3/1 with a valid lifetime of 1000 seconds, a preferred lifetime of 800 seconds, and the Onlink and Autoconfig flags set, enter the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 nd prefix-advertisement 2001:DB8:a487:7365::/64 1000
800 onlink autoconfig
```

Syntax: [no] ipv6 nd prefix-advertisement ipv6-prefix/prefix-length valid-lifetime preferred-lifetime [**autoconfig**] [**onlink**]

You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The valid lifetime and preferred lifetime is a numerical value between 0 - 4294967295 seconds. The default valid lifetime is 2592000 seconds (30 days), while the default preferred lifetime is 604800 seconds (7 days).

To remove a prefix from the router advertisement messages sent from a particular interface, use the **no** form of this command.

Configuring the Domain Name of DNS suffix

This section provides information about the IPv6 RA option that allows IPv6 routers to advertise domain names of DNS suffixes (the DNS name excluding the host) to IPv6 hosts in a local area network. This option to configure domain names is valid for any network that supports the use of ND6. The domain names that are advertised by routers are sent through RA messages to IPv6 hosts.

This option is supported only when IPv6 routing is active on the network. The newly configured domain name can be used as long as the RA router lifetime has not expired.

Configuration considerations

- A maximum of 4 domain names and their corresponding lifetime-multiplier values can be configured at the global configuration level.
- A maximum of 4 domain names and their corresponding lifetime-multiplier values can be configured per interface.
- The domain name that is configured on the interface overrides all other domain name configurations at the system level for this interface.

By default, the domain name of the DNS suffix and the lifetime multiplier information is not configured. The following examples are used to configure the domain names of a DNS suffix for a lifetime-multiplier value of 200.

```
device(config)# ipv6 nd ra-domain-name brocade.com lifetime-multiplier 200
device(config-if-e10000-1/10)# ipv6 nd ra-domain-name brocade.com lifetime-multiplier
200
```

Syntax: [no] ipv6 nd ra-domain-name string [**lifetime-multiplier decimal**]

The *string* parameter specifies the domain name of the DNS suffix.

The **lifetime-multiplier is the percentage value of the maximum router advertisement interval. The maximum router advertisement interval is the maximum time that can be allowed between sending unsolicited RA messages for DNS name resolution for a DNS entry. The lifetime-multiplier value is calculated as twice the RA lifetime. The maximum router advertisement interval percentage range is 100 through 200%. The default value for maximum router advertisement interval is 200%.**

To disable the advertisement of the specified domain name of DNS suffix in the RA message, use the **no** form of the respective command.

Configuring the recursive DNS server addresses and lifetime multiplier

This section provides information about the IPv6 RA attribute that allows IPv6 routers to advertise recursive DNS server addresses and lifetime multiplier values to IPv6 hosts in a local area network. This option to configure recursive DNS server addresses is valid for any network that supports the use of neighbor discovery (ND6). The recursive server addresses that are advertised by routers are sent through RA messages and are used to translate domain names to IP addresses.

This option is supported only when IPv6 routing is active on the network. The newly configured recursive DNS server address can be used as long as the RA router lifetime has not expired.

Configuration considerations

- A maximum of 4 recursive DNS server addresses and their corresponding lifetime-multiplier values can be configured at the global configuration level.
- A maximum of 4 recursive DNS server addresses and their corresponding lifetime-multiplier values can be configured per interface.
- The recursive DNS server address that is configured on the interface overrides all other recursive DNS server configurations at the system level for this interface.

By default, the recursive DNS server address and the lifetime multiplier information is not configured. The following examples configure the recursive DNS address for a lifetime-multiplier value of 200.

```
device(config)# ipv6 nd ra-dns-server 2001:DC8:200::3 lifetime-multiplier 200
device(config-if-e100-3/1)# ipv6 nd ra-dns-server 2001:DC8:200::3 lifetime-
multiplier 200
```

Syntax: [no] ipv6 nd ra-dns-server ipv6-address [**lifetime-multiplier decimal**]

The *ipv6-address* parameter specifies the global IPv6 address of the DNS server.

The **lifetime-multiplier is the percentage value of the maximum router advertisement interval. The maximum router advertisement interval is the maximum time that can be allowed between sending unsolicited RA messages for DNS name resolution for a DNS entry. The lifetime-multiplier decimal value is calculated as twice the RA lifetime. The percentage range is 100 through 200%. The default value for the maximum router advertisement interval is 200%.**

To disable the advertisement of the specified server address in the RA message, use the **no** form of the command.

Setting flags in IPv6 router advertisement messages

An IPv6 router advertisement message can include the following flags:

- **Managed Address Configuration** -- This flag indicates to hosts on a local link if they should use the stateful autoconfiguration feature to get IPv6 addresses for their interfaces. If the flag is set, the

hosts use stateful autoconfiguration to get addresses as well as non-IPv6-address information. If the flag is not set, the hosts do not use stateful autoconfiguration to get addresses and if the hosts can get non-IPv6-address information from stateful autoconfiguration is determined by the setting of the Other Stateful Configuration flag.

- **Other Stateful Configuration** -- This flag indicates to hosts on a local link if they can get non-IPv6 address autoconfiguration information. If the flag is set, the hosts can use stateful autoconfiguration to get non-IPv6-address information.

NOTE
When determining if hosts can use stateful autoconfiguration to get non-IPv6-address information, a set Managed Address Configuration flag overrides an unset Other Stateful Configuration flag. In this situation, the hosts can obtain non address information. However, if the Managed Address Configuration flag is not set and the Other Stateful Configuration flag is set, then the setting of the Other Stateful Configuration flag is used.

By default, the Managed Address Configuration and Other Stateful Configuration flags are not set in router advertisement messages. For example, to set these flags in router advertisement messages sent from Ethernet interface 3/1, enter the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 nd managed-config-flag
device(config-if-e100-3/1)# ipv6 nd other-config-flag
```

Syntax: [no] ipv6 nd managed-config-flag

Syntax: [no] ipv6 nd other-config-flag

To remove either flag from router advertisement messages sent on an interface, use the **no** form of the respective command.

Configuring reachable time for remote IPv6 nodes

You can configure the duration (in seconds) that a device considers a remote IPv6 node reachable. By default, an interface uses the value of 30 seconds.

The router advertisement messages sent by an interface include the amount of time specified by the **ipv6 nd reachable-time** command so that nodes on a link use the same reachable time duration. By default, the messages include a default value of 0.

NOTE
The device uses seconds, instead of milliseconds, for the interval at which it sends router advertisement messages.

It is not recommended to configure a short reachable time duration, because a short duration causes the IPv6 network devices to process the information at a greater frequency.

For example, to configure the reachable time of 40 seconds for Ethernet interface 3/1, enter the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 nd reachable-time 40
```

Syntax: [no] ipv6 nd reachable-time seconds

For the *seconds* parameter, you can specify between 0-3600 seconds. To restore the default time, use the **no** form of this command.

IPv6 source routing security enhancements

The IPv6 specification (RFC 2460) specifies support for IPv6 source-routed packets using a type 0 Routing extension header, requiring device and host to process the type 0 routing extension header. However, this requirement may leave a network open to a DoS attack.

A security enhancement disables sending IPv6 source-routed packets to IPv6 devices either completely or selectively as described in the following sections. (This enhancement conforms to RFC 5095.)

Complete filtering of IPv6 source-routed packets

Brocade devices are configured to drop all IPv6 source-routed packets in hardware and software as described:

- **Hardware** - IPv6 source-routed packets that contain a type 0 routing extension header immediately after the IPv6 header are dropped in hardware by default.
- **Software** - IPv6 source-routed packets addressed to any IPv6 address on a device (regardless of where the routing extension header is located) are dropped in software by default.

Details of hardware and software filtering of IPv6 source-routed packets is provided in the following.

Hardware filtering of IPv6 source-routed packets

All IPv6 source-routed packets that contain a type 0 routing extension header immediately after the IPv6 header are automatically dropped in hardware. This filtering is performed on both IPv6 packets that require forwarding and IPv6 packets addressed to one of the IPv6 addresses on the device without sending an ICMP error message. This filtering behavior is enabled by default. Consequently, if you want a the device to process IPv6 source-routed packets that contain a type 0 routing extension header immediately after the IPv6 header you must direct it to perform this action through use of the **ipv6 forward-source-route** command, as shown in the following.

```
device(config)# ipv6 forward-source-route
```

Syntax: [no] ipv6 forward-source-route

The default condition is for source-routed packets to be dropped. If you enable forwarding using this command, you can return to the default state by using the **no** option in front of the command.

NOTE
Source routed, IPv6 packets where the type 0 routing extension header does not follow directly after the IPv6 header are not automatically dropped in hardware.

Software filtering of IPv6 source-routed packets

By default, all IPv6 source-routed packets addressed to any IPv6 address on a Brocade device are dropped by software (regardless of where the Routing Extension Header resides). You can enable the forwarding of these packets by using the **ipv6 source-route** command, as the following example shows.

```
device(config)# ipv6 source-route
```

Syntax: [no] ipv6 source-route

The default condition is to disallow the forwarding of source-routed packets to IPv6 addresses. If you enable forwarding by using this command, you can return to the default state by using the **no** option of the command.

The **ipv6 forward-source-route** command must be enabled for the **ipv6 source-route** command to operate.

By default, ICMP error messages are sent for packets dropped by software. You can use the **ipv6 icmp source-route** command to disable the generation of ICMPv6 parameter problem for software discarded IPv6 source-routed packets addressed to one of the IPv6 addresses of a device. This is described in Disabling ICMP error messages for source-routed IPv6 packets on page 421.

Selective filtering of IPv6 source-routed packets using ACLs

You can selectively filter IPv6 source-routed packets using ACLs. This is accomplished by creating an IPv6 ACL that specifies a type 0 routing extension header. This is done using the **routing-header-type** option when configuring an IPv6 ACL. An example of an IPv6 ACL that selectively drops IPv6 source-routed packets is shown in the following.

```
device(config)# ipv6 access-list deny-access1
device(config-ipv6-access-list deny-access1)#deny ipv6 any any routing-header-type 0
```

As with complete filtering, selective filtering can be done in both hardware and software as described:

- **Hardware** - Inbound and outbound IPv6 source-routed packets that contain a type 0 routing extension header immediately after the IPv6 header can be selectively dropped in hardware through use of an IPv6 ACL and bound to an interface using the **ipv6 traffic-filter** command.
- **Software** - Inbound IPv6 source-routed packets that contain a routing extension header anywhere in a packet can be selectively dropped in software using an IPv6 ACL and bound to interfaces using the **ipv6 access-class** command.

Details about how to configure selective hardware and software filtering of IPv6 source-routed packets are provided in the following.

Selective hardware filtering of IPv6 source-routed packets

Both inbound and outbound IPv6 source-routed packets that contain a type 0 routing extension header immediately after the IPv6 header can be selectively dropped in hardware using an IPv6 ACL. source-routed packets dropped in hardware are dropped without an ICMP error message being sent. To apply an IPv6 ACL with the **routing-header-type** option for hardware filtering, you must apply the IPv6 ACL to specific ports using the **ipv6 traffic-filter** command as shown in the following example.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 traffic-filter deny-access1 in
```

Additionally, you must also enable forwarding using the**ipv6 forward-source-route** command (as shown in the following) to allow any forwarding of IPv6 source-routed packets.

```
device(config)# ipv6 forward-source-route
```

Selective software filtering of IPv6 source-routed packets

Inbound IPv6 source-routed packets that contain a routing extension header anywhere in a packet can be selectively dropped in software using an IPv6 ACL. source-routed packets dropped in software generate ICMP Destination Unreachable error messages.

NOTE
This filtering only applies to packets addressed to one of the IPv6 addresses of the device.

To apply an IPv6 ACL with the **routing-header-type** option for software filtering, you must apply the IPv6 ACL system wide using the**ipv6 access-class** command.

```
device(config)# # ipv6 access-class deny-access1 in
```

Additionally, you must also enable forwarding using the**ipv6 forward-source-route** and **ipv6 source-route** commands (as shown in the following) to allow any forwarding of IPv6 source-routed packets.

```
device(config)# ipv6 forward-source-route
device(config)# ipv6 source-route
```

Complete and selective filtering combination and order of application

If the complete filtering of IPv6 source-routed packets is enabled (the default state) then selective filtering cannot be performed. Consequently, you must use the **ipv6 forward-source-route** and **ipv6 source-route** commands to allow IPv6 source-routed packets when you are selectively allowing some IPv6 source-routed packets.

The following configuration of complete hardware and software filtering can be used with selective filtering if the commands are configured in the correct order:

- When the **ipv6 forward-source-route** command is configured, IPv6 source-routed packets that contain a type 0 routing extension header immediately after the IPv6 header are not dropped by hardware.
- All IPv6 source-routed packets addressed to any IPv6 address on a Brocade device (regardless of where the Routing Extension Header is located) are dropped by software. This is the default configuration.

When using the **ipv6 forward-source-route** and **ipv6 source-route** commands as described, the filtering is performed in the order described below.

1. Inbound filtering is performed on the receiving interface using an ACL applied using the **ipv6 traffic-filter** command. This filtering is performed using hardware.
2. Complete filtering for IPv6 source route. This filtering is performed by the CPU.
3. Selective filtering using an IPv6 ACL applied on a system-wide basis using the **ipv6 access-class** command.
4. Selective filtering by hardware using an IPv6 ACL bound to an interface for outbound traffic using the**ipv6 traffic-filter** command.

Configuration examples for complete and selective filtering of source routed packets

The following examples demonstrate how to use this feature for different purposes:

- Dropping all IPv6 Source Routed Packets on all Ports
- Dropping all IPv6 Source Routed Packets on a Specified Port
- Silently Dropping all IPv6 Source Routed Packets Addressed to IPv6 Addresses
- Dropping all IPv6 Source Routed Packets Addressed to IPv6 Addresses from a Specified Source
- Allowing IPv6 Source Routed Packets from a Specified Source on a Specified Interface

Each of these examples are described in detail in the following sections.

Dropping all IPv6 source-routed packets on all ports

By default, all IPv6 source-routed packets received on all device ports are dropped.

Dropping all IPv6 source-routed packets on a specified port

The following example shows a configuration that will drop all IPv6 source-routed packets received on port 1/1 of a device.

In this example, the IPv6 ACL is configured to drop any IPv6 packet with a type 0 routing header immediately after the IPv6 header.

```
device(config)# ipv6 access-list deny-access1
device(config-ipv6-access-list deny-access1)# deny any any ipv6 routing-header-type 0
device(config-ipv6-access-list deny-access1# permit ipv6 any any
device(config-ipv6-access-list deny-access1)# exit
```

The default is for the device to drop all IPv6 source-routed packets in hardware and software. Forwarding of these packets must be explicitly enabled using the **ipv6 forward-source-route** and**ipv6 source-route** commands as shown.

```
device(config)# ipv6 forward-source-route
device(config)# ipv6 source-route
```

The IPv6 ACL must then be bound to the interface it is intended to filter as shown in the following example for the Ethernet 1/1 interface.

```
device(config)# interface ethernet 1/1
device(config-if-e100-1/1)# ipv6 traffic-filter deny-access1 in
```

Silently dropping all IPv6 source-routed packets sent to IPv6 addresses

The following configuration drops all IPv6 source-routed packets addressed to the IPv6 addresses on a device without sending an ICMP error message.

ICMPv6 parameter problem error messages are sent for dropped IPv6 source-routed packets addressed to the IPv6 addresses on the device. To disable these messages, use the **no** option with the **ipv6 icmp source-route** command.

```
device(config)# no ipv6 icmp source-route
```

By default, the device drops all IPv6 source-routed packets in hardware and software. Use the **ipv6 forward-source- route** command to enable the forwarding of IPv6 source-routed packets with a type 0 routing extension header immediately after the IPv6 header, as shown in this example.

```
device(config)# ipv6 forward-source-route
```

Dropping all IPv6 source-routed packets to IPv6 addresses from a specified source

This configuration demonstrates how to drop all IPv6 source-routed packets sent from a specified IPv6 address.

In this example, IPv6 ACL is configured to deny IPv6 source-routed packets with a destination address of 2001:DB8:1, and permit any other IPv6 packets.

```
device(config)# ipv6 access-list deny-access2
device(config-ipv6-access-list deny-access2)# deny host 2001:DB8:1 any routing-header-
type 0
device(config-ipv6-access-list deny-access2)# permit ipv6 any any
device(config-ipv6-access-list deny-access2)# exit
```

The IPv6 ACL is then applied globally to the device for inbound traffic using the **ipv6 access-class** command as shown.

```
device(config)#ipv6 access-class deny-access2 in
```

By default, the device drops all IPv6 source-routed packets in hardware and software. Use the **ipv6 forward-source- route** and ipv6 source-route commands to enable forwarding of IPv6 source-routed packets, as shown.

```
device(config)# ipv6 forward-source-route
device(config)# ipv6 source-route
```

Changing the IPv6 MTU

The IPv6 MTU is the maximum length of an IPv6 packet that can be transmitted on a particular interface. If an IPv6 packet is longer than an MTU, the host that originated the packet breaks the packet into fragments and transmits the fragments in multiple packets that are shorter than the configured MTU. You can configure the MTU on individual interfaces. Per RFC 2460, the minimum IPv6 MTU for any interface is 1280 bytes.

NOTE
The maximum number of unique MTUs that can be configured on a Brocade NetIron CES Series or Brocade NetIron CER Series device is 12.

To configure the MTU on interface 3/1 to 1280 bytes, enter the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 mtu 1280
```

Syntax: [no] ipv6 mtu *bytes*

You can specify between 1284 - (**default-max-frame-size** minus 18). If a non-default value is configured for an interface, router advertisements include an MTU option. The minimum values you can configure are: 1298 (**IP6_MIN_MTU** + 18) for Ethernet ports.

You can configure IPv6 MTU for to be greater than 1500 bytes, although the default remains at 1500 bytes.

At the global CLI level, use the **ipv6 global-mtu** command. To define IPv6 MTU globally, enter.

```
device(config)#ipv6 global-mtu 1300
```

Syntax: [no] ipv6 global-mtu *value*

NOTE
After the configuration of **ipv6 global-mtu** , the system needs to be rebooted to enable the management ethernet controller to reconfigure it.

To define IPv6 MTU on an interface, enter.

```
device(config-if-e1000-2/1)#ipv6 mtu
```

Syntax: ipv6 mtu *value*

NOTE
If the size of a jumbo packet received on a port is equal to the maximum frame size of - 18 (Layer 2 MAC header + CRC) and if this value is greater than the IPv4/IPv6 MTU of the outgoing port, it will be forwarded to the CPU.

How to determine the actual IPv6 MTU value

An IPv6 port can obtain an MTU value from any of the following sources:

- Default IP MTU setting
- Global MTU Setting
- Interface MTU Setting

An interface determines the actual MTU value through these processes.

1. If an IPv6 interface MTU value is configured, that value is used.
2. If an IPv6 interface MTU value is not configured and an IPv6 global MTU value is configured, the configured global MTU value is used.
3. If neither an IPv6 interface MTU value or an IPv6 global MTU value are configured, the default IPv6 MTU value of 1500 is used.

Configuring static neighbor entries

In some cases, a neighbor cannot be reached using neighbor discovery. In this situation, you can add a static entry to the IPv6 neighbor discovery cache, which causes a neighbor to be reachable at all times without using neighbor discovery. (A static entry in the IPv6 neighbor discovery cache functions like a static ARP entry in IPv4.)

For example, to add a static entry for a neighbor with the IPv6 address 2001:DB8:2678::2 and link-layer address 0000.002b.8641 that is reachable through Ethernet interface 3/1, enter the following command.

```
device(config)# ipv6 neighbor 2001:DB8:2678::2 ethernet 3/1 0000.002b.8641
```

Syntax: [no] ipv6 neighbor *ipv6-address* **ethernet** *port* | **ve** *ve-number* [**ethernet** *port*] *link-layer-address*

The *ipv6-address* parameter specifies the address of the neighbor.

The **ethernet | ve** parameter specifies the interface through which to reach a neighbor. If you specify an Ethernet interface, you must also specify the port number. If you specify a VE, specify the VE number and then the Ethernet port numbers associated with the VE. The link-layer address is a 48-bit hardware address of the neighbor.

If you attempt to add an entry that already exists in the neighbor discovery cache, the software changes the already existing entry to a static entry.

To remove a static IPv6 entry from the IPv6 neighbor discovery cache, use the **no** form of this command.

Limiting the number of hops an IPv6 packet can traverse

By default, the maximum number of hops an IPv6 packet can traverse is 64. You can change this value to between 1 - 255 hops. For example, to change the maximum number of hops to 70, enter the following command.

```
device(config)# ipv6 hop-limit 70
```

Syntax: [no] ipv6 hop-limit *number*

The number of hops can be from 1 - 255.

QoS for IPv6 traffic

Configuring QoS for IPv6 traffic is generally the same as it is for IPv4 traffic. The QoS policies you configure on the device apply to both incoming IPv6 and IPv4 traffic. ACLs can be used to perform QoS for IPv6 traffic for the following values:

- dscp
- fragments
- priority-force
- priority-mapping
- source routing
- drop-precedence
- drop-precedence force

To enable QoS for IPv6 traffic, enter the following commands.

```
device(config)# port-priority
device(config)# write memory
device(config)# end
device# reload
```

Syntax: [no] port-priority

NOTE
You must save the configuration and reload the software to place the change into effect. This applies whether you are enabling QoS for IPv6 or IPv4 traffic.

The **port-priority** command globally enables QoS for IPv6 traffic on all interfaces. On Brocade devices, when QoS is enabled with the **port-priority** command, the device inserts a value in the internal header based on a combination of the following information:

- 802.1p priority
- Interface priority (if configured)
- VLAN priority (if configured)
- The DSCP field in the Type of Service (ToS) header

Configuring the rate limit for IPv6 subnet traffic

To set the rate limit for IPv6 subnet addresses, enter the **ipv6 rate-limit subnet policy-map** command.

```
device(config)# ipv6 rate-limit subnet policy-map policy1
```

Syntax: [no] ipv6 rate-limit subnet policy-map *policy-map*

The *policy-map* variable is the name you will use to reference the policy map and it can be a character string up to 64 characters long.

Clearing global IPv6 information

You can clear the following global IPv6 information:

- Entries from the IPv6 cache.
- Entries from the IPv6 neighbor table.
- IPv6 routes from the IPv6 route table.
- IPv6 traffic statistics.
- IPv6 session flows

Clearing the IPv6 cache

You can remove all entries from the IPv6 cache or specify an entry based on the following:

- IPv6 prefix.
- IPv6 address.
- Interface type.

For example, to remove entries for IPv6 address 2000:e0ff::1, enter the following command at any configuration level of the CLI.

```
device# clear ipv6 cache 2000:e0ff::1
```

Syntax: clear ipv6 cache [*ipv6-prefix/prefix-length* | *ipv6-address* | **ethernet** *port* | **tunnel** *number* | **ve** *number*] [**vrf** *vrf-name*]

You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

You must specify the *ipv6-address* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **ethernet** |**tunnel** |**ve** parameter specifies the interfaces for which you can remove cache entries. If you specify an Ethernet interface, you must also specify the port number associated with the interface. If you specify a VE or tunnel interface, also specify the VE or tunnel number, respectively.

The *vrf-name* parameter specifies the VRF for which you want to clear the cache entry. If no vrf parameter is entered, the default VRF is used.

Clearing IPv6 neighbor information

You can remove all entries from the IPv6 neighbor table or specify an entry based on the following:

- IPv6 prefix.
- IPv6 address.
- Interface type.

For example, to remove entries for Ethernet interface 3/1, enter the following command at the Privileged EXEC level or any of the CONFIG levels of the CLI.

```
device# clear ipv6 neighbor ethernet 3/1
```

Syntax: clear ipv6 neighbor [*ipv6-prefix/prefix-length* | *ipv6-address* | **ethernet** *port* | **ve** *number*]

You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

You must specify the *ipv6-address* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **ethernet** |**ve** parameter specifies the interfaces for which you can remove cache entries. If you specify an Ethernet interface, you must also specify the port number associated with the interface. If you specify a VE, you must also specify the VE number.

Clearing IPv6 routes from the IPv6 route table

You can clear all IPv6 routes or only those routes associated with a particular IPv6 prefix from the IPv6 route table and reset the routes.

For example, to clear IPv6 routes associated with the prefix 2000:7838::/32, enter the following command at any configuration level of the CLI.

```
device# clear ipv6 route 2000:7838::/32
```

Syntax: clear ipv6 route [*ipv6-prefix/prefix-length*] | **nexthop** *nexthop_ID*

The *ipv6-prefix/prefix-length* parameter clears routes associated with a particular IPv6 prefix. You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The **nexthop** option clears the nexthop information for all next hops in the routing table or for a specific entry. The *nexthop_id* parameter is a specific nexthop entry from the next hop table.

Clearing IPv6 traffic statistics

To clear all IPv6 traffic statistics (reset all fields to zero), enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
device(config)# clear ipv6 traffic
```

Syntax: clear ipv6 traffic

Clearing statistics for IPv6 subnet rate limiting

To clear the rate limit statistics for IPv6 subnet addresses, enter the **clear rate-limit ipv6 subnet** command at the configuration level.

Syntax: clear rate-limit ipv6 subnet

Displaying global IPv6 information

You can display output for the following global IPv6 parameters:

- IPv6 cache.
- IPv6 interfaces.
- IPv6 neighbors.
- IPv6 route table.
- Local IPv6 routers.
- IPv6 TCP connections and the status of individual connections.
- IPv6 traffic statistics.
- IPv6 session flows

Displaying IPv6 cache information

The IPv6 cache contains an IPv6 host table with indices to the next hop gateway and the interface on which the route was learned.

To display IPv6 cache information, enter the following command at any CLI level.

```
device# show ipv6 cache
Total number of IPv6 and IPv6 VPN cache entries: 3
       IPv6 Address                    Next Hop                    Interface
1      6000::                          LOCAL                       ve 60
2      6000::2                         LOCAL                       ve 60
3      fe80::768e:f8ff:fe2a:6200       LOCAL                       ve 60
```

Syntax: show ipv6 cache [*index-number* | *ipv6-prefix/prefix-length* | *ipv6-address* | **ethernet** *port* | **ve** *number* | **tunnel** *number*] [**vrf** *vrf-name*]

The *index-number* parameter restricts the display to the entry for the specified index number and subsequent entries.

The *ipv6-prefix/prefix-length* parameter restricts the display to the entries for the specified IPv6 prefix. You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The **ethernet | ve | tunnel** parameter restricts the display to the entries for the specified interface. The *ipv6-address* parameter restricts the display to the entries for the specified IPv6 address. You must specify this parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **vrf***vrf-name* parameter specifies the VRF for which you want to display the cache entry. If a vrf parameter is not entered, then the default VRF is used.

If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE interface, you must also specify the VE number. If you specify a tunnel interface, you must also specify the tunnel number.

This display shows the following information.

TABLE 73 IPv6 cache information fields

This field...	Displays...
Total number of cache entries	The number of entries in the cache table.
IPv6 Address	The host IPv6 address.
Next Hop	The next hop, which can be one of the following: • Direct - The next hop is directly connected to the device. • Local - The next hop is originated on this device. • *ipv6 address* - The IPv6 address of the next hop.
Port	The port on which the entry was learned.

Displaying IPv6 interface information

To display IPv6 interface information, enter the following command at any CLI level.

```
device# show ipv6 interface
device# show ipv6 interface
Type Codes - I:ISIS  O:OSPF R:RIP
Interface     Stat/Prot IGPs IPv6 Address                              VRF
eth 2/4       down/down                                                default-vrf
                        2001:db8:2017::c017:101/64
ve 60         up/up     fe80::768e:f8ff:fe2a:6200                      default-vrf
                        6000::2/64
                        6000::/64[Anycast]
```

Syntax: show ipv6 interface [*interface* [*port-number* | *number*]]

The *interface* parameter displays detailed information for a specified interface. For the interface, you can specify the **Ethernet**, **loopback**, **tunnel**, or **VE** keywords. If you specify an Ethernet interface, you must also specify the port number associated with the interface. If you specify a loopback, tunnel, or VE interface, you must also specify the number associated with the interface.

This display shows the following information.

TABLE 74 General IPv6 interface information fields

This field...	Displays...
Routing protocols	A one-letter code that represents a routing protocol that can be enabled on an interface.
Interface	The interface type, and the port number or number of the interface.
Status	The status of the interface. The entry in the Status field will be either "up/up" or "down/down".
Routing	The routing protocols enabled on the interface.
Global Unicast Address	The global unicast address of the interface.

Displaying IPv6 interface information for a specified interface

To display detailed information for a specific interface, enter a command such as the following at any CLI level.

```
device# show ipv6 interface ethernet 3/1
Brcoade# show ipv6 interface ethernet 2/2
Interface Ethernet 2/2 is up, line protocol is up
  IPv6 is enabled, link-local address is fe80::768e:f8ff:fe2a:6231 [Preferred]
  Global unicast address(es):
    180::1 [Preferred],  subnet is 180::/64
    180:: [Anycast],  subnet is 180::/64
  Joined group address(es):
    ff02::1:ff00:1
    ff02::1:ff00:0
    ff02::1:ff2a:6231
Port belongs to VRF: default-vrf
  MTU is 1500 bytes
  ICMP redirects are disabled
  ND DAD is enabled, number of DAD attempts: 3
  ND reachable time is 30000 Milliseconds
  ND advertised reachable time is 0 seconds
  ND retransmit interval is 1000 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND next router advertisement will be sent in 270 seconds
  ND router advertisements live for 1800 seconds
  No Inbound Access List Set
  No Outbound Access List Set
  IPv6 RPF mode: None  IPv6 RPF Log: Disabled
  RxPkts:       5                     TxPkts:   16
  RxBytes:      730                   TxBytes:  1936
  IPv6 unicast RPF drop: 0
  IPv6 unicast RPF suppressed drop: 0
```

This display shows the following information.

TABLE 75 Detailed IPv6 interface information fields

This field...	Displays...
Interface/line protocol status	The status of interface and line protocol. If you have disabled the interface with the**disable** command, the status will be "administratively down". Otherwise, the status is either "up" or "down".
IPv6 status/link-local address	The status of IPv6. The status is either "enabled" or "disabled". Displays the link-local address, if one is configured for the interface.
Global unicast address(es)	Displays the global unicast addresses, if one or more are configured for the interface.
Joined group address(es)	The multicast addresses that a device interface listens for and recognizes.
MTU	The setting of the maximum transmission unit (MTU) configured for the IPv6 interface. The MTU is the maximum length an IPv6 packet can have to be transmitted on the interface. If an IPv6 packet is longer than an MTU, the host that originated the packet fragments the packet and transmits its contents in multiple packets that are shorter than the configured MTU.
ICMP	The setting of the ICMP redirect parameter for the interface.
ND	The setting of the various neighbor discovery parameters for the interface.
Access List	The inbound and outbound access lists applied to the interface.

TABLE 75 Detailed IPv6 interface information fields (Continued)

This field...	Displays...
Routing protocols	The routing protocols enabled on the interface.
RxPkts	The number of packets received at the specified port. This field supports IPv4 and IPv6 packet and byte counters.
TxPkts	The number of packets transmitted from the specified port. This field supports IPv4 and IPv6 packet and byte counters.
RxBytes	The number of bytes received at the specified port. This field supports IPv4 and IPv6 packet and byte counters.s
TxBytes	The number of bytes transmitted from the specified port. This field supports IPv4 and IPv6 packet and byte counters.

Displaying interface counters for all ports

Previous versions of the Multi-Service IronWare software support IPv4 and IPv6 packet and byte counters. The contents of these counters can be displayed for all ports on a device or per port. Output from the**show ipv6 interface ethernet** command includes packet and byte counter information on a per-port basis. Refer to Displaying IPv6 interface information for a specified interface on page 441.

The default byte counters include the 20-byte per-packet Ethernet overhead. You can configure a device to exclude the 20-byte per-packet Ethernet overhead from byte accounting using the**vlan-counter exclude-overhead** command.

IPv4 and IPv6 commands display the interface counters for all ports on a Brocade device. The following example displays packet and byte counter information for all ports.

```
device# show ipv6 interface counters
Interface      RxPkts     TxPkts     RxBytes     TxBytes
eth 3/3        200        200        850000      850000
eth 3/4        500        500        40000       40000
```

Syntax: show ipv6 interface counters

Table 76 describes the fields that display interface counter statistics.

TABLE 76 Interface counter display statistics

This field...	Displays...
Interface	The interface for which counter statistics are being displayed.
RxPkts	The number of packets received at the specified port.
TxPkts	The number of packets transmitted from the specified port.
RxBytes	The number of bytes received at the specified port.
TxBytes	The number of bytes transmitted from the specified port.

Clearing the interface counters

Use the following command to clear all interface counters on a Brocade device.

```
device# clear ipv6 interface counters
```

Syntax: clear ipv6 interface counters

Use the following command to clear the interface counters for a specified port.

```
device# clear ipv6 interface ethernet 3/2
```

Syntax: clear ipv6 interface ethernet *port-number*

The*port-number* variable specifies the slot and port number for which you want to clear the interface counters.

Displaying IPv6 neighbor information

You can display the IPv6 neighbor table, which contains an entry for each IPv6 neighbor with which the device exchanges IPv6 packets.

To display the IPv6 neighbor table, enter the following command at any CLI level.

```
device(config)# show ipv neighbor ethernet 3/11
Total number of Neighbor entries: 1
Type Codes - *:Static
Entries on interface eth 3/11 :
    IPv6 Address                              VLAN LinkLayer-Addr State  Age  Port  R
1     128::                                     1    0024.3898.0f0a *REACH 422  3/11  0
device(config)#
device(config)# show ipv neighbor ethernet 3/11
Total number of Neighbor entries: 1
Entries on interface eth 3/11 :
    IPv6 Address                              VLAN LinkLayer-Addr State  Age  Port  R
1     128::                                     1    0024.3898.0f0a *REACH 432  3/11  0
After I Ping the neighbor
device(config)# ping ipv6 128::
Sending 1, 16-byte ICMPv6 Echo to 128::
timeout 5000 msec, Hop Limit 64
Type Control-c to abort
Reply from 128::: bytes=16 time=2ms Hop Limit=64
Success rate is 100 percent (1/1), round-trip min/avg/max=2/2/2 ms.
device(config)#
device(config)# show ipv neighbor ethernet 3/11
Total number of Neighbor entries: 2
Entries on interface eth 3/11 :
    IPv6 Address                              VLAN LinkLayer-Addr State  Age  Port  R
1     128::                                     1 0024.3898.0f0a *REACH 429493/11  0
2     fe80::224:38ff:fe98:f0a                   1 0024.3898.0f0a STALE  264 3/11  1
device(config)#
device(config)# show ipv neighbor ethernet 3/11
Total number of Neighbor entries: 2
Entries on interface eth 3/11 :
    IPv6 Address                              VLAN LinkLayer-Addr State  Age  Port  R
1     128::                                     1 0024.3898.0f0a *REACH 42949 3/11  0
2     fe80::224:38ff:fe98:f0a                   1 0024.3898.0f0a STALE 266 3/11  1
device(config)# show ipv neighbor ethernet 3/11
Total number of Neighbor entries: 2
Entries on interface eth 3/11 :
    IPv6 Address                              VLAN LinkLayer-Addr State  Age  Port  R
1     128::                                     1    0024.3898.0f0a *REACH 35   3/11  0
2     fe80::224:38ff:fe98:f0a                   1    0024.3898.0f0a STALE  60 3/11  1
```

Syntax: show ipv6 neighbor [*ipv6-prefix/prefix-length* | *ipv6-address* | *interface* [*port* | *number*]]

The *ipv6-prefix/prefix-length* parameters restrict the display to the entries for the specified IPv6 prefix. You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as

documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The *ipv6-address* parameter restricts the display to the entries for the specified IPv6 address. You must specify this parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *interface* parameter restricts the display to the entries for the specified Brocade device interface. For this parameter, you can specify the **Ethernet** or **VE** keywords. If you specify an Ethernet interface, you must also specify the port number associated with the interface. If you specify a VE interface, you must also specify the VE number.

This display shows the following information.

TABLE 77 IPv6 neighbor information fields

This field...	Displays...
Total number of neighbor entries	The total number of entries in the IPv6 neighbor table.
Type Codes	Shows the route type, which can be one of the following: • B - The route is learned from BGP4+. • C - The destination is directly connected to the router. • I - The route is learned from IPv6 IS-IS. • L - The route is the host address of a loopback interface that is assigned an IPv6 address. • O - The route is learned from OSPFv3. • R - The route is learned from RIPng. • S - The route is a static route.
IPv6 Address	The 128-bit IPv6 address of the neighbor.
Link-Layer Address	The 48-bit interface ID of the neighbor.
State	The current state of the neighbor. Possible states are as follows: • INCOMPLETE - Address resolution of the entry is being performed. • REACH - The forward path to the neighbor is functioning properly. • STALE - This entry has remained unused for the maximum interval. While stale, no action takes place until a packet is sent. • DELAY - This entry has remained unused for the maximum interval, and a packet was sent before another interval elapsed. • PROBE - Neighbor solicitation are transmitted until a reachability confirmation is received.
Age	Specifies the time (in seconds) for which the IPv6 neighbor is active. When the global timer 7200 seconds expires, the IPv6 neighbor entry is cleared from the neighbor table.
Port	The port on which the entry was learned. If the ND6 entry is configured without the need to include physical interface on a VE interface, then the PORT value is indicated as INV.

TABLE 77 IPv6 neighbor information fields (Continued)

This field...	Displays...
R	Determines if the neighbor is a device or host: 0 - Indicates that the neighbor is a host. 1 - Indicates that the neighbor is a device.

The following command example indicates specific static ND6 entries.

```
device#show ipv6 neighbor
Total number of Neighbor entries: 4
Entries in default VRF:
    IPv6 Address                                VLAN      LinkLayer-Addr   State
Age    Port  R
1     fe80::204:80ff:fea0:4060                    1    0004.80a0.4060      REACH
11     3/1   1
2     fe80::204:80ff:fea0:4061                    1    0004.80a0.4061      STALE
4622   3/2   1
3     99::2                                       1    0004.80a0.4060
*INCOMP           0      Inv   1
4     199::2                                      1    0004.80a0.4061      *REACH
13     3/2   1
```

Displaying the IPv6 route table

To display the IPv6 route table, enter the following command at any CLI level.

```
device# show ipv6 route
IPv6 Routing Table - 2 entries:
Type Codes - B:BGP C:Connected I:ISIS L:Local O:OSPF R:RIP S:Static
BGP  Codes - i:iBGP e:eBGP
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2
STATIC Codes - d:DHCPv6
Type IPv6 Prefix          Next Hop Router Interface   Dis/Metric Uptime  src-vrf
C    2001:db8::/64        ::              eth 1/7     0/0        45m18s  -
C    2001:db8:0:25::/64   ::              loopback 1  0/0        1h0m    -
L    2001:db8:0:25::1/128 ::              loopback 1  0/0        13m18s  -
C    2001:db8:2000::/64   ::              eth 1/13    0/0        1h0m    -
O    2001:db8:4000::1/128       fe80::202:17ff:fe6e:c41c  eth 1/13    110/1
2m42s   -
```

Syntax: show ipv6 route [*ipv6-address* | *ipv6-prefix/prefix-length* | **bgp** | **connect** | **ospf** | **rip** | **isis** | **static** | **summary** | **tags** | **nexthop** *nexthop_id* | **ref-routes**]

The *ipv6-address* parameter restricts the display to the entries for the specified IPv6 address. You must specify the *ipv6-address* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *ipv6-prefix*/*prefix-length* parameters restrict the display to the entries for the specified IPv6 prefix. You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The**bgp** keyword restricts the display to entries for BGP4+ routes.

The**connect** keyword restricts the display to entries for directly connected interface IPv6 routes.

The**isis** keyword restricts the display to entries for IPv6 IS-IS routes.

The **ospf** keyword restricts the display to entries for OSPFv3 routes.

The**rip** keyword restricts the display to entries for RIPng routes.

The**static** keyword restricts the display to entries for static IPv6 routes.

The**summary** keyword displays a summary of the prefixes and different route types.

The **tags** keyword displays the label information for the IPv6 routes.

The **nexthop** option displays the next-hop information for all next hops in the routing table or for a specific entry. The *nexthop_id* parameter is a specific nexthop entry from the next hop table.

The **ref-routes** option allows you to display IPv6 routes in the forwarding table that refer to the specified nexthop entry.

The following table lists the information displayed by the**show ipv6 route** command.

TABLE 78 IPv6 route table fields

This field...	Displays...
Number of entries	The number of entries in the IPv6 route table.
Type	The route type, which can be one of the following: • B - The route is learned from BGP4+. • C - The destination is directly connected to the device. • I - The route is learned from IPv6 IS-IS. • L - The route is the host address of a loopback interface that is assigned an ipv6 address. • O - The route is learned from OSPFv3. • R - The route is learned from RIPng. • S - The route is a static route.
OSPF Type	• i - an internal route calculated by OSPF. • 1 - An OSPF type 1 external route. • 2 - An OSPF type 2 external route. • e - an external route calculated by OSPF.
IPv6 Prefix	The destination network of the route.
Next-Hop Router	The next-hop device.
Interface	The interface through which this device sends packets to reach the route destination.
Dis/Metric	The administrative distance and metric value of the route.

To display a summary of the IPv6 route table, enter the following command at any CLI level.

```
device# show ipv6 route summary
IPv6 Routing Table - 7 entries:
  4 connected, 2 static, 0 RIP, 1 OSPF, 0 BGP
  Number of prefixes:
  /16: 1 /32: 1 /64: 3 /128: 2
```

Table 79 lists the information displayed by the **show ipv6 route summary** command.

TABLE 79 IPv6 route table summary fields

This field...	Displays...
Number of entries	The number of entries in the IPv6 route table.
Number of route types	The number of entries for each route type.
Number of prefixes	A summary of prefixes in the IPv6 route table, sorted by prefix length.

To display the label information for the IPv6 route, enter the following command.

```
device# show ipv6 route tags
IPv6 Routing Table - 4 entries:
Type Codes - B:BGP C:Connected I:ISIS L:Local O:OSPF R:RIP S:Static
BGP  Codes - i:iBGP e:eBGP
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2
Type IPv6 Prefix           Next Hop Router    Interface     Dis/Metric     Uptime
S    2001:db8:1::/64       2001:db8:1111::2   eth 1/1       1/1            1h3m
label information: 794624(IN)
Bi   2001:db8:2::/64       ::                 lsp toPE-4    200/1          30m20s
label information: 794624(OUT)
C    2001:db8:1111::/64    ::                 eth 1/1       0/0            1h4m
label information: 794624(IN)
Bi   2001:db8:2222::/64    ::                 lsp toPE-4    200/0          30m20s
label information: 794624(OUT)
```

The label information for the IPv6 route is shown in bold text in the previous output.

Table 80 describes the output parameters of the**show ipv6 route tags** command.

TABLE 80 Output parameters of the **show ipv6 route tags** command

Field	Description
Number of entries	Shows the number of entries in the IPv6 route table.
Type Codes	Shows the route type, which can be one of the following: • B - The route is learned from BGP4+. • C - The destination is directly connected to the router. • I - The route is learned from IPv6 IS-IS. • L - The route is the host address of a loopback interface that is assigned an IPv6 address.
Type Codes (continued)	• O - The route is learned from OSPFv3. • R - The route is learned from RIPng. • S - The route is a static route.
BGP Codes	Shows the BGP type, which can be one of the following: • i - An IBGP route. • e - An EBGP route.
ISIS Codes	Shows the IS-IS type, which can be one of the following: • L1 - An IS-IS level 1 route. • L2 - An IS-IS level 2 route.

TABLE 80 Output parameters of the **show ipv6 route tags** command (Continued)

Field	Description
OSPF Codes	Shows the OSPF type, which can be one of the following: • i - An internal route calculated by OSPF. • 1 - An OSPF type 1 external route. • 2 - An OSPF type 2 external route. • e - An external route calculated by OSPF.
IPv6 Prefix	Shows the destination network of the route.
Next Hop Router	Shows the address of the next hop router.
Interface	Shows the interface through which this router sends the IPv6 packets to reach the destination.
Dis/Metric	Shows the administrative distance and metric value of the IPv6 route.
Uptime	Shows the amount of time the interface has been running.
label information	Shows the label information for the IPv6 route.

Using the nexthop option

You can display nexthop information for all next hops in the routing table or for a specific entry. To display all the nexthop entries, use the **show ipv6 route nexthop** command, and then use the option to display the next hop for a specific table entry.

```
device# show ipv6 route nexthop
Total number of IPv6 nexthop entries: 261; Forwarding Use: 259
       NextHopIp        Port             RefCount      ID      Age
1      ::               eth 1/2          1/1           1       973
2      ::               drop             1/1           65536   1013
5      ::               ve 257           1/1           898     973
6      ::               ve 279           1/1           920     973
7      ::               ve 299           1/1           940     973
8     192::1            eth 1/2      255959/255960    65538   1109
...
```

Syntax: show ipv6 route nexthop *nexthop_id*

The *nexthop_id* is under the column labeled ID in the output of the **show ip route nexthop** command. In the following example, the output of the **show ip route nexthop** command is displayed for a nexthop ID 65538.

```
device# show ipv6 route nexthop 65538
       NextHopIp        Port          RefCount        ID      Age
1      192::1            eth 1/2      255950/255951   65538   1384
```

Displaying IPv6 routes with nexthop ID

By using the **nexthop** option with the **ref-routes** keyword, you can display IPv6 routes in the forwarding table that refer to a specified nexthop entry, as the following example illustrates (using nexthop ID 65538).

```
device#show ipv6 route nexthop 65538 ref-routes
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
     Destination      Gateway         Port       Cost     Type   Uptime
1    3::/64           300:300::2      eth 1/2    20/0     B      15m27s
2    4::/64           300:300::2      eth 1/2    20/0     B      18m17s
3    4:21:103::0/126  300:300::2      eth 1/2    20/0     B      15m48s
4    4:23:112::0/126  300:300::2      eth 1/2    20/0     B      19m12s
5    4:23:113::0/126  300:300::2      eth 1/2    20/0     B      19m12s
6    4:23:114::0/126  300:300::2      eth 1/2    20/0     B      19m12s
```

Syntax: show ipv6 route nexthop *nexthop_id* **ref-routes**

Description of command output fields

The following table lists the information in the **show ipv6 route** command output when you run the **show ipv6 route nexthop***nexthop_id***ref-routes** command.

This display shows the following information.

TABLE 81 show ipv6 route nexthop ref-routes information fields

This field...	Displays...
Destination	The destination network of the IPv6 route.
Gateway	The next-hop router.
Port	The port through which this device sends packets to reach the route's destination.
Cost	The route's cost.
Type	The route type, which can be one of the following: • B - The route was learned from BGP. • D - The destination is directly connected to this Brocade device. • I - The route is an ISIS route. • O - The route is an OSPF route. • R - The route was learned from RIP. • S - The route is a static route. • * - The route is a candidate default route.
Uptime	The amount of time since the route was last modified. The format of this display parameter may change depending upon the age of the route to include the seconds (s), minutes (m), hours (h), and days (d), as described in the following: • 400d - Only days (d) displayed • 20d23h - days (d) and hours (h) displayed • 14h33m - hours (h) and minutes (m) displayed • 10m59s - minutes (m) and seconds (s) displayed

Displaying IPv6 routes using the detail option

By using the **detail** option with the **show ipv6 route** command, you can display the nexthop entry and the reference count. The following command output is displayed for a nexthop ID 65538.

```
device#show ipv6 route nexthop 65538 ref-routes
```

```
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
     Destination      Gateway         Port       Cost     Type   Uptime
1    3::/64           300:300::2      eth 1/2    20/0     B      15m27s
2    4::/64           300:300::2      eth 1/2    20/0     B      18m17s
3    4:21:103::0/126  300:300::2      eth 1/2    20/0     B      15m48s
4    4:23:112::0/126  300:300::2      eth 1/2    20/0     B      19m12s
Nexthop Entry ID:65538, Paths: 1, Ref_Count:256001/256002
```

Syntax: show ipv6 route *specific-route* **detail**

Displaying local IPv6 devices

The device can function as an IPv6 host, if you configure IPv6 addresses on the interfaces but do not enable IPv6 routing using the **ipv6 unicast-routing** command.

From the IPv6 host, you can display information about IPv6 devices to which the host is connected. The host learns about the devices through their router advertisement messages. To display information about the IPv6 devices connected to an IPv6 host, enter the following command at any CLI level.

```
device# show ipv6 router
Router fe80::2e0:80ff:fe46:3431 on Ethernet 50, last update 0 min
Hops 64, Lifetime 1800 sec
Reachable time 0 msec, Retransmit time 0 msec
```

Syntax: show ipv6 router

If you configure your device to function as an IPv6 device (configure IPv6 addresses on the interfaces and enable IPv6 routing using the **ipv6 unicast-routing** command) and then enter the **show ipv6 router command** , you will receive the following output.

```
No IPv6 router in table
```

Meaningful output for this command is generated for devices configured to function as IPv6 hosts only.

This display shows the following information.

TABLE 82 IPv6 local router information fields

This field...	Displays...
Router *ipv6 address* on *interface port*	The IPv6 address for a particular interface.
Last update	The amount of elapsed time (in minutes) between the current and previous updates received from a device.
Hops	The default value that should be included in the Hop Count field of the IPv6 header for outgoing IPv6 packets. The hops value applies to the device for which you are displaying information and should be followed by IPv6 hosts attached to the device. A value of 0 indicates that the device leaves this field unspecified.
Lifetime	The amount of time (in seconds) that the device is useful as the default device.
Reachable time	The amount of time (in milliseconds) that a device assumes a neighbor is reachable after receiving a reachability confirmation. The reachable time value applies to the device for which you are displaying information and should be followed by IPv6 hosts attached to the device. A value of 0 indicates that the device leaves this field unspecified.

TABLE 82 IPv6 local router information fields (Continued)

This field...	Displays...
Retransmit time	The amount of time (in milliseconds) between retransmissions of neighbor solicitation messages. The retransmit time value applies to the device for which you are displaying information and should be followed by IPv6 hosts attached to the device. A value of 0 indicates that the device leaves this field unspecified.

Displaying IPv6 TCP information

You can display the following IPv6 TCP information:

- General information about each TCP connection on the device, including the percentage of free memory for each of the internal TCP buffers.
- Detailed information about a specified TCP connection.

To display general information about each TCP connection on the device, enter the following command at any CLI level.

```
device# show ipv6 tcp connections
Local IP address:port <-> Remote IP address:port TCP state
192.168.182.110:23 <->    192.168.8.186:4933     ESTABLISHED
192.168.182.110:8218 <->  192.168.182.106:179    ESTABLISHED
192.168.182.110:8039 <->  192.168.2.119:179      SYN-SENT
192.168.182.110:8159 <->  192.168.2.102:179      SYN-SENT
2001:db8::110:179 <->     2001:db8::106:8222       ESTABLISHED (1440)
Total 5 TCP connections
TCP MEMORY USAGE PERCENTAGE
FREE TCB = 98 percent
FREE TCP QUEUE BUFFER = 99 percent
FREE TCP SEND BUFFER = 97 percent
FREE TCP RECEIVE BUFFER = 100 percent
FREE TCP OUT OF SEQUENCE BUFFER = 100 percent
```

Syntax: show ipv6 tcp connections

This display shows the following information.

TABLE 83 General IPv6 TCP connection fields

This field...	Displays...
Local IP address:port	The IPv4 or IPv6 address and port number of the local interface over which the TCP connection occurs.
Remote IP address:port	The IPv4 or IPv6 address and port number of the remote interface over which the TCP connection occurs.

TABLE 83 General IPv6 TCP connection fields (Continued)

This field...	Displays...
TCP state	The state of the TCP connection. Possible states include the following: • LISTEN - Waiting for a connection request. • SYN-SENT - Waiting for a matching connection request after having sent a connection request. • SYN-RECEIVED - Waiting for a confirming connection request acknowledgment after having both received and sent a connection request. • ESTABLISHED - Data can be sent and received over the connection. This is the normal operational state. • FIN-WAIT-1 - Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent. • FIN-WAIT-2 - Waiting for a connection termination request from the remote TCP. • CLOSE-WAIT - Waiting for a connection termination request from the local user. • CLOSING - Waiting for a connection termination request acknowledgment from the remote TCP. • LAST-ACK - Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request). • TIME-WAIT - Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request. • CLOSED - There is no connection state.
FREE TCB = *percentage*	The percentage of free TCP control block (TCB) space.
FREE TCB QUEUE BUFFER = *percentage*	The percentage of free TCB queue buffer space.
FREE TCB SEND BUFFER = *percentage*	The percentage of free TCB send buffer space.
FREE TCB RECEIVE BUFFER = *percentage*	The percentage of free TCB receive buffer space.
FREE TCB OUT OF SEQUENCE BUFFER = *percentage*	The percentage of free TCB out of sequence buffer space.

To display detailed information about a specified TCP connection, enter a command such as the following at any CLI level.

```
device# show ipv6 tcp status 2001:db8::110 179 2001:db8::106 8222
TCP: TCB = 0x217fc300
TCP: 2001:db8::110:179 <-> 2001:db8::106:8222: state: ESTABLISHED Port: 1
  Send: initial sequence number = 242365900
  Send: first unacknowledged sequence number = 242434080
  Send: current send pointer = 242434080
  Send: next sequence number to send = 242434080
  Send: remote received window = 16384
  Send: total unacknowledged sequence number = 0
  Send: total used buffers 0
  Receive: initial incoming sequence number = 740437769
  Receive: expected incoming sequence number = 740507227
  Receive: received window = 16384
```

```
Receive: bytes in receive queue = 0
Receive: congestion window = 1459
```

Syntax: show ipv6 tcp status *local-ip-address local-port-number remote-ip-address remote-port-number*

The *local-ip-address* parameter can be the IPv4 or IPv6 address of the local interface over which the TCP connection is taking place.

The *local-port-number* parameter is the local port number over which a TCP connection is taking place.

The *remote-ip-address* parameter can be the IPv4 or IPv6 address of the remote interface over which the TCP connection is taking place.

The *remote-port-number* parameter is the local port number over which a TCP connection is taking place.

This display shows the following information.

TABLE 84 Specific IPv6 TCP connection fields

This field...	Displays...
TCB = *location*	The location of the TCB.
local-ip-address local-port-number remote-ip-address remote-port-number state port	This field provides a general summary of the following: • The local IPv4 or IPv6 address and port number. • The remote IPv4 or IPv6 address and port number. • The state of the TCP connection. For information on possible states, refer to Table 83 . • The port numbers of the local interface.
Send: initial sequence number = *number*	The initial sequence number sent by the local device.
Send: first unacknowledged sequence number = *number*	The first unacknowledged sequence number sent by the local device.
Send: current send pointer = *number*	The current send pointer.
Send: next sequence number to send = *number*	The next sequence number sent by the local device.
Send: remote received window = *number*	The size of the remote received window.
Send: total unacknowledged sequence number = *number*	The total number of unacknowledged sequence numbers sent by the local device.
Send: total used buffers *number*	The total number of buffers used by the local device in setting up the TCP connection.
Receive: initial incoming sequence number = *number*	The initial incoming sequence number received by the local device.
Receive: expected incoming sequence number = *number*	The incoming sequence number expected by the local device.

TABLE 84 Specific IPv6 TCP connection fields (Continued)

This field...	Displays...
Receive: received window = *number*	The size of the local device receive window.
Receive: bytes in receive queue = *number*	The number of bytes in the local device receive queue.
Receive: congestion window = *number*	The size of the local device receive congestion window.

Displaying IPv6 traffic statistics

To display IPv6 traffic statistics, enter the following command at any CLI level.

```
device# show ipv6 traffic
IP6 Statistics
  36947 received, 66818 sent, 0 forwarded, 36867 delivered, 0 rawout
  0 bad vers, 23 bad scope, 0 bad options, 0 too many hdr
  0 no route, 0 can't forward, 0 redirect sent, 0 source routed
  0 frag recv, 0 frag dropped, 0 frag timeout, 0 frag overflow
  0 reassembled, 0 fragmented, 0 ofragments, 0 can't frag
  0 too short, 0 too small, 11 not member
  0 no buffer, 66819 allocated, 21769 freed
  0 forward cache hit, 46 forward cache miss
ICMP6 Statistics
Received:
  0 dest unreach, 0 pkt too big, 0 time exceeded, 0 param prob
  2 echo req, 1 echo reply, 0 mem query, 0 mem report, 0 mem red
  0 router soli, 2393 router adv, 106 nei soli, 3700 nei adv, 0 redirect
  0 bad code, 0 too short, 0 bad checksum, 0 bad len
  0 reflect, 0 nd toomany opt, 0 badhopcount
Sent:
  0 dest unreach, 0 pkt too big, 0 time exceeded, 0 param prob
  1 echo req, 2 echo reply, 0 mem query, 0 mem report, 0 mem red
  0 router soli, 2423 router adv, 3754 nei soli, 102 nei adv, 0 redirect
  0 error, 0 can't send error, 0 too freq
Sent Errors:
  0 unreach no route, 0 admin, 0 beyond scope, 0 address, 0 no port
  0 source address policy, 0 reject route
  0 pkt too big, 0 time exceed transit, 0 time exceed reassembly
  0 param problem header, 0 nextheader, 0 option, 0 redirect, 0 unknown
UDP Statistics
  470 received, 7851 sent, 6 no port, 0 input errors
TCP Statistics
  57913 active opens, 0 passive opens, 57882 failed attempts
  159 active resets, 0 passive resets, 0 input errors
  565189 in segments, 618152 out segments, 171337 retransmission
```

Syntax: show ipv6 traffic

This display shows the following information.

TABLE 85 IPv6 traffic statistics fields

This field...	Displays...
IPv6 statistics	
received	The total number of IPv6 packets received by the device.
sent	The total number of IPv6 packets originated and sent by the device.

TABLE 85 IPv6 traffic statistics fields (Continued)

This field...	Displays...
forwarded	The total number of IPv6 packets received by the Brocade device and forwarded to other devices.
delivered	The total number of IPv6 packets delivered to the upper layer protocol.
rawout	This information is used by Brocade Technical Support.
bad vers	The number of IPv6 packets dropped by the device because the version number is not 6.
bad scope	The number of IPv6 packets dropped by the device because of a bad address scope.
bad options	The number of IPv6 packets dropped by the device because of bad options.
too many hdr	The number of IPv6 packets dropped by the device because the packets had too many headers.
no route	The number of IPv6 packets dropped by the device because there was no route.
can not forward	The number of IPv6 packets the device could not forward to another device.
redirect sent	This information is used by Brocade Technical Support.
source routed	The number of IPv6 source-routed packets dropped.
frag recv	The number of fragments received by the device.
frag dropped	The number of fragments dropped by the device.
frag timeout	The number of fragment timeouts that occurred.
frag overflow	The number of fragment overflows that occurred.
reassembled	The number of fragmented IPv6 packets that the device reassembled.
fragmented	The number of IPv6 packets fragmented by the device to accommodate the MTU of this device or of another device.
ofragments	The number of output fragments generated by the device.
can not frag	The number of IPv6 packets the device could not fragment.
too short	The number of IPv6 packets dropped because they are too short.
too small	The number of IPv6 packets dropped because they do not have enough data.
not member	The number of IPv6 packets dropped because the recipient is not a member of a multicast group.

TABLE 85 IPv6 traffic statistics fields (Continued)

This field...	Displays...
no buffer	The number of IPv6 packets dropped because there is no buffer available.
forward cache miss	The number of IPv6 packets received for which there is no corresponding cache entry.
ICMP6 statistics Some ICMP statistics apply to both Received and Sent, some apply to Received only, some apply to Sent only, and some apply to Sent Errors only.	
Applies to Received and Sent	
dest unreach	The number of Destination Unreachable messages sent or received by the device.
pkt too big	The number of Packet Too Big messages sent or received by the device.
time exceeded	The number of Time Exceeded messages sent or received by the device.
param prob	The number of Parameter Problem messages sent or received by the device.
echo req	The number of Echo Request messages sent or received by the device.
echo reply	The number of Echo Reply messages sent or received by the device.
mem query	The number of Group Membership Query messages sent or received by the device.
mem report	The number of Membership Report messages sent or received by the device.
mem red	The number of Membership Reduction messages sent or received by the device.
router soli	The number of Router Solicitation messages sent or received by the device.
router adv	The number of Router Advertisement messages sent or received by the device.
nei soli	The number of Neighbor Solicitation messages sent or received by the device.
nei adv	The number of Router Advertisement messages sent or received by the device.
redirect	The number of redirect messages sent or received by the device.
Applies to Received Only	
bad code	The number of Bad Code messages received by the device.
too short	The number of Too Short messages received by the device.
bad checksum	The number of Bad Checksum messages received by the Brocade device.
bad len	The number of Bad Length messages received by the device.

TABLE 85 IPv6 traffic statistics fields (Continued)

This field...	Displays...
nd toomany opt	The number of Neighbor Discovery Too Many Options messages received by the device.
badhopcount	The number of Bad Hop Count messages received by the device.
Applies to Sent Only	
error	The number of Error messages sent by the device.
can not send error	The number of times the device encountered errors in ICMP error messages.
too freq	The number of times the device has exceeded the frequency of sending error messages.
Applies to Sent Errors Only	
unreach no route	The number of Unreachable No Route errors sent by the device.
admin	The number of Admin errors sent by the device.
beyond scope	The number of Beyond Scope errors sent by the device.
address	The number of Address errors sent by the device.
no port	The number of No Port errors sent by the device.
pkt too big	The number of Packet Too Big errors sent by the device.
source address policy	The number of ICMPv6 destination unreachable messages sent with code 5 because an IPv6 packet is dropped by an Access Control policy and the IPv6 source address of a packet matches the source address filtering policy.
reject route	The number of ICMPv6 destination unreachable messages sent code 6 because an IPv6 packet is dropped due to the destination address in the packet matching a route that has been configured to drop the packet.
time exceed transit	The number of Time Exceed Transit errors sent by the device.
time exceed reassembly	The number of Time Exceed Reassembly errors sent by the device.
param problem header	The number of Parameter Problem Header errors sent by the device.
nextheader	The number of Next Header errors sent by the device.
option	The number of Option errors sent by the device.
redirect	The number of Redirect errors sent by the device.

TABLE 85 IPv6 traffic statistics fields (Continued)

This field...	Displays...
unknown	The number of Unknown errors sent by the device.
UDP statistics	
received	The number of UDP packets received by the device.
sent	The number of UDP packets sent by the device.
no port	The number of UDP packets dropped because the packet did not contain a valid UDP port number.
input errors	This information is used by Brocade Technical Support.
TCP statistics	
active opens	The number of TCP connections opened by the device by sending a TCP SYN to another device.
passive opens	The number of TCP connections opened by the device in response to connection requests (TCP SYNs) received from other devices.
failed attempts	This information is used by Brocade Technical Support.
active resets	The number of TCP connections the device reset by sending a TCP RESET message to the device at the other end of the connection.
passive resets	The number of TCP connections the device reset because the device at the other end of the connection sent a TCP RESET message.
input errors	This information is used by Brocade Technical Support.
in segments	The number of TCP segments received by the device.
out segments	The number of TCP segments sent by the device.
retransmission	The number of segments that the device retransmitted because the retransmission timer for the segment had expired before the device at the other end of the connection had acknowledged receipt of the segment.

Displaying statistics for IPv6 subnet rate limiting

Run the **show rate-limit ipv6 subnet** and **show rate-limit protocol** commands to display information about IPv6 rate limiting.

```
device# show rate-limit ipv6 subnet
  Fwd:        252                     Drop:  155 bytes
  Re-mark:    0                       Total: 407 bytes
```

Syntax: show rate-limit ipv6-subnet

Table 86 describes the fields from the output of **show rate-limit ipv6 subnet** command.

TABLE 86 Output from the **show rate-limit ipv6 subnet** command

Field	Description
Fwd	IPv6 traffic that has been forwarded after the device was started or the counter was reset due to the rate limit policy.
Drop	IPv6 traffic that has been dropped after the device was started or the counter was reset due to the rate limit policy.
Re-mark	The number of IPv6 packets whose priority has been remarked as a result of exceeding the bandwidth available in the CIR bucket for the specific rate limit policy.
Total	IPv6 traffic that has been carried on the interface after the device was started or the counter was reset due to the rate limit policy.

```
device# show rate-limit protocol
Index                0
In use               TRUE
Protocol             0 (arp)
Policy Map           abc
Index                2
In use               TRUE
Protocol             2 (ipv6 subnet)
Policy Map           abc
```

Syntax: show rate-limit protocol

Table 87 describes the fields from the output of **show rate-limit protocol** command.

TABLE 87 Output from the **show rate-limit protocol** command

Field	Description
Index	Numeric index of the protocol supported by the device.
In use	Whether the protocol is in use or not (True: In use / False: Not in use)
Protocol	Protocol name (0: arp / 2: IPv6 subnet)
Policy Map	The rate limit policy applied on this protocol.

Displaying IPv6 information for Router Advertisement Options

Run the **show ipv6** command to display IPv6 information about the newly configured DNS recursive server addresses, domain name suffixes, and the corresponding lifetime values on an IPv6 host network.

```
device# show ipv6
Global Settings
  IPv6 Router-Id: 2.2.2.1   load-sharing path: 4
  unicast-routing enabled, ipv6  allowed to run, hop-limit 64
  reverse-path-check disabled
```

```
  host drop cam limit disabled
  urpf-exclude-default disabled
  session-logging-age 5
  No Inbound Access List Set
  No Outbound Access List Set
  source-route disabled, forward-source-route disabled, icmp-redirect disabled
OSPF (default VRF): enabled
BGP: enabled, 1 active neighbor(s) configured
  ND6 RA DNS Attributes
    ipv6 nd ra-dns-server abcd:abcd:abcd::3 lifetime 122
    ipv6 nd ra-dns-server 1::1 lifetime 150
    ipv6 nd ra-dns-server abcd:abcd:abcd::2 lifetime 196
    ipv6 nd ra-dns-server abcd:abcd:abcd::1 lifetime 200
    ipv6 nd ra-domain-name brocade.com.abc.123.abbbc lifetime 102
    ipv6 nd ra-domain-name abc-011223.brocade.com lifetime 141
    ipv6 nd ra-domain-name abc.com lifetime 155
    ipv6 nd ra-domain-name abcd.com.abc.123 lifetime 200
device#
```

Syntax: show ipv6

Displaying IPv6 interface information for Router Advertisement Options

Run the **show ipv6 interface** command to display IPv6 interface information about the newly configured DNS recursive server addresses, domain name suffixes, and the corresponding lifetime values on an IPv6 host network.

```
device# show ipv6 interface ethernet 2/1
Interface Ethernet 2/1 is up, line protocol is up
  IPv6 is enabled, link-local address is fe80::224:38ff:fe90:e430 [Preferred]
  Global unicast address(es):
    7:7:7::1 [Preferred],  subnet is 7:7:7::/64
    7:7:7:: [Anycast],  subnet is 7:7:7::/64
  Joined group address(es):
    ff02::1:ff00:1
    ff02::1:ff00:0
    ff02::1:ff90:e430
    ff02::2
    ff02::1
  Port belongs to VRF: default-vrf
  MTU is 1500 bytes
  ICMP redirects are disabled
  ND DAD is enabled, number of DAD attempts: 3
  ND reachable time is 30000 Milliseconds
  ND advertised reachable time is 0 seconds
  ND retransmit interval is 1000 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND next router advertisement will be sent in 258 seconds
  ND router advertisements live for 1800 seconds
  No Inbound Access List Set
  No Outbound Access List Set
  IPv6 RPF mode: None  IPv6 RPF Log: Disabled
  RxPkts:       0                     TxPkts:   63
  RxBytes:      0                     TxBytes:  12010
  IPv6 unicast RPF drop: 0
  IPv6 unicast RPF suppressed drop: 0
  ND6 RA DNS Attributes
    ipv6 nd ra-dns-server 11::1176 lifetime 176
    ipv6 nd ra-dns-server 11::11 lifetime 200
    ipv6 nd ra-domain-name abc.com.abb lifetime 150
    ipv6 nd ra-domain-name abc-aaa.com lifetime 199
    ipv6 nd ra-domain-name abc.com lifetime 200
device#
```

Syntax: show ipv6 interface

Commands

The following commands support the features described in this chapter:

- ipv6 nd ra-dns-server on page 461
- ipv6 nd ra-domain-name on page 462

ipv6 nd ra-dns-server

Advertises the recursive Domain Name System (DNS) server address and the lifetime multiplier information to IPv6 hosts in the Router Advertisement (RA) message. The **no** form of this command disables the advertisement of the specified server address in the RA message.

Syntax **ipv6 nd ra-dns-server** *ipv6-address* [**lifetime-multiplier** *decimal*]

no ipv6 nd ra-dns-server *ipv6-address* [**lifetime-multiplier** *decimal*]

Command Default By default, the recursive DNS server address and the lifetime multiplier information is not configured.

Parameters ***ipv6-address***

Specifies the global IPv6 address of the DNS server.

lifetime-multiplier ***decimal*** **Specifies the percentage value of maximum router advertisement interval. The maximum router advertisement interval is the maximum time that can be allowed between sending unsolicited RA messages for DNS name resolution. The lifetime-multiplier** ***decimal***

value is calculated as percentage of the RA lifetime. The maximum router advertisement interval percentage range is 100 through 200% and the default value is 200%.

Modes Global configuration mode

Interface configuration mode

Usage Guidelines You can configure a maximum of four recursive DNS server addresses and corresponding lifetime multiplier values at a given instance.

NOTE
The **ipv6 nd ra-dns-server** command at the interface configuration level takes precedence over global configuration. In other words, if at least one DNS server address is configured on an interface, it will override other DNS server address configurations at the global configuration.

Examples The following examples configure the recursive DNS address for a lifetime-multiplier value of 200.

```
device(config)# ipv6 nd ra-dns-server 2001:DC8:200::3 lifetime 200
device(config-if-e10000-1/10)# ipv6 nd ra-dns-server 2001:DC8:200::3 lifetime 200
```

ipv6 nd ra-domain-name

Configures the domain name of the Domain Name System (DNS) suffix and the lifetime multiplier information to IPv6 hosts in the Router Advertisement (RA) message. The **no** form of this command disables the advertisement of the specified domain name of DNS suffix in the RA message.

Syntax **ipv6 nd ra-domain-name** *string* [**lifetime-multiplier** *decimal*]

no ipv6 nd ra-domain-name *string* [**lifetime-multiplier** *decimal*]

Parameters ***string***

Specifies the domain name of the DNS suffix.

lifetime-multiplier ***decimal***

Specifies the percentage value of maximum router advertisement interval. The maximum router advertisement interval is the maximum time that can be allowed between sending unsolicited RA messages for DNS name resolution. The **lifetime-multiplier** *decimal* **value is calculated as percentage of** the RA lifetime. **The maximum router advertisement interval percentage range is 100 through 200% and the default value is 200%.**

Modes Global configuration mode

Interface configuration mode

Usage Guidelines You can configure a maximum of four different domain names of DNS suffix and corresponding lifetime multiplier values at a given instance.

The domain name of a DNS suffix at the global configuration level is used on all IPv6 routed interfaces that do not have a domain name of DNS suffix configured on them.

NOTE
The **ipv6 nd ra-domain-name** command at the interface configuration takes precedence over global configuration. In other words, if at least one DNS server address is configured on an interface, it will override other DNS server address configurations at the global configuration.

Examples The following examples configure the domain names of a DNS suffix for a lifetime-multiplier value of 200.

```
device (config)# ipv6 nd ra-domain-name brocade.com lifetime 200
device (config-if-e10000-1/10)# ipv6 nd ra-domain-name brocade.com lifetime 200
```

History

Release	Command History
Multi-Service IronWare R05.5.00	This command was introduced.

IPv6 Addressing

- IPv6 addressing overview 463
- IPv6 stateless auto-configuration 465

IPv6 addressing overview

Table 88 displays the individual devices and the IPv6 Addressing features they support.

TABLE 88 Supported IPv6 addressing features

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
IPv6 Addressing	Yes	Yes	No	Yes	Yes	Yes	Yes
IPv6 Address Unicast	Yes	Yes	No	Yes	Yes	Yes	Yes
IPv6 Address Multicast	Yes	Yes	No	Yes	Yes	Yes	Yes
IPv6 Address Anycast	Yes	Yes	No	Yes	Yes	Yes	Yes
IPv6 Stateless auto-configuration	Yes	Yes	No	Yes	Yes	Yes	Yes
IPv6 Address in the Configuration	Yes	Yes	No	Yes	Yes	Yes	Yes

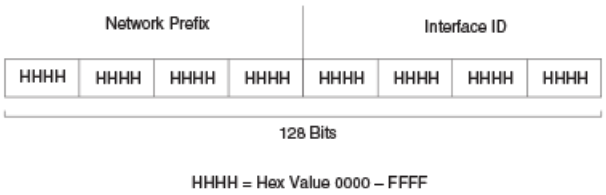
This chapter includes overview information about the following topics:

- IPv6 addressing.
- The IPv6 stateless auto-configuration feature, which enables a host on a local link to automatically configure its interfaces with new and globally unique IPv6 addresses associated with its location.

A limitation of IPv4 is its 32-bit addressing format, which is unable to satisfy potential increases in the number of users, geographical needs, and emerging applications. To address this limitation, IPv6 introduces a new 128-bit addressing format.

An IPv6 address is composed of 8 fields of 16-bit hexadecimal values separated by colons (:). Figure 33 shows the IPv6 address format.

FIGURE 33 IPv6 address format



As shown in Figure 33 , HHHH is a 16-bit hexadecimal value, while H is a 4-bit hexadecimal value. The following is an example of an IPv6 address:

2001:DB8:0000:0000:002D:D0FF:FE48:4672

Note that the sample IPv6 address includes hexadecimal fields of zeros. To make the address less cumbersome, you can do the following:

- Omit the leading zeros; for example, 2001:DB8:0:0:2D:D0FF:FE48:4672.
- Compress the successive groups of zeros at the beginning, middle, or end of an IPv6 address to two colons (::) once per address; for example, 2001:DB8::2D:D0FF:FE48:4672.

When specifying an IPv6 address in a command syntax, keep the following in mind:

- You can use the two colons (::) once in the address to represent the longest successive hexadecimal fields of zeros.
- The hexadecimal letters in the IPv6 addresses are not case-sensitive.

As shown in Figure 33 , the IPv6 network prefix is composed of the left-most bits of the address. As with an IPv4 address, you can specify the IPv6 prefix using the *prefix* or *prefix-length* format, where the following applies:

The *prefix* parameter is specified as 16-bit hexadecimal values separated by a colon.

The *prefix-length* parameter is specified as a decimal value that indicates the left-most bits of the IPv6 address.

The following is an example of an IPv6 prefix:

2001:DB8:49EA:D088::/64

IPv6 address types

As with IPv4 addresses, you can assign multiple IPv6 addresses to a Brocade device interface. Table 89 presents the three major types of IPv6 addresses that you can assign to a Brocade device interface.

A major difference between IPv4 and IPv6 addresses is that IPv6 addresses support **scope**, which describes the topology in which the address may be used as a unique identifier for an interface or set of interfaces.

Unicast and multicast addresses support scoping as follows:

- Unicast addresses support two types of scope: global scope and local scope. In turn, local scope supports link-local addresses. Table 89 describes global and link-local addresses and the topologies in which they are used.
- Multicast addresses support a scope field, which Table 89 describes.

TABLE 89 IPv6 address types

Address type	Description	Address structure
Unicast	An address for a single interface. A packet sent to a unicast address is delivered to the interface identified by the address.	Depends on the type of the unicast address: • Aggregatable global address -- An address equivalent to a global or public IPv4 address. The address structure is as follows: a fixed prefix of 2000::/3 (001), a 45-bit global routing prefix, a 16-bit subnet ID, and a 64-bit interface ID. • Unique local address -- An address used within a site or intranet. For more information on ULAs, refer to RFC 4193. • Link-local address -- An address used between directly connected nodes on a single network link. The address structure is as follows: a fixed prefix of FE80::/10 (1111 1110 10) and a 64-bit interface ID. • IPv4-compatible address -- An address used in IPv6 transition mechanisms that tunnel IPv6 packets dynamically over IPv4 infrastructures. The address embeds an IPv4 address in the low-order 32 bits and the high-order 96 bits are zeros. The address structure is as follows: 0:0:0:0:0:0:A.B.C.D. • Loopback address -- An address (0:0:0:0:0:0:0:1 or ::1) that a Brocade device can use to send an IPv6 packet to itself. You cannot assign a loopback address to a physical interface. • Unspecified address -- An address (0:0:0:0:0:0:0:0 or ::) that a node can use as a source address only until the node has its own address that is auto-configured.
Multicast	An address for a set of interfaces belonging to different nodes. Sending a packet to a multicast address results in the delivery of the packet to all interfaces in the set.	A multicast address has a fixed prefix of FF00::/8 (1111 1111). The next 4 bits define the address as a permanent or temporary address. The next 4 bits define the scope of the address (node, link, site, organization, global).
Anycast	An address for a set of interfaces belonging to different nodes. Sending a packet to an anycast address results in the delivery of the packet to the closest interface identified by the address.	An anycast address looks similar to a unicast address, because it is allocated from the unicast address space. If you assign a unicast address to multiple interfaces, it is an anycast address. An interface assigned an anycast address must be configured to recognize the address as an anycast address. An anycast address can be assigned to a router only. An anycast address must not be used as the source address of an IPv6 packet.

A Brocade device automatically configures a link-local unicast address for an interface by using the prefix of FE80::/10 (1111 1110 10) and a 64-bit interface ID. The 128-bit IPv6 address is then subjected to duplicate address detection to ensure that the address is unique on the link. If desired, you can override this automatically configured address by explicitly configuring an address.

IPv6 stateless auto-configuration

Brocade devices use the IPv6 stateless auto-configuration feature to enable a host on a local link to automatically configure its interfaces with new and globally unique IPv6 addresses associated with its location. The automatic configuration of a host interface is performed without the use of a server, such as a Dynamic Host Configuration Protocol (DHCP) server, or manual configuration.

The automatic configuration of a host interface works in the following way: a router on a local link periodically sends router advertisement messages containing network-type information, such as the 64-bit prefix of the local link and the default route, to all nodes on the link. When a host on the link receives the message, it takes the local link prefix from the message and appends a 64-bit interface ID, thereby automatically configuring its interface. (The 64-bit interface ID is derived from the MAC address of the host's NIC.) The 128-bit IPv6 address is then subjected to duplicate address detection to ensure that the address is unique on the link.

The duplicate address detection feature verifies that a unicast IPv6 address is unique before it is assigned to a host interface by the stateless auto configuration feature. Duplicate address detection uses neighbor solicitation messages to verify that a unicast IPv6 address is unique.

NOTE
For the stateless auto configuration feature to work properly, the advertised prefix length in router advertisement messages must always be 64 bits.

The IPv6 stateless auto-configuration feature can also automatically reconfigure a host's interfaces if you change the ISP for the host's network. (The host's interfaces must be renumbered with the IPv6 prefix of the new ISP.)

The renumbering occurs in the following way: a router on a local link periodically sends advertisements updated with the prefix of the new ISP to all nodes on the link. (The advertisements still contain the prefix of the old ISP.) A host can use the addresses created from the new prefix and the existing addresses created from the old prefix on the link. When you are ready for the host to use the new addresses only, you can configure the lifetime parameters appropriately using the **ipv6 nd prefix-advertisement** command. During this transition, the old prefix is removed from the router advertisements. At this point, only addresses that contain the new prefix are used on the link.

IPv6 Prefix List

- Configuring an IPv6 prefix list..... 467
- Displaying prefix list information.....468
- IPv6 ND Prefix Suppress..... 469

Table 90 displays the individual devices and the IPv6 Prefix List features they support.

TABLE 90 Supported IPv6 prefix list features

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
IPv6 Prefix List	Yes	Yes	No	Yes	Yes	Yes	Yes
Displaying Prefix List Information	Yes	Yes	No	Yes	Yes	Yes	Yes
ND6 IPv6 Prefix Suppress	Yes	Yes	No	Yes	Yes	Yes	Yes

Configuring an IPv6 prefix list

Brocade devices support IPv6 prefix lists, which you can use for basic traffic filtering. You can configure up to 100 IPv6 prefix lists.

An IPv6 prefix list is composed of one or more conditional statements that pose an action (permit or deny) if a packet matches a specified prefix. In prefix lists with multiple statements, you can specify a sequence number for each statement. The specified sequence number determines the order in which the statement appears in the prefix.

You can configure an IPv6 prefix list on a global basis, then use it as input to other commands or processes, such as route aggregation, route redistribution, route distribution, route maps, and so on. When a Brocade device interface sends or receives an IPv6 packet, it applies the statements within the IPv6 prefix list in their order of appearance to the packet. As soon as a match occurs, the device takes the specified action (permit or deny the packet) and stops further comparison for that packet.

You can use permit statements in the prefix list to specify the traffic that you want to send to the other feature. If you use deny statements, the traffic specified by the deny statements is not supplied to the other feature.

To configure an IPv6 prefix list and use it as input to the RIPng **distribute-list** command, enter commands such as the following.

```
device(config)# ipv6 prefix-list routesfor2001 permit 2001:db8::/32
device(config)# ipv6 router rip
device(config-ripng-router)# distribute-list prefix-list routesfor2001 out ethernet
3/1
```

These commands permit the inclusion of routes with the IPv6 prefix 2001:db8::/32 in RIPng routing updates sent from Ethernet interface 3/1.

Syntax: [no] ipv6 prefix-list *name* [**seq** *sequence-number*] **deny** *ipv6-prefix/prefix-length* | **permit** *ipv6-prefix/prefix-length* | **description** *string* [**ge** *ge-value*] [**le** *le-value*]

The *name* parameter specifies the prefix list name. You use this name when using the prefix list as input to command or route map.

The **seq***seq-number* parameter is optional and specifies the IPv6 prefix list's sequence number. If you do not specify a sequence number, the software numbers them in increments of 5, beginning with prefix list entry 5. The Brocade device interprets the prefix list entries in numerical order, beginning with the lowest sequence number.

The **description***string* parameter is a text string describing the prefix list.

The **deny***ipv6-prefix*/*prefix-length* | **permit***ipv6-prefix*/*prefix-length* parameters specify the action the Brocade device takes if a packet contains a route specified in this prefix list.

You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The prefix list matches only on the specified prefix/prefix length unless you use the **ge***prefix-length* or **le***prefix-length* parameters. (See below.)

You can specify a range of prefix lengths for prefixes that are more specific than *ipv6-prefix*/*prefix-length*.

- If you specify only **ge***ge-value*, then the range is from *ge-value* to 128.
- If you specify only **le***le-value*, then the range is from *le-value* to the *prefix-length* parameter.

The *ge-value* or *le-value* you specify must meet the following condition.

prefix-length < **ge-value** <= *l***e-value** <= 128

If you do not specify **ge***ge-value* or **le***le-value*, the prefix list matches only on the exact prefix you specify with the *ipv6-prefix*/*prefix-length* parameter.

To delete the prefix list entry, use the **no** form of this command.

Displaying prefix list information

To display the IPv6 prefix lists configured on a Brocade device, enter the following command at any level of the CLI.

```
device(config)# show ipv6 prefix-lists
ipv6 prefix-list routesfor2001: 1 entries
     seq 5 permit 2001:db8::/32
```

Syntax: show ipv6 prefix-lists [**name**]

The *name* parameter restricts the display to the specified prefix list. Specify the name of the prefix list that you want to display.

IPv6 ND Prefix Suppress

Brocade devices support IPv6 ND Prefix Suppress, which is useful in an LAN where multiple hosts are connected to router(s). Prefix Suppress performs these functions:

- Advertisement of on-link prefix information is suppressed in router advertisement (RA) messages.
- Hosts are prevented from auto configuring based on the prefix in the RA message.
- DHCPv6 is used for security and accountability.
- Advertisement of identical prefixes by multiple routers is suppressed.
- Global Suppress option suppresses IPv6 addresses defined on an interface from getting advertised in the RA message.

NOTE
When the user configures the Global Suppress option, an RA is generated with all deprecated IPv6 address entries that are not advertised in subsequent RA messages.

- Prefix advertisement entry in the RA message is advertised if a duplicate entry exists in the prefix advertisement list and IPv6 address list

Configuring the suppress option to specific IPv6 addresses defined on an interface generates deprecated IPv6 address entries (i.e. with `preferred lifetime = 0 hours` and `valid lifetime = 2 hours`) in RA messages. When a host receives deprecated IPv6 address entries, the address is forbidden for new sessions although existing sessions can continue using the address.

Deprecated entries are advertised in the following scenarios:

- Suppress option is configured for IPv6 address entries.
- Prefix advertisement entries are un-configured.
- IPv6 address entries are un-configured.

Configuring IPv6 Prefix Suppress

Command syntax for configuring the suppress option for an IPv6 address entry:

```
(config-if-x)#[no] ipv6 nd address <ipv6-address> suppress
```

Command syntax for configuring the suppress option for all IPv6 address entries:

```
(config-if-x)#[no] ipv6 nd address suppress
```

Command syntax for Show IPv6 interface output is modified to display individual or globally suppressed entries:

```
Router-A# show ipv6 interface ethernet 1/2
  MTU is 1500 bytes
  ICMP redirects are disabled
  ND DAD is enabled, number of DAD attempts: 3
  ND reachable time is 30000 Milliseconds
  ND advertised reachable time is 0 seconds
  ND retransmit interval is 1000 milliseconds
```

```
  ND advertised retransmit interval is 0 milliseconds
  ND next router advertisement will be sent in 2 seconds
  ND router advertisements live for 1800 seconds
  ND suppress-ra disabled
  ND address-prefixes suppressed in router advertisement - all
  ND address-prefixes suppressed in router advertisement -
  300::1/64
Router-A#
```

Command syntax for debugging IPv6 Prefix Suppress:

```
Router-A# debug ipv6 ra
```

NOTE
No additional debug commands are added for this feature. Debug commands available for IPv6 ND can be used for this feature.

Configuration Considerations for IPv6 Prefix Suppress

The following considerations should be considered prior to configuring IPv6 Prefix Suppress:

- - Suppress option is not configurable for a non-existent IPv6 address entry. As a result, the suppress option is not applicable to future references.
  - Suppress option is not supported for suppressing prefix advertisement entries.

NOTE
The user may un-configure the prefix advertisement entry so it is not advertised in the RA message.

- - Configuration of the suppress option is not allowed for a duplicate entry in any combination.
  - When multiple IPv6 addresses of the same subnet are defined on an interface, apply the suppress option on individual entries.

IPv6 ND Router Advertisement Control

IPv6 ND Router Advertisement Control allows for disabling sending out router advertisements at the interface level. The **no ipv6 nd suppress-ra** command at the interface level allows the user to disable and enable the sending of the ND Router Advertisement on an interface. By default, the sending of ND Router Advertisement (RA) is enabled on all interfaces, except for the tunnel and loopback interfaces, providing that the IPv6 Unicast Routing is enabled and the interfaces are active for IPv6.

The IPv6 ND Router Advertisement Control gives the ability to quickly turn off the sending of IPv6 ND Router Advertisement message on an IPv6 enabled interfaces.

By default,

- The ND Router Advertisement is enabled.
- Interface is enabled to send ND Router Advertisements.
- The **ipv6 nd suppress-ra** and **ipv6 nd send-ra** interface commands, when configured, override the system and VRF global **ipv6 nd global-suppress-ra** command.

Users sometimes require the ability to quickly turn off the sending of IPv6 ND Router Advertisement message on an IPv6 enabled interfaces. This is achieved by providing the following additional configuration command at interface level:

```
device(config-if-e10000-1/1)#no ipv6 nd suppress-ra
```

The **ipv6 nd send-ra** command is a new interface level command added as part of this enhancement. This allows the user to configure the sending of RA messages on some selected interfaces when the **ipv6 nd global-suppress-ra** command is set to disable the sending of RA messages on all other interfaces.

Syntax: [no]ipv6 nd suppress-ra

Managing a Device Over IPv6

- Using the IPv6 copy command..... 473
- Using the IPv6 ncopy command..... 475
- Using the IPv6 ping command..... 477
- Using the IPv6 traceroute command.....478
- Using Telnet.....478
- Using Secure Shell.....480

Table 91 displays the individual Brocade devices and the supported features on how to Manage a Device Over IPv6.

TABLE 91 Supported Brocade IPv6 routes features

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
IPv6 copy Command	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Copying a File from an IPv6 TFTP Server	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Using the IPv6 ncopy Command	Yes	Yes	Yes	Yes	Yes	Yes	Yes
IPv6 Ping Command	Yes	Yes	Yes	Yes	Yes	Yes	Yes
IPv6 Traceroute Command	Yes	Yes	Yes	Yes	Yes	Yes	Yes
IPv6 Telnet	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Secure Shell	Yes	Yes	Yes	Yes	Yes	Yes	Yes

You can perform system management tasks for the device using the **copy** , **ncopy** , **ping** , **telnet** , and **traceroute** commands and Secure Shell (SSH). These commands and SSH now function over IPv6.

This section describes the IPv6-related syntax added to the commands and SSH. It does not describe the already existing command syntax for IPv4.

Using the IPv6 copy command

The **copy** command for IPv6 allows you to do the following:

- Copy a file from a specified source to an IPv6 TFTP server.
- Copy a file from an IPv6 TFTP server to a specified destination.

Copying a file to an IPv6 TFTP server

You can copy a file from the following sources to an IPv6 TFTP server:

- Flash memory.
- Running configuration.
- Startup configuration.

Copying a file from flash memory

For example, to copy the primary or secondary boot image from the device's flash memory to an IPv6 TFTP server, enter a command such as the following.

```
device# copy flash tftp ipv6 2001:db8:e0ff:7837::3 test.img secondary
```

This command copies the secondary boot image named test.img from flash memory to a TFTP server with the IPv6 address of 2001:db8:e0ff:7837::3.

Syntax: copy flash tftp ipv6 source-file-name primary | **secondary**

The *ipv6-address* parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *source-file-name* parameter specifies the name of the file you want to copy to the IPv6 TFTP server.

The **primary** keyword specifies the primary boot image, while the **secondary** keyword specifies the secondary boot image.

Copying a file from the running or startup configuration

For example, to copy the running configuration to an IPv6 TFTP server, enter a command such as the following.

```
device# copy running-config tftp ipv6 2001:db8:e0ff:7837::3 newrun.cfg
```

This command copies the running configuration to a TFTP server with the IPv6 address of 2001:db8:e0ff:7837::3 and names the file on the TFTP server newrun.cfg.

Syntax: copy running-config | **startup-config tftp ipv6 destination-file-name**

Specify the **running-config** keyword to copy the running configuration file to the specified IPv6 TFTP server.

Specify the **startup-config** keyword to copy the startup configuration file to the specified IPv6 TFTP server.

The tftp *ipv6-address* parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *destination-file-name* parameter specifies the name of the file that is copied to the IPv6 TFTP server.

Copying a file from an IPv6 TFTP server

You can copy a file from an IPv6 TFTP server to the following destinations:

- Flash memory.
- Running configuration.
- Startup configuration.

Copying a file to flash memory

For example, to copy a boot image from an IPv6 TFTP server to the primary or secondary storage location in the device's flash memory, enter a command such as the following.

```
device# copy tftp flash ipv6 2001:db8:e0ff:7837::3 test.img secondary
```

This command copies an application image named test.img from an IPv6 TFTP server with the IPv6 address of 2001:db8:e0ff:7837::3 to the secondary storage location in the device's flash memory.

Syntax: copy tftp flash ipv6 source-file-name primary | **secondary**

The *ipv6-address* parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *source-file-name* parameter specifies the name of the file you want to copy from the IPv6 TFTP server.

The **primary** keyword specifies the primary storage location in the device's flash memory, while the **secondary** keyword specifies the secondary storage location in the device's flash memory.

Copying a file to the running or startup configuration

For example, to copy a configuration file from an IPv6 TFTP server to the router's running or startup configuration, enter a command such as the following.

```
device# copy tftp running-config ipv6 2001:db8:e0ff:7837::3 newrun.cfg overwrite
```

This command copies the newrun.cfg file from the IPv6 TFTP server and overwrites the router's running configuration file with the contents of newrun.cfg.

NOTE
To activate this configuration, you must reload (reset) the device.

Syntax: copy tftp running-config | **startup-config ipv6 source-file-name** [**overwrite**]

Specify the **running-config** keyword to copy the running configuration from the specified IPv6 TFTP server.

Specify the **startup-config** keyword to copy the startup configuration from the specified IPv6 TFTP server.

The *ipv6-address* parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *source-file-name* parameter specifies the name of the file that is copied from the IPv6 TFTP server.

The **overwrite** keyword specifies that the device should overwrite the current configuration file with the copied file. If you do not specify this parameter, the device copies the file into the current running or startup configuration but does not overwrite the current configuration.

NOTE
You cannot use the overwrite option from non-console sessions, because it will disconnect the session.

When a configuration file is loaded using the `copy tftp running-config` command, the following commands within the configuration file are supported.

- isis metric command - refer to Copying a file to the running or startup configuration.
- set-overload-bit command - refer to Copying a file to the running or startup configuration on page 255 .
- admin-group - refer to the *Multi-Service IronWare Multiprotocol Label Switch (MPLS) Configuration Guide*
- cspf-group - refer to the *Multi-Service IronWare Multiprotocol Label Switch (MPLS) Configuration Guide*
- bypass-lsp - refer to the *Multi-Service IronWare Multiprotocol Label Switch (MPLS) Configuration Guide*

Using the IPv6 ncopy command

The **ncopy** command for IPv6 allows you to do the following:

- Copy a primary or secondary boot image from flash memory to an IPv6 TFTP server.
- Copy the running configuration to an IPv6 TFTP server.
- Copy the startup configuration to an IPv6 TFTP server
- Upload various files from an IPv6 TFTP server.

Copying a primary or secondary boot image from flash memory to an IPv6 TFTP server

For example, to copy the primary or secondary boot image from the device's flash memory to an IPv6 TFTP server, enter a command such as the following.

```
device# ncopy flash primary tftp ipv6 2001:db8:e0ff:7837::3 primary.img
```

This command copies the primary boot image named primary.img from flash memory to a TFTP server with the IPv6 address of 2001:db8:e0ff:7837::3.

Syntax: ncopy flash primary | secondary tftp ipv6 source-file-name

The **primary** keyword specifies the primary boot image, while the **secondary** keyword specifies the secondary boot image.

The **tftp***ipv6-address* parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *source-file-name* parameter specifies the name of the file you want to copy from flash memory.

Copying the running or startup configuration to an IPv6 TFTP server

For example, to copy a device's running or startup configuration to an IPv6 TFTP server, enter a command such as the following.

```
device# ncopy running-config tftp ipv6 2001:db8:e0ff:7837::3 bakrun.cfg
```

This command copies a device's running configuration to a TFTP server with the IPv6 address of 2001:db8:e0ff:7837::3 and names the destination file bakrun.cfg.

Syntax: ncopy running-config | **startup-config tftp ipv6 destination-file-name**

Specify the **running-config** keyword to copy the device's running configuration or the **startup-config** keyword to copy the device's startup configuration.

The **tftp***ipv6-address* parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *destination-file-name* parameter specifies the name of the running configuration that is copied to the IPv6 TFTP server.

Uploading files from an IPv6 TFTP server

You can upload the following files from an IPv6 TFTP server:

- Primary boot image.
- Secondary boot image.
- Running configuration.
- Startup configuration.

Uploading a primary or secondary boot image from an IPv6 TFTP server

For example, to upload a primary or secondary boot image from an IPv6 TFTP server to a device's flash memory, enter a command such as the following.

```
device# ncopy tftp ipv6 2001:db8:e0ff:7837::3 primary.img flash primary
```

This command uploads the primary boot image named primary.img from a TFTP server with the IPv6 address of 2001:db8:e0ff:7837::3 to the device's primary storage location in flash memory.

Syntax: ncopy tftp ipv6 source-file-name flash primary | **secondary**

The **tftp***ipv6-address* parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *source-file-name* parameter specifies the name of the file you want to copy from the TFTP server.

The **primary** keyword specifies the primary location in flash memory, while the **secondary** keyword specifies the secondary location in flash memory.

Uploading a running or startup configuration from an IPv6 TFTP server

For example to upload a running or startup configuration from an IPv6 TFTP server to a device, enter a command such as the following.

```
device# ncopy tftp ipv6 2001:db8:e0ff:7837::3 newrun.cfg running-config
```

This command uploads a file named newrun.cfg from a TFTP server with the IPv6 address of 2001:db8:e0ff:7837::3 to the device.

Syntax: ncopy tftp ipv6 source-file-name running-config | **startup-config**

The **tftp***ipv6-address* parameter specifies the address of the TFTP server. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *source-file-name* parameter specifies the name of the file you want to copy from the TFTP server.

Specify the **running-config** keyword to upload the specified file from the IPv6 TFTP server to the device. The device copies the specified file into the current running configuration but does not overwrite the current configuration.

Specify the **startup-config** keyword to upload the specified file from the IPv6 TFTP server to the device. The device copies the specified file into the current startup configuration but does not overwrite the current configuration.

Using the IPv6 ping command

The **ping** command allows you to verify the connectivity from a device to an IPv6 device by performing an ICMP for IPv6 echo test.

For example, to ping a device with the IPv6 address of 2001:db8:847f:a385:34dd::45 from the device, enter the following command.

```
device# ping ipv6 2001:db8:847f:a385:34dd::45
```

Syntax: ping ipv6 ipv6-address [**outgoing-interface** [**port** | **ve number**]] [**source ipv6-address**] [**count number**] [**timeout milliseconds**] [**ttl number**] [**size bytes**] [**quiet**] [**numeric**] [**no-fragment**] [**verify**] [**data** *1-to-4 bytehex*] [**brief**]

The *ipv6-address* parameter specifies the address of the router. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **outgoing-interface** keyword specifies a physical interface over which you can verify connectivity. If you specify a physical interface, such as an Ethernet interface, you must also specify the port number of the interface. If you specify a virtual interface, such as a VE, you must specify the number associated with the VE.

The **source***ipv6-address* parameter specifies an IPv6 address to be used as the origin of the ping packets.

The**count***number* parameter specifies how many ping packets the router sends. You can specify from 1 - 4294967296. The default is 1.

The **timeout***milliseconds* parameter specifies how many milliseconds the router waits for a reply from the pinged device. You can specify a timeout from 1 - 4294967296 milliseconds. The default is 5000 (5 seconds).

The **ttl** *number* parameter specifies the maximum number of hops. You can specify a TTL from 1 - 255. The default is 64.

The**size***bytes* parameter specifies the size of the ICMP data portion of the packet. This is the payload and does not include the header. You can specify from 0 - 4000. The default is 16.

The **no-fragment** keyword turns on the "don't fragment" bit in the IPv6 header of the ping packet. This option is disabled by default.

The **quiet** keyword hides informational messages such as a summary of the ping parameters sent to the device and instead only displays messages indicating the success or failure of the ping. This option is disabled by default.

The **verify** keyword verifies that the data in the echo packet (the reply packet) is the same as the data in the echo request (the ping). By default the device does not verify the data.

The **data***1 - 4 byte hex* parameter lets you specify a specific data pattern for the payload instead of the default data pattern, "abcd", in the packet's data payload. The pattern repeats itself throughout the ICMP message (payload) portion of the packet.

NOTE
For parameters that require a numeric value, the CLI does not check that the value you enter is within the allowed range. Instead, if you do exceed the range for a numeric value, the software rounds the value to the nearest valid value.

The**brief** keyword causes ping test characters to be displayed. The following ping test characters are supported:

! Indicates that a reply was received.

. Indicates that the network server timed out while waiting for a reply.

U Indicates that a destination unreachable error PDU was received.

I Indicates that the user interrupted ping.

Using the IPv6 traceroute command

The **traceroute** command allows you to trace a path from the device to an IPv6 host.

The CLI displays trace route information for each hop as soon as the information is received. Traceroute requests display all responses to a minimum TTL of 1 second and a maximum TTL of 30 seconds. In addition, if there are multiple equal-cost routes to the destination, the device displays up to three responses.

For example, to trace the path from the device to a host with an IPv6 address of 2001:db8:349e:a384::34, enter the following command.

```
device# traceroute ipv6 2001:db8:349e:a384::34
```

Syntax: traceroute ipv6 ipv6-address

The *ipv6-address* parameter specifies the address of a host. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

Using Telnet

This section explains how to do the following:

- Use the **telnet** command to establish a Telnet session from the device to a remote IPv6 host.
- Establish a Telnet session from a remote IPv6 host to the device.

Using the IPv6 Telnet command

The **telnet** command allows a Telnet connection from a device to a remote IPv6 host using the console. Up to five read-access and one write-access inbound Telnet session are supported on the router at one time. Up to five simultaneous outbound Telnet sessions can also be supported from the console session, from inbound Telnet sessions, from inbound SSH sessions or from a Web session.

To see the Telnet sessions currently open on the device, enter the **show telnet** command; to see both the open Telnet and open SSH sessions, enter the **show who** command as shown below.

```
device# show who
Console connections:
       established
       3 days 17 hours 31 minutes 27 seconds in idle
Telnet server status: Enabled
Telnet connections (inbound):
 1      established, client ip address 10.53.1.86
        you are connecting to this session
        1 seconds in idle
 2      established, client ip address 10.53.1.86
        7 seconds in idle
 3      closed
 4      closed
 5      closed
Telnet connections (outbound):
 6      established, server ip address 10.47.2.200, from Telnet session 1
        4 seconds in idle
 7      closed
 8      closed
 9      closed
 10     closed
SSH server status: Enabled
SSH connections:
 1      closed
 2      closed
 3      closed
 4      closed
...
```

Syntax: show who

To establish a Telnet connection to a remote host, use the **telnet** command. The following example will establish an outbound Telnet connection to a remote host with the IPv6 address of 2001:db8:3de2:c37::6.

```
device# telnet 2001:db8:3de2:c37::6
```

Syntax: telnet ipv6-address [**port-number** | **outgoing-interface ethernet port** | **ve number**]

The **ipv6-address** parameter specifies the address of a remote host. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *port-number* parameter specifies the port number on which the device establishes the Telnet connection. You can specify a value between 1 - 65535. If you do not specify a port number, the device establishes the Telnet connection on port 23.

If the IPv6 address you specify for the **telnet***ipv6-address* command is a link-local address, you must specify the **outgoing-interface** ethernet *port* | *ve number* parameter. This parameter specifies the interface that must be used to reach the remote host. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE interface, also specify the VE number.

Establishing a Telnet session from an IPv6 host

To establish a Telnet session from an IPv6 host to the device, open your Telnet application and specify the IPv6 address of the router.

Using Secure Shell

Secure Shell (SSH) is a mechanism that allows secure remote access to management functions on the device. SSH provides a function similar to Telnet. You can log into and configure the device using a publicly or commercially available SSH client program, just as you can with Telnet. However, unlike Telnet, which provides no security, SSH provides a secure, encrypted connection to the device.

To open an SSH session from an IPv6 host running an SSH client program to the device, open your SSH client program and specify the IPv6 address of the router.

Static IPv6 Routes

- Configuring a static IPv6 route.....481
- Configuring a IPv6 static multicast route.....483
- BFD for Static Routes.....484

Table 92 displays the individual Brocade devices and the IPv6 Routes features they support.

TABLE 92 Supported Brocade IPv6 routes features

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
Static IPv6 Route	Yes	Yes	No	Yes	Yes	Yes	Yes
IPv6 Static Multicast Route	Yes	Yes	No	Yes	Yes	Yes	Yes
BFD for Static Routes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

This chapter describes how to configure a static IPv6 route. A **static IPv6 route** is a manually configured route, which creates a path between two IPv6 devices. A static IPv6 route is similar to a static IPv4 route. Static IPv6 routes have their advantages and disadvantages; for example, a static IPv6 route does not generate updates, which reduces processing time for an IPv6 router. Conversely, if a static IPv6 route fails or if you want to change your network topology, you might need to manually reconfigure the static IPv6 route.

Configuring a static IPv6 route

You can configure a static IPv6 route to be redistributed into a routing protocol, but you cannot redistribute routes learned by a routing protocol into the static IPv6 routing table.

Before configuring a static IPv6 route, you must enable the forwarding of IPv6 traffic on the Brocade device using the **ipv6 unicast-routing** command and enable IPv6 on at least one interface by configuring an IPv6 address or explicitly enabling IPv6 on that interface.

To configure a static IPv6 route for a destination network with the prefix 2001:db8::0/32, a next-hop gateway with the global address 2001:db8:0:ee44::1, and an administrative distance of 110, enter the following command.

```
device(config)# ipv6 route 2001:db8::0/32 2001:db8:0:ee44::1 distance 110
```

Syntax: [no] ipv6 route dest-ipv6-prefix/prefix-length next-hop-ipv6-address [**metric**] [**distance number**]

To configure a static IPv6 route for a destination network with the prefix 2001:db8::0/32 and a next-hop gateway with the link-local address fe80::1 that the Brocade device can access through Ethernet interface 3/1, enter the following command.

```
device(config)# ipv6 route 2001:db8::0/32 ethernet 1 fe80::1
```

Syntax: [no] ipv6 route dest-ipv6-prefix/prefix-length [**ethernet slot/port** | **ve num** | **null0**] **next-hop-ipv6-address** [**metric**] [**tag** *num*] [**distance number**]

To configure a static IPv6 route for a destination network with the prefix 2001:db8::0/32 and a next-hop gateway that the Brocade device can access through tunnel 1, enter the following command.

```
device(config)# ipv6 route 2001:db8::0/32 tunnel 1
```

Syntax: [no] ipv6 route dest-ipv6-prefix/prefix-length interface port [**metric**] [**distance number**]

Table 93 describes the parameters associated with this command and indicates the status of each parameter.

TABLE 93 Static IPv6 route parameters

Parameter	Configuration details	Status
The IPv6 prefix and prefix length of the route's destination network.	You must specify the **dest-ipv6-prefix** parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the **prefix-length** parameter as a decimal value. A slash mark (/) must follow the **ipv6-prefix** parameter and precede the **prefix-length** parameter.	Mandatory for all static IPv6 routes.
The route's next-hop gateway, which can be one of the following: • The IPv6 address of a next-hop gateway. • A tunnel interface.	You can specify the next-hop gateway as one of the following types of IPv6 addresses: • A global address. • A link-local address. If you specify a global address, you do not need to specify any additional parameters for the next-hop gateway. If you specify a link-local address, you must also specify the interface through which to access the address. You can specify one of the following interfaces: • An Ethernet interface. • A tunnel interface. • A virtual interface (VE). If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE or tunnel interface, also specify the VE or tunnel number. You can also specify the next-hop gateway as a tunnel interface. If you specify a tunnel interface, also specify the tunnel number.	Mandatory for all static IPv6 routes.

TABLE 93 Static IPv6 route parameters (Continued)

Parameter	Configuration details	Status
The route's metric.	You can specify a value from 1 - 16.	Optional for all static IPv6 routes. (The default metric is 1.)
Tag *number*	This parameter specifies the tag value of the route.	The possible values are 0 - 4294967295. The default is 0.
The route's administrative distance.	You must specify the **distance** keyword and any numerical value.	Optional for all static IPv6 routes. (The default administrative distance is 1.)

A metric is a value that the Brocade device uses when comparing this route to other static routes in the IPv6 static route table that have the same destination. The metric applies only to routes that the device has already placed in the IPv6 static route table.

The administrative distance is a value that the Brocade device uses to compare this route with routes from other route sources that have the same destination. (The device performs this comparison before placing a route in the IPv6 route table.) This parameter does not apply to routes that are already in the IPv6 route table. In general, a low administrative distance indicates a preferred route. By default, static routes take precedence over routes learned by routing protocols. If you want a dynamic route to be chosen over a static route, you can configure the static route with a higher administrative distance than the dynamic route.

Configuring a IPv6 static multicast route

IPv6 multicast routes allow you to control the network path used by multicast traffic. Static multicast routes are especially useful when the unicast and multicast topologies of a network are different. You can avoid the need to make the topologies similar by instead configuring static multicast routes.

You can configure more than one static IPv6 multicast route. The Brocade device by default uses the most specific route that matches a multicast source address. You can also specify route preference using the **route-preference** command as described in the *Configuring IP Multicast Protocols* chapter in the *Multi-Service IronWare IP Multicast Configuration Guide* . Thus, if you want to configure a multicast static route for a specific multicast source and also configure another multicast static route for all other sources, you can configure two static routes.

To configure a IPv6 mroute for a destination network with the prefix 2001:db8::0/32, a next-hop gateway with the global address 2001:db8:0:ee44::1, and an administrative distance of 110, enter the following command.

```
device(config)# ipv6 mroute 2001:db8::0/32 2001:db8:0:ee44::1 distance 110
```

Syntax: [no] ipv6 mroute dest-ipv6-prefix/prefix-length next-hop-ipv6-address next-hop-enable-default next-hop-recursion [**metric**] [**distance number**] [**tag number**]

Syntax: [no] ipv6 mroute ipv6-addr interface ethernet slot/portnum | **ve num** | **tunnel num** [**distance num**] [**tag number**]

The `ipv6-addr` command specifies the next-hop IP address.

NOTE
In IPv6 multicasting, a route is handled in terms of its source, rather than its destination.

You can use the **ethernet***slot/portnum* parameter to specify a physical port or the **ve***num* parameter to specify a virtual interface.

NOTE
The **ethernet***slot/portnum* parameter does not apply to PIM SM.

The **next-hop-enable-default** parameter sets the default route to resolve the static route nexthop.

The**next-hop-recursion** parameter sets the static route to resolve the static route nexthop.

The **distance***num* parameter sets the administrative distance for the route. When comparing multiple paths for a route, the Brocade device prefers the path with the lower administrative distance.

NOTE
Regardless of the administrative distances, the Brocade device always prefers directly connected routes over other routes.

BFD for Static Routes

BFD enables prompt detection of faults in the bidirectional path between two BFD peers. BFD for Static Routes allows you to detect failures that impact the forwarding path of a static route. This feature supports both single hop and multi-hop BFD Static Routes for both IPv4 and IPv6.

Static routes and BFD neighbors are configured separately. A static route is automatically associated with a static BFD neighbor if the static route's next-hop exactly matches the neighbor address of the static BFD neighbor and the BFD monitoring is enabled for the static route.

When a static BFD neighbor is configured, BFD asks the routing table manager (RTM) if there is a route to the neighbor. If a route exists, and if the route is directly connected, then BFD initiates a single hop session. If the route is not directly connected, BFD establishes a multi-hop session. Once the session comes up, BFD adds the corresponding static routes to RTM. If no route exists, then BFD will not add the corresponding static routes to RTM.

When the BFD session goes down because the BFD neighbor is no longer reachable, static routes monitored by BFD are removed from the RTM. The removed routes can be added back if the BFD neighbor becomes reachable again. Single-hop BFD sessions use the BFD timeout values configured on the outgoing interface. Timeout values of multi-hop BFD sessions are specified along with each BFD neighbor. Multiple static routes going to the same BFD neighbor use the same BFD session and timeout values.

Configuration considerations

- In a multi-hop session, the protocol must be stated in the command IProute next-hop *protocol*.
- BFD multi-hop is supported for a nexthop resolved through OSPF, BGP, ISIS, RIP, and MPLS.
- BFD multi-hop is not supported for a nexthop resolved through Default Route.
- BFD for static routes is not supported for static routes with an LSP name as nexthop.

- Upon reboot, the router will first bring up the static BFD sessions, and then install the static routes in the routing table manager (RTM). There may be a delay of 90 seconds before the BFD sessions become available.
- BFD is not supported in MCT.
- BFD for static routes will not support interface-based static routes for both IPv4 and IPv6.

Configuring BFD for static routes

The following example assumes the configured interface Ethernet 1/1 is as follows:

```
interface ethernet 1/1
  bfd interval 100 min-rx 100 multiplier 5
  ip address 10.0.0.1/24
```

Single hop configuration

To configure BFD for static routes, configure BFD neighbors using the following commands. BFD neighbors can be configured in unassociated mode using this command.

The following example uses interface ethernet 1/1 as the outgoing interface and uses the BFD intervals on ethernet 1/1. The next hop address 10.0.0.5 is the BFD neighbor and the configured address 10.0.0.1 on Ethernet 1/1 is the local address.

```
device(config)#ip route static-bfd 10.0.0.5 10.0.0.1
```

Syntax to configure BFD Static neighbor for IPv4:

Syntax: [no] ip route [**vrf** *vrf-name*] **static-bfd** *neighbor-ip-address local-ip-address* **interval** *tx-rate min-rxrx-rate* **multiplier value**

Syntax to configure BFD Static neighbor for IPv6:

Syntax: [no] ipv6 route [**vrf** *vrf-name*] **static-bfd** *neighbor-ipv6-address local-ipv6-address* **interval** *tx-rate min-rx rx-rate* **multiplier** *value*

The **no** version of the command removes the BFD monitoring by removing the BFD static neighbor 10.0.0.5 and eliminating the BFD session, while keeping the static route in the RTM, and retaining the existing traffic to IP route 20.0.0.0. You only need to specify the BFD neighbor address and the local address when removing a BFD neighbor.

To enable BFD for static routes use the following command. The BFD parameter allows you to enable BFD monitoring for the static route.

```
device(config)#ip route 20.0.0.0/24 10.0.0.5 bfd
```

Syntax to enable BFD monitoring for IPv4:

Syntax: [no] ip route *Destination IPaddress Next hop Router IPaddress* **... bfd**

Syntax to enable BFD monitoring for IPv6:

Syntax: [no] ipv6 route *Destination IPv6address Next hop Router IPv6address* **... bfd**

The **no** version of the command removes BFD monitoring from the static route.

Multi-hop configuration

The following example shows a multi-hop configuration using the commands explained in the single hop section.

```
device(config)#ip route static-bfd 30.0.0.5 10.0.0.1 interval 90 min-rx 90
multiplier 3
device(config)#ip route 20.0.0.0/24 30.0.0.5 bfd
```

The **multi-hop BFD** session to the next hop (BFD neighbor) 30.0.0.5 uses the TX and RX intervals of 90ms.

When configuring**multi-hop static route** and **multi-hop bfd neighbor** , the protocol by which the nexthop is to be resolved must be stated using the IP route next-hop command.

Show commands

The **show ip static route** and **show ipv6 static route** command output indicates that BFD monitoring is enabled by the **b** next to the static route.

```
device# show ip static route
IP Static Routing Table - 3 entries:
STATIC Codes - b:BFD monitoring
   IP Prefix          Next Hop        Interface  Dis/Metric/Tag  Name
*  0.0.0.0/0          10.37.73.129    -          1/1/0
   0.0.0.0/0          10.37.73.1      -          1/1/0
b
 100.0.0.0/8        10.0.0.2        -          1/1/0
b
150.0.0.0/8        20.0.0.3        -          1/1/0
device#
device# show ipv6 static route
IPv6 Static Routing Table - 2 entries:
STATIC Codes - b:BFD monitoring
   IPv6 Prefix        Interface  Next Hop Router      Met/Dis/Tag Name
b
100::/64            eth 1/5    10::2               1/1/0
b
150::/64            eth 1/5    20::3               1/1/0
device#
```

The **show bfd** *applications* output indicates that BFD monitoring is enabled by the **static** and **static6** .

```
device# show bfd applications
  Registered Protocols Count: 4
     Protocol  VRFID     Parameter HoldoverInterval
     static6
   0         1         0
     static
    0         1         0
     bgp       1         0         0
     ospf      0         0         0
device#
```

The **show bfd neighbors details** output indicates that BFD monitoring is enabled by the **static** and **static6** .

```
device# show bfd neighbors details 20.0.0.3
NeighborAddress        State  Interface Holddown   Interval  R/H
20.0.0.3               UP     eth 1/5   300000     100000    Y/M
  Registered Protocols(Protocol/VRFID): static/0
  Local: Disc: 3, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 100000, MinRxInterval: 100000, Multiplier: 3
  Remote: Disc: 5, Diag: 0, Demand: 0 Poll: 0
          MinTxInterval: 100000, MinRxInterval: 100000, Multiplier: 3
  Stats: RX: 89596 TX: 87853 SessionUpCount: 1 at SysUpTime: 0:5:10:53.575
  Session Uptime: 0:1:0:16.300, LastSessionDownTimestamp: 0:0:0:0.0
  Tx Port: eth 1/1(eth 1/1),Rx Port: eth 1/1(eth 1/1)
```

```
  Using PBIF Assist: Y
device#
```

Configuring BGP4+

- Supported BGP4+ features....488
- BGP4+ overview....489
- Address family configuration level....489
- Configuring BGP4+....490
- Clearing BGP4+ information....500
- Displaying BGP4+ information....504
- Configuring BGP4+ graceful restart....546

Supported BGP4+ features

Lists BGP4+ features supported on NetIron devices.

The following features are documented in the *Multi-Service IronWare Routing Configuration Guide.*

Feature	Brocade NetIron XMR Series	Brocade NetIron MLX Series	Brocade NetIron CES 2000 Series BASE package	Brocade NetIron CES 2000 Series ME_PREM package	Brocade NetIron CES 2000 Series L3_PREM package	Brocade NetIron CER 2000 Series Base package	Brocade NetIron CER 2000 Series Advanced Services package
BGP4+	Yes	Yes	No	No	Yes	Yes	Yes
Configuring BGP4+ Neighbors Using Global or Unique Link Local IPv6 Addresses	Yes	Yes	No	No	Yes	Yes	Yes
Importing Routes into BGP4+	Yes	Yes	No	No	Yes	Yes	Yes
Advertising the Default BGP4+ Route	Yes	Yes	No	No	Yes	Yes	Yes
Using the IP default route as a valid next-hop for a BGP4+ route	Yes	Yes	No	No	Yes	Yes	Yes
Enabling next-hop recursion	Yes	Yes	No	No	Yes	Yes	Yes
BGP4+ Graceful Restart	Yes	Yes	Yes	Yes	Yes	Yes	Yes
BGP VPNv4 and VPNv6 Support	Yes	Yes	No	Yes	Yes	No	Yes

Feature	Brocade NetIron XMR Series	Brocade NetIron MLX Series	Brocade NetIron CES 2000 Series BASE package	Brocade NetIron CES 2000 Series ME_PREM package	Brocade NetIron CES 2000 Series L3_PREM package	Brocade NetIron CER 2000 Series Base package	Brocade NetIron CER 2000 Series Advanced Services package
BGP VRF4 and VRF6 Support	Yes	Yes	Yes	Yes	Yes	Yes	Yes

BGP4+ overview

The implementation of IPv6 supports multi protocol BGP (MBGP) extensions, which allow IPv6 BGP (known as BGP4+) to distribute routing information for protocols such as IPv4 BGP. The supported protocols are identified by address families. The extensions allow a set of BGP4+ peers to exchange routing information for multiple address families and sub-address families.

IPv6 MBGP functions similarly to IPv4 MBGP except for the following enhancements:

- An IPv6 unicast address family and network layer reachability information (NLRI).
- An IPv6 multicast address family
- Next hop attributes that use IPv6 addresses.

The implementation of BGP4+ supports the advertising of routes among different address families. BGP4+ multicast and unicast routes are supported. IPv6 unicast-routing is enabled by default.

Address family configuration level

The implementation of BGP4+ includes a new configuration level: address family. For IPv6, Brocade devices currently support the BGP4+ multicast and unicast address family configuration levels. The device enters the BGP4+ unicast address family configuration level when you enter the following command while at the global BGP configuration level:

```
device(config-bgp)# address-family ipv6 unicast
device(config-bgp-ipv6u)#
```

The **(config-bgp-ipv6u)#** prompt indicates that you are at the BGP4+ unicast address family configuration level.

While at the BGP4+ unicast address family configuration level, you can access several commands that allow you to configure BGP4+ unicast routes. The commands that you enter at this level apply only to IPv6 unicast address family only. You can generate a configuration for BGP4+ unicast routes that is separate and distinct from configurations for IPv4 unicast routes and IPv4 BGP multicast routes.

NOTE
The commands that you can access while at the IPv6 unicast address family configuration level are also available at the IPv4 unicast and multicast address family configuration levels. Where relevant, this

section discusses and provides IPv6-unicast-specific examples. You must first configure IPv6 unicast-routing in order for any IPv6 routing protocol to be active.

NOTE
Each address family configuration level allows you to access commands that apply to that particular address family only. To enable a feature in a particular address family, you must specify any associated commands for that feature in that particular address family. You cannot expect the feature, which you may have configured in the BGP4 unicast address family, to work in the BGP4+ unicast address family unless it is explicitly configured in the BGP4+ unicast address family.

To exit from the IPv6 unicast address family configuration level, enter the following command:

```
device(config-bgp-ipv6u)# exit-address-family
device(config-bgp)#
```

Entering this command returns you to the global BGP configuration level.

Configuring BGP4+

Before enabling BGP4+ on a device, you must enable the forwarding of IPv6 traffic on the device using the **ipv6 unicast-routing** command and enable IPv6 on at least one interface by configuring an IPv6 address or explicitly enabling IPv6 on that interface.

To configure BGP4+, you must do the following:

- Enable BGP4+.
- Configure BGP4+ neighbors using one of the following methods:
  - Add one neighbor at a time (neighbor uses global or unique local IPv6 address).
  - Add one neighbor at a time (neighbor uses a link-local IPv6 address).
  - Create a peer group and add neighbors individually.

The following configuration tasks are optional:

- Advertise the default route.
- Import specified routes into BGP4+.
- Redistribute prefixes into BGP4+.
- Aggregate routes advertised to BGP4 neighbors.
- Use route maps.

Enabling BGP4+

To enable BGP4+, enter commands such as the following:

```
device(config)# router bgp
BGP: Please configure 'local-as' parameter in order to run BGP4.
device(config-bgp)# local-as 1000
```

These commands enable BGP4+ and configure the autonomous system (1000) in which your device resides.

Syntax:[no] router bgp

To disable BGP, enter the **no** form of this command.

Syntax: local-as *number*

Specify the AS number in which the device you are configuring resides.

After enabling BGP4+, you can add neighbors to a BGP4+ device by entering commands such as the following:

```
device(config-bgp)# address-family ipv6 unicast
device(config-bgp-ipv6u)# neighbor 2001:db8:e0ff:783a::4 remote-as 1001
device(config-bgp-ipv6u)# neighbor 2001:db8:e0ff:783a::5 remote-as 1001
```

These commands add two neighbors with global IPv6 addresses 2001:db8:e0ff:783a::4 and 2001:db8:e0ff:783a::5 in AS 1001.

NOTE
The example above adds IPv6 neighbors at the BGP4+ unicast address family configuration level. These neighbors, by default, are enabled to exchange BGP4+ unicast prefixes. However, if you add IPv6 neighbors while at the global BGP configuration or IPv4 BGP unicast address family configuration level, the neighbors will not exchange BGP4+ unicast prefixes until you explicitly enable them to do so by entering the **neighbor activate** command at the BGP4+ unicast address family configuration level.

This section provides minimal information about adding BGP4+ neighbors, because its focus is to provide information about configuring BGP4+.

Configuring BGP4+ neighbors using global or unique link local IPv6 addresses

To configure BGP4+ neighbors using global or unique link local IPv6 addresses, you must add the IPv6 address of a neighbor in a remote autonomous system to the BGP4+ neighbor table of the local device. You must repeat this procedure for each neighbor that you want to add to a local device.

For example, to add the IPv6 address 2001:db8:93e8:cc00::1 of a neighbor in remote AS 4500 to the BGP4+ neighbor table of a device, enter commands such as the following:

```
device(config-bgp)# address-family ipv6 unicast
device(config-bgp-ipv6u)# neighbor 2001:db8:93e8:cc00::1 remote-as 4500
```

Syntax: neighbor *ipv6-address* **remote-as** *as-number*

NOTE
The example above adds an IPv6 neighbor at the BGP4+ unicast address family configuration level. This neighbor, by default, is enabled to exchange BGP4+ unicast prefixes. However, if you add an IPv6 neighbor while at the global BGP configuration or IPv4 BGP unicast address family configuration level, the neighbor will not exchange BGP4+ unicast prefixes until you explicitly enable it to do so by entering the **neighbor activate** command at the BGP4+ unicast address family configuration level.

The *ipv6-address* parameter specifies the IPv6 address of the neighbor. You must specify the *ipv6-address* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *as-number* parameter indicates the number of the AS in which the neighbor resides.

To delete the neighbor from the BGP4+ neighbor table, enter the **no** form of this command.

Adding BGP4+ neighbors using link-local addresses

To configure BGP4+ neighbors that use link-local addresses, you must do the following:

- Add the IPv6 address of a neighbor in a remote autonomous system to the BGP4+ neighbor table of the local device.
- Identify the neighbor interface over which the neighbor and local device will exchange prefixes.
- Configure a route map to set up a global next hop for packets destined for the neighbor.

Adding BGP4+ neighbor

To add the IPv6 link-local address fe80:4398:ab30:45de::1 of a neighbor in remote autonomous system 1000 to the BGP4+ neighbor table of a device, enter the following commands:

```
device(config-bgp)# address-family ipv6 unicast
device(config-bgp-ipv6u)# neighbor fe80:4398:ab30:45de::1 remote-as 1000
```

Syntax: neighbor *ipv6-address* **remote-as** *as-number*

NOTE
The example above adds an IPv6 neighbor at the BGP4+ unicast address family configuration level. This neighbor, by default, is enabled to exchange BGP4+ unicast prefixes. However, if you add an IPv6 neighbor while at the global BGP configuration or IPv4 BGP unicast address family configuration level, the neighbor will not exchange BGP4+ unicast prefixes until you explicitly enable it to do so by entering the **neighbor activate** command at the BGP4+ unicast address family configuration level.

The *ipv6-address* parameter specifies the IPv6 link-local address of the neighbor. A link-local address has a fixed prefix of FE80::/10. You must specify the *ipv6-address* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *as-number* parameter indicates the number of the AS in which the neighbor resides.

To delete the neighbor from the BGP4+ neighbor table, enter the **no** form of this command.

Identifying a neighbor interface

To specify Ethernet interface 3/1 as the neighbor interface over which the neighbor and local device will exchange prefixes, enter the following command:

```
device(config-bgp)# neighbor fe80:4398:ab30:45de::1 update-source ethernet 3/1
```

Syntax: neighbor *ipv6-address* **update-source** *ipv6-address* | **ethernet** *slot* | *port* | **loopback** *number* | **ve** *number*

The *ipv6-address* parameter specifies the IPv6 link-local address of the neighbor. A link-local address has a fixed prefix of fe80::/10. You must specify the *ipv6-address* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **ethernet**, **loopback**, and **ve** parameters specify the neighbor interface over which the neighbor and local device will exchange prefixes. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a loopback or VE interface, also specify the loopback or VE number.

Configuring a route map

To configure a route map that filters routes advertised to a neighbor or sets up a global next hop for packets destined for the neighbor with the IPv6 link-local address fe80:4398:ab30:45de::1, enter commands such as the following (start at the BGP4+ unicast address family configuration level):

```
device(config-bgp-ipv6u)# neighbor fe80:4398:ab30:45de::1 route-map out next-hop
device(config-bgp-ipv6u)# exit
device(config)# route-map next-hop permit 10
device(config-route-map)# match ipv6 address prefix-list next-hop-ipv6
device(config-route-map)# set ipv6 next-hop 2001:db8:3764::34
```

This route map applies to the BGP4+ unicast address family under which the **neighbor ipv6-address route-map** command is entered. This route map applies to the outgoing routes on the neighbor with the IPv6 link-local address fe80:4393:ab30:45de::1. If an outgoing route on the neighbor matches the route map, the route is distributed through the next hop router with the global IPv6 address 2001:db8:3764::34.

Syntax: neighbor *ipv6-address* **route-map** [**in** | **out**] *name*

The *ipv6-address* parameter specifies the IPv6 link-local address of the neighbor. A link-local address has a fixed prefix of fe80::/10. You must specify the *ipv6-address* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **in** keyword applies the route map to incoming routes. The **out** keyword applies the route map to outgoing routes.

The *name* parameter specifies a route map name.

Syntax: route-map *name* **deny** | **permit** *sequence-number*

The *name* parameter specifies a route map name.

The **deny** keyword denies the distribution of routes that match the route map. The **permit** keyword permits the distribution of routes that match the route map.

The *sequence-number* parameter specifies a sequence number for the route map statement.

Syntax: match ipv6 address prefix-list *name*

The **match ipv6 address prefix-list** command distributes any routes that have a destination IPv6 address permitted by a prefix list.

The *name* parameter specifies an IPv6 prefix list name.

Syntax: set ipv6 next-hop *ipv6-address*

The *ipv6-address* parameter specifies the IPv6 global address of the next-hop router. You must specify the *ipv6-address* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

Configuring a BGP4+ peer group

If a peer group has multiple neighbors with similar attributes, you can configure a peer group, then add neighbors to the group instead of configuring neighbors individually for all parameters.

NOTE
You can add IPv6 neighbors only to an IPv6 peer group. Brocade NetIron devices do not support adding an IPv4 neighbor to an IPv6 peer group and vice versa. IPv4 and IPv6 peer groups must remain separate.

To configure a BGP4+ peer group, you must perform the tasks listed below.

1. Create a peer group.
2. Add a neighbor to the local device.
3. Assign the IPv6 neighbor to the peer group.
4. Activate the IPv6 neighbor and peer group.

Creating a BGP4+ peer group

To create a peer group named "peer_group1," enter commands such as the following:

```
device(config-bgp)# address-family ipv6 unicast
device(config-bgp-ipv6u)# neighbor peer_group1 peer-group
```

Syntax: neighbor *peer-group-name* **peer-group**

Specify a name for the peer group.

To delete the peer group, enter the **no** form of this command.

Adding a neighbor to a local device

To add the IPv6 address 2001:db8:89::23 of a neighbor in remote AS 1001 to the BGP4+ neighbor table of a device, enter the following command:

```
device(config-bgp-ipv6u)# neighbor 2001:db8:89::23 remote-as 1001
```

NOTE
The example above adds an IPv6 neighbor at the BGP4+ unicast address family configuration level. This neighbor, by default, is enabled to exchange BGP4+ unicast prefixes. However, if you add an IPv6 neighbor while at the global BGP configuration or IPv4 BGP unicast address family configuration level, the neighbor will not exchange BGP4+ unicast prefixes until you explicitly enable it to do so by entering the **neighbor activate** command at the BGP4+ unicast address family configuration level.

Syntax: neighbor *ipv6-address* **remote-as** *as-number*

The *ipv6-address* parameter specifies the IPv6 address of the neighbor. You must specify the *ipv6-address* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *as-number* parameter indicates the number of the autonomous system in which the neighbor resides.

To delete the neighbor from the BGP4+ neighbor table, enter the **no** form of this command.

Assigning IPv6 neighbor to peer group

To assign an already configured neighbor (2001:db8:89::23) to the peer group peer_group1, enter the following command at the BGP4+ unicast address family configuration level:

```
device(config-bgp-ipv6u)# neighbor 2001:db8:89::23 peer-group peer_group1
```

Syntax: neighbor *ipv6-address* **peer-group** *peer-group-name*

The *ipv6-address* parameter specifies the IPv6 address of the neighbor. You must specify the *ipv6-address* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **peer-group** *peer-group-name* parameter indicates the name of the already created peer group.

To delete the mapping of the neighbor IPv6 address to the peer group, enter the **no** form of this command.

Activating the IPv6 neighbor /peer group

By default, a peer group is activated only in "address-family ipv4 unicast" mode. To activate the neighbor/peer group in "address-family ipv6-unicast" mode, use the **activate** command:

```
device(config-bgp-ipv6u)# neighbor 2001:db8:89::23 activate
device(config-bgp-ipv6u)# neighbor peer_group1 activate
```

Syntax: neighbor *ipv6-address* | *peer-group-name* **activate**

The *peer-group-name* parameter indicates the name of the already created peer group.

The following peer-group attributes/route policies are inherited by a group member when the peer-group is active in an ipv6 address-family:

- activate (address family)
- prefix-list
- route-map
- distribute-list
- filter-list
- unsuppress-map
- originate-default
- route-reflect-client
- weight
- max-prefix
- send-community
- send-extended-community

To deactivate the neighbor/peer group, enter the **no** form of this command.

Advertising the default BGP4+ route

By default, the BGP4+ device does not originate and advertise a default BGP4+ route. A default route is the IPv6 address :: and the route prefix 0; that is, ::/0.

You can enable the BGP4+ device to advertise the default BGP4+ route by specifying the **default-information-originate** command at the BGP4+ unicast address family configuration level. Before entering this command, the default route ::/0 must be present in the IPv6 route table.

To enable the BGP4+ device to advertise the default route, enter the following command:

```
device(config-bgp-ipv6u)# default-information-originate
```

Syntax: [no] default-information-originate

You can also enable the BGP4+ device to send the default route to a particular neighbor by specifying the **neighbor default-originate** command at the BGP4+ unicast address family configuration level. This command does not require the presence of the default route ::/0 in the IPv6 route table.

For example, to enable the BGP4+ device to send the default route to a neighbor with the IPv6 address of 2001:db8:89::23, enter a command such as the following:

```
device(config-bgp-ipv6u)# neighbor 2001:db8:89::23 default-originate
```

Syntax: [no] neighbor *ipv6-address* **default-originate** [**route-map** *name*]

The *ipv6-address* parameter specifies a neighbor by its IPv6 address. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

Specifying the optional **route-map** *name* parameter injects the default route conditionally, based on the match conditions in the route map.

Importing routes into BGP4+

By default, the device does not import routes into BGP4+. This section explains how to use the **network** command to enable the importing of specified routes into BGP4+.

NOTE
The routes imported into BGP4+ must first exist in the IPv6 unicast route table.

For example, to import the IPv6 prefix 2001:db8::/32 into the BGP4+ database, enter the following command at the BGP4+ unicast address family configuration level:

```
device(config-bgp-ipv6u)# network 2001:db8::/32
```

Syntax: network *ipv6-prefixlprefix-length* [**route-map** *name*]

You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

You can specify the optional **route-map** *name* parameter if you want to change attributes of a route when importing it into BGP4+.

To disable the importing of a specified route, enter the **no** form of this command without the route-map parameter.

Redistributing prefixes into BGP4+

You can configure the device to redistribute routes from the following sources into BGP4+:

- Static IPv6 routes
- Directly connected IPv6 networks
- OSPFv3
- RIPng
- IS-IS

You can redistribute routes in the following ways:

- By route types, for example, the device redistributes all IPv6 static and RIPng routes.
- By using a route map to filter which routes to redistribute, for example, the device redistributes specified IPv6 static and RIPng routes only.

For example, to configure the redistribution of all RIPng routes into the BGP4+ unicast database, enter the following commands at the BGP4+ address family configuration level:

```
device(config-bgp-ipv6u)# redistribute rip
```

Syntax: redistribute *protocol* [**level-1** | **level-1-2** | **level-2**] [**match external1** | **external2** | **internal**] [**metric** *metric-value*] [**route-map** *name*]

The *protocol* parameter can be **connected**, **ospf**, **rip**, **static**, or ISIS.

If you specify **ospf** as the protocol, you can optionally specify the redistribution of external 1, external 2, or internal routes. (The default is internal.)

The **metric** *metric-value* parameter specifies the metric used for the redistributed route. If a value is not specified for this option, and no value is specified using the **default-metric** command at the BGP4+ unicast address family configuration level, the metric value for the IPv6 static, RIPng, or IPv6 OSPF route is used. Use a value consistent with the destination protocol.

The *name* parameter specifies a route map name.

Aggregating routes advertised to BGP4 neighbors

By default, a device advertises individual BGP4+ routes for all the networks. The aggregation feature allows you to configure a device to aggregate routes in a range of networks into a single IPv6 prefix. For example, without aggregation, a will individually advertise routes for networks 2001:db8:0001:0000::/64, 2001:db8:0002:0000::/64, 2001:db8:0003:0000::/64, and so on. You can configure the device to instead send a single, aggregate route for the networks. The aggregate route would be advertised as 2001:db8::/24 to BGP4 neighbors.

To aggregate BGP4+ routes for 2001:db8:0001:0000::/64, 2001:db8:0002:0000::/64, 2001:db8:0003:0000::/64, enter the following command.

```
device(config-bgp-ipv6u)# aggregate-address 2001:db8::/24 summary-only
```

Syntax: aggregate-address *ipv6-prefix*|*prefix-length* [**as-set**] [**summary-only**] [**suppress-map** *map-name*] [**advertise-map** *map-name*] [**attribute-map** *map-name*]

The *ipv6-prefix and prefix-length* parameters specify the aggregate value for the networks. You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The **as-set** keyword causes the device to aggregate AS-path information for all the routes in the aggregate address into a single AS-path.

The **summary-only** keyword prevents the device from advertising more specific routes contained within the aggregate route.

The **suppress-map** *map-name* parameter prevents the more specific routes contained in the specified route map from being advertised.

The **advertise-map** *map-name* parameter configures the device to advertise the more specific routes in the specified route map.

The **attribute-map** *map-name* parameter configures the device to set attributes for the aggregate routes based on the specified route map.

NOTE
For the **suppress-map**, **advertise-map**, and **attribute-map** parameters, the route map must already be defined.

To remove an aggregate route from a BGP4 neighbor advertisement, use the **no** form of this command without any parameters.

Using route maps

You can use a route map to filter and change values in BGP4+ routes. Currently, you can apply a route map to IPv6 unicast routes that are independent of IPv4 routes.

To configure a route map to match on IPv6 unicast routes, enter commands such as the following.

```
device(config)# router bgp
device(config-bgp)# address-family ipv6 unicast
device(config-bgp-ipv6u)# neighbor 2001:db8:df78::67 remote-as 1001
device(config-bgp-ipv6u)# neighbor 2001:db8:df78::67 route-map in map1
device(config-bgp-ipv6u)# exit
device(config)# ipv6 prefix-list ipv6_uni seq 10 permit 2001:db8::/32
device(config)# route-map map1 permit 10
device(config-routemap-map1)# match ipv6 address prefix-list ipv6_uni
```

This example configures a route map named "map1" that permits incoming IPv6 unicast routes that match the prefix list named "ipv6_uni" (2001:db8::/32). Note that you apply the route map while at the BGP4+ unicast address family configuration level.

Enabling next-hop recursion

For each BGP4+ route learned, the device performs a route lookup to obtain the IPv6 address of the next-hop for the route. A BGP4+ route is eligible for addition in the IPv6 route table only if the following conditions are true:

- The lookup succeeds in obtaining a valid next-hop IPv6 address for the route.
- The path to the next-hop IPv6 address is an IGP path or a static route path.

By default, the software performs only one lookup for the next-hop IPv6 address for the BGP4+ route. If the next-hop lookup does not result in a valid next-hop IPv6 address, or the path to the next-hop IPv6 address is a BGP4+ path, the software considers the BGP4+ route destination to be unreachable. The route is not eligible to be added to the IPv6 route table.

The BGP4+ route table can contain a route with a next-hop IPv6 address that is not reachable through an IGP route, even though the device can reach a hop farther away through an IGP route. This can occur when the IGPs do not learn a complete set of IGP routes, so the device learns about an internal route through IBGP instead of through an IGP. In this case, the IPv6 route table will not contain a route that can be used to reach the BGP4+ route destination.

To enable the device to find the IGP route to the next-hop gateway for a BGP4+ route, enable recursive next-hop lookups. With this feature enabled, if the first lookup for a BGP4+ route results in an IBGP path that originated within the same AS, rather than an IGP path or static route path, the device performs a lookup on the next-hop IPv6 address for the next-hop gateway. If this second lookup results in an IGP path, the software considers the BGP4+ route to be valid and adds it to the IPv6 route table. Otherwise, the device performs another lookup on the next-hop IPv6 address of the next-hop for the next-hop gateway, and so on, until one of the lookups results in an IGP route.

You must configure a static route or use an IGP to learn the route to the EBGP multihop peer.

Enabling recursive next-hop lookups

The recursive next-hop lookups feature is disabled by default.

To enable recursive next-hop lookups, enter the following command at the BGP4+ address family configuration level of the CLI.

```
Brocade(config-bgp-ipv6u)# next-hop-recursion
```

Syntax: [no] next-hop-recursion

Example when recursive route lookups are disabled

The output here shows the results of an unsuccessful next-hop lookup for a BGP4+ route. In this case, next-hop recursive lookups are disabled. This example is for the BGP4+ route to network 10.240.0.0/24.

In this example, the device cannot reach 10.240.0.0/24, because the next-hop IP address for the route is an IBGP route instead of an IGP route, and is considered unreachable by the device. The IP route table entry for the next-hop gateway for the BGP4+ route's next-hop gateway (10.102.0.1/24) is shown here.

Since the route to the next-hop gateway is a BGP4+ route, and not an IGP route, it cannot be used to reach 10.240.0.0/24. In this case, the device tries to use the default route, if present, to reach the subnet that contains the BGP4+ route next-hop gateway.

The output here shows the results of an unsuccessful next-hop lookup for a BGP4+ route. In this case, next-hop recursive lookups are disabled. This example is for the BGP4+ route to network 2001:db8::/64.

```
device# show ipv6 bgp route
Total number of BGP Routes: 4
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH m:NOT-INSTALLED-MULTIPATH
       S:SUPPRESSED F:FILTERED s:STALE
       Prefix             Next Hop        MED        LocPrf     Weight Status
1      2001:ab::1/128     2116:21:16::2   1          100        0      BI
         AS_PATH:
2      2001:db8::/64      2001:ab::1                 100        0      I
         AS_PATH: 65000 65001
3      2007:7002:17::/64  2071:34::1      0          100        0      BE
         AS_PATH: 60750
4      2007:7002:17::/64  2071:33::1      0          100        0      I
         AS_PATH: 60750
```

In this example, the device cannot reach 2001:db8::/64, because the next-hop IPv6 address for the route is an IBGP route instead of an IGP route, and is considered unreachable by the device. The IPv6 route table entry for the BGP4+ route's next-hop gateway (2001:ab::1) is shown here.

```
device# show ipv6 route 2001:ab::1
Type Codes - B:BGP C:Connected I:ISIS L:Local O:OSPF R:RIP S:Static
BGP  Codes - i:iBGP e:eBGP
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2
Type IPv6 Prefix           Next Hop Router    Interface     Dis/Metric     Uptime
Bi   2001:ab::1/128        fe80::768e:f8ff:fef9:7d80
                                              ve 40         200/1          29m27s
```

Since the route to the next-hop gateway is a BGP4+ route, and not an IGP route, it cannot be used to reach 2001:db8::/64. In this case, the device tries to use the default route, if present, to reach the subnet that contains the BGP4+ route next-hop gateway.

Example when recursive route lookups are enabled

When recursive next-hop lookups are enabled, the device continues to look up the next-hop gateways along the route until the device finds an IGP route to the BGP4+ route destination.

The first lookup results in an IBGP route, to network 10.102.0.0/24

Since the route to 10.102.0.1/24 is not an IGP route, the device cannot reach the next hop through IP, and so cannot use the BGP4+ route. In this case, since recursive next-hop lookups are enabled, the device next performs a lookup for the 10.102.0.0.1's next-hop gateway, 10.0.0.1.

The next-hop IP address for 10.102.0.1 is not an IGP route, which means the BGP4+ route destination still cannot be reached through IP. The recursive next-hop lookup feature performs a lookup on the next-hop gateway for 10.0.0.1

This lookup results in an IGP route that is a directly-connected route. As a result, the BGP4+ route destination is now reachable through IGP, which means the BGP4+ route can be added to the IP route table. The IP route table with the BGP4+ route is shown here.

The device can use this route because it has an IPv6 route to the next-hop gateway. Without recursive next-hop lookups, this route would not be in the IPv6 route table.

Clearing BGP4+ information

This section contains information about clearing the following for BGP4+:

- Route flap dampening.
- Route flap dampening statistics.
- Neighbor information.
- BGP4+ routes in the IPv6 route table.
- Neighbor traffic counters.

NOTE
The **clear** commands implemented for BGP4+ correspond to the **clear** commands implemented for IPv4 BGP. For example, you can specify the **clear ipv6 bgp flap-statistics** command for IPv6 and the **clear ip bgp flap-statistics** for IPv4.

Removing route flap dampening

You can un-suppress routes by removing route flap dampening from the routes. The device allows you to un-suppress all routes at once or un-suppress individual routes.

To un-suppress all the suppressed routes, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
device# clear ipv6 bgp dampening
```

Syntax: clear ipv6 bgp dampening [*ipv6-prefix*/*prefix-length*]

You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

To un-suppress a specific route, enter a command such as the following:

```
device# clear ipv6 bgp dampening 2001:db8::/32
```

This command un-suppresses only the routes for network 2001:db8::/32.

Clearing route flap dampening statistics

The device allows you to clear all route flap dampening statistics or statistics for a specified IPv6 prefix or a regular expression.

NOTE
Clearing the dampening statistics for a route does not change the dampening status of the route.

To clear all the route dampening statistics, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
device# clear ipv6 bgp flap-statistics
```

Syntax: clear ipv6 bgp flap-statistics [*ipv6-prefix*|*prefix-length* | **neighbor** *ipv6-address* | **regular-expression** *regular-expression*]

The *ipv6-prefix* and *prefix-length* parameters clear route flap dampening statistics for a specified IPv6 prefix. You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The **neighbor** *ipv6-address* parameter clears route flap dampening statistics only for routes learned from the neighbor with the specified IPv6 address.

The **regular-expression** *regular-expression* parameter is a regular expression.

Clearing BGP4+ local route information

You can clear locally imported or routes redistributed into BGP4+.

To clear all local route information, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
device# clear ipv6 bgp local routes
```

Syntax: clear ipv6 bgp local routes

Clearing BGP4+ neighbor information

You can perform the following tasks related to BGP4+ neighbor information:

- Clear diagnostic buffers.
- Reset a session to send and receive Outbound Route Filters (ORFs).
- Close a session, or reset a session and resend or receive an update.
- Clear traffic counters.
- Clear route flap dampening statistics.

Clearing BGP4+ neighbor diagnostic buffers

You can clear the following BGP4+ neighbor diagnostic information in buffers:

- The first 400 bytes of the last packet that contained an error.
- The last NOTIFICATION message either sent or received by the neighbor.

To display these buffers, use the **last-packet-with-error** keyword with the **show ipv6 bgp neighbors** command.

You can clear the buffers for all neighbors, for an individual neighbor, or for all the neighbors within a specific peer group or AS.

To clear these buffers for neighbor 2000:db8::1, enter the following commands at the Privileged EXEC level or any of the Config levels of the CLI.

```
device# clear ipv6 bgp neighbor 2001:db8:37::1 last-packet-with-error
device# clear ipv6 bgp neighbor 2001:db8:37::1 notification-errors
```

Syntax: clear ipv6 bgp neighbor all | *ipv6-address* | *peer-group-name* | *as-number* **last-packet-with-error** | **notification-errors**

The **all**, *ipv6-address*, *peer-group-name*, and *as-num* parameters specify the neighbor. The *ipv6-address* parameter specifies a neighbor by its IPv6 address. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373. The *peer-group-name* specifies all neighbors in a specific peer group. The *as-num* parameter specifies all neighbors within the specified AS. The **all** keyword specifies all neighbors.

The **last-packet-with-error** keyword clears the buffer containing the first 400 bytes of the last packet that contained errors.

The **notification-errors** keyword clears the notification error code for the last NOTIFICATION message sent or received.

Resetting a BGP4+ neighbor session to send and receive ORFs

You can perform a hard or soft reset of a BGP4+ neighbor session to send or receive ORFs.

To perform a hard reset of a neighbor session and send ORFs to the neighbor, enter a command such as the following.

```
device# clear ipv6 bgp neighbor 2001:db8:38::1
```

This command resets the BGP4+ session with neighbor 2001:db8:38::1 and sends the ORFs to the neighbor when the neighbor comes up again. If the neighbor sends ORFs to the device, the accepts them if the send capability is enabled.

To perform a soft reset of a neighbor session and send ORFs to the neighbor, enter a command such as the following.

```
device(config)# clear ipv6 bgp neighbor peer_group1 soft in prefix-list
```

Syntax: clear ipv6 bgp neighbor *ipv6-address* | *peer-group-name* [**soft in prefix-filter**]

The *ipv6-address* parameter specifies a neighbor by its IPv6 address. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *peer-group-name* specifies all neighbors in a specific peer group.

If you use the **soft in prefix-filter** keyword, the device sends an updated IPv6 prefix list to the neighbor as part of its route refresh message to the neighbor.

Closing or resetting a BGP4+ neighbor session

You can close a neighbor session or resend route updates to a neighbor. You can specify all neighbors, a single neighbor, or all neighbors within a specific peer group or autonomous system.

If you close a neighbor session, the device and the neighbor clear all the routes they learned from each other. When the and neighbor establish a new BGP4+ session, they exchange route tables again. Use this method if you want the device to relearn routes from the neighbor and resend its own route table to the neighbor.

If you use the **soft-outbound** keyword, the device compiles a list of all the routes it would normally send to the neighbor at the beginning of a session. However, before sending the updates, the also applies the filters and route maps you have configured to the list of routes. If the filters or route maps result in changes to the list of routes, the sends updates to advertise, change, or even withdraw routes on the neighbor as needed. This ensures that the neighbor receives only the routes you want it to contain. Even if the neighbor already contains a route learned from the that you later decided to filter out, using the soft-outbound option removes that route from the neighbor. If no change is detected from the previously sent routes, an update is not sent.

For example, to close all neighbor sessions and thus flush all the routes exchanged by the device and all neighbors, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
device# clear ipv6 bgp neighbor all
```

Syntax: clear ipv6 bgp neighbor all | *ipv6-address* | *peer-group-name* | *as-number* [**soft-outbound** | **soft** [**in** | **out**]]

The **all**, i*pv6-address*, *peer-group-name*, and *as-number* specify the neighbor. The *ipv6-address* parameter specifies a neighbor by its IPv6 address. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373. The *peer-group-name* specifies all neighbors in a specific peer group. The *as-number* parameter specifies all neighbors within the specified AS. The **all** keyword specifies all neighbors.

Use the **soft-outbound** keyword to perform a soft reset of a neighbor session and resend only route update changes to a neighbor.

Use the **soft in** parameter to perform a soft reset of a neighbor session and requests a route update from a neighbor.

Use the **soft out** parameter to perform a soft reset of a neighbor session and resend all routes to a neighbor.

Clearing BGP4+ neighbor traffic counters

You can clear the BGP4+ message counter (reset them to 0) for all neighbors, a single neighbor, or all neighbors within a specific peer group or autonomous system.

For example, to clear the BGP4+ message counter for all neighbors within an autonomous system 1001, enter a command such as the following at the Privileged EXEC level or any of the Config levels of the CLI.

```
device# clear ipv6 bgp neighbor 1001 traffic
```

Syntax: clear ipv6 bgp neighbor all | *ipv6-address* | *peer-group-name* | *as-number* **traffic**

The **all**, *ipv6-address*, *peer-group-name*, and *as-number* specify the neighbor. The *ipv6-address* parameter specifies a neighbor by its IPv6 address. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373. The *peer-group-name* specifies all neighbors in a specific peer group. The *as-number* parameter specifies all neighbors within the specified autonomous system. The **all** keyword specifies all neighbors.

Specify the **traffic** keyword to clear the BGP4+ message counter.

Clearing BGP4+ neighbor route flap dampening statistics

The device allows you to clear all route flap dampening statistics for a specified BGP4+ neighbor.

NOTE
Clearing the dampening statistics for a neighbor does not change the dampening status of a route.

To clear all of the route flap dampening statistics for a neighbor, enter a command such as the following at the Privileged EXEC level or any of the Config levels of the CLI.

```
device# clear ipv6 bgp neighbor 2001:db8:47::1 flap-statistics
```

Syntax: clear ipv6 bgp neighbor *ipv6-address* **flap-statistics**

The *ipv6-address* parameter specifies a neighbor by its IPv6 address. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

Specify the **flap-statistics** keyword to clear route flap dampening statistics for the specified neighbor.

Clearing and resetting BGP4+ routes in the IPv6 route table

You can clear all BGP4+ routes or only those routes associated with a particular IPv6 prefix from the IPv6 route table and reset the routes. When cleared, the BGP4+ routes are removed from the IPv6 main route table and then restored again.

For example, to clear all BGP4+ routes and reset them, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
device# clear ipv6 bgp routes
```

Syntax: clear ip bgp routes [*ipv6-prefix*|*prefix-length*]

The *ipv6-prefix* and *prefix-length* parameters clear routes associated with a particular IPv6 prefix. You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

Clearing traffic counters for all BGP4+ neighbors

To clear the message counters (reset them to 0) for all BGP4+ neighbors, enter the following command.

```
device# clear ipv6 bgp traffic
```

Syntax: clear ipv6 bgp traffic

Displaying BGP4+ information

You can display the following BGP4+ information:

- BGP4+ route table.
- BGP4+ route information.
- BGP4+ route-attribute entries.
- BGP4+ configuration information.
- Dampened BGP4+ paths.
- Filtered-out BGP4+ routes.
- BGP4+ route flap dampening statistics.
- BGP4+ neighbor information.
- BGP4+ peer group configuration information.
- BGP4+ summary information.

NOTE
The **show** commands implemented for BGP4+ correspond to the **show** commands implemented for IPv4 BGP. For example, you can specify the **show ipv6 bgp** command for IPv6 and the **show ip bgp** command for IPv4. Also, the displays for the IPv4 and IPv6 versions of the **show** commands are

similar except where relevant, IPv6 neighbor addresses replace IPv4 neighbor addresses, IPv6 prefixes replace IPv4 prefixes, and IPv6 next-hop addresses replace IPv4 next-hop addresses.

Displaying the BGP4+ route table

BGP4+ uses filters you define, as well as an algorithm to determine the preferred route to a destination. BGP4+ sends only the preferred route to the device's IPv6 table. However, if you want to view all the routes BGP4+ knows about, you can display the BGP4+ table.

To display the BGP4+ route table, enter the following command at any level of the CLI.

```
device# show ipv6 bgp routes
Total number of BGP Routes: 4
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH m:NOT-INSTALLED-MULTIPATH
       S:SUPPRESSED F:FILTERED s:STALE
       Prefix             Next Hop    MED    LocPrf   Weight Status
1          2001:db8:1::/64   2001:db8:1111::2     1      100      32768       BL

        AS_PATH:
2          2001:db8:2::/64   2001:db8::30.30.30.1    1      100      0           BI

        AS_PATH:
3          2001:db8:1111::/64 ::                   0      100      32768       BL

        AS_PATH:
4          2001:db8:2222::/64 2001:db8::30.30.30.1    0      100     0           BI

        AS_PATH:
```

TABLE 94 show ipv6 bgp routes output descriptions

Field	Description
Number of BGP4+ Routes	The number of routes displayed by the command.
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the Status column of the display. The status codes are described in the command's output.
Prefix	The route's prefix.
Next Hop	For normal IPv6 routes, next hop is the next hop IPv6 router to reach the destination. For the 6PE routes, next hop is the IPv4-mapped IPv6 address of the peer 6PE router.
Metric	The value of the route's MED attribute. If the route does not have a metric, this field is blank.
LocPrf	The degree of preference for the advertised route relative to other routes in the local AS. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 - 4294967295.
Weight	The value that this device associates with routes from a specific neighbor. For example, if the receives routes to the same destination from two BGP4+ neighbors, the prefers the route from the neighbor with the larger weight.

TABLE 94 show ipv6 bgp routes output descriptions (Continued)

Field	Description
Status	The route's status, which can be one or more of the following: • A - AGGREGATE. The route is an aggregate route for multiple networks. • B - BEST. BGP4+ has determined that this is the optimal route to the destination. • b - NOT-INSTALLED-BEST - BGP4+ has determined that this is the optimal route to the destination but did not install it in the IPv6 route table because the device received better routes from other sources (such as OSPFv3, RIPng, or static IPv6 routes). • C - CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D - DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • E - EBGP. The route was learned through a in another AS. • H - HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I - IBGP. The route was learned through a in the same AS. • L - LOCAL. The route originated on this. • M - MULTIPATH. BGP4+ load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE** If the "m" is shown in lowercase, the software was not able to install the route in the IPv6 route table. • S - SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors.
AS-PATH	The AS-path information for the route.

Syntax: show ipv6 bgp routes [*ipv6-prefixIprefix-length* | *table-entry-number* | **age** *seconds* | **as-path-access-list** *name* | **as-path-filter** *number* | **best** | **cidr-only** | [**community** *number* | **no-export** | **no-advertise** | **internet** | **local-as**] | **community-access-list** *name* | **community-filter** *number* | **detail** [*option*] | **local** | **neighbor** *ipv6-address* | **nexthop** *ipv6-address* | **no-best** | **prefix-list** *name* | **regular-expression** *regular-expression* | **route-map** *name* | **summary** | **unreachable**]

You can use the following options with the **show ipv6 bgp routes** command to determine the content of the display:

The *ipv6-prefix and prefix-length* parameters display routes for a specific network. You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The *table-entry-number* parameter specifies the table entry with which you want the display to start. For example, if you specify 100, the display shows entry 100 and all entries subsequent to entry 100.

The **age***seconds* parameter displays only the routes that have been received or updated more recently than the number of seconds you specify.

The **as-path-access-list** *name* parameter filters the display using the specified AS-path ACL.

The **as-path-filter** *number* parameter filters the display using the specified AS-path filter.

The **best** keyword displays the routes received from neighbors that the device selected as the best routes to their destinations.

The **cidr-only** keyword lists only the routes whose network masks do not match their class network length.

The **community** *number* parameter lets you display routes for a specific community. You can specify **local-as** , **no-export**, **no-advertise**, **internet**, or a private community number. You can specify the community number as either two five-digit integer values of up to 1- 65535, separated by a colon (for example, 12345:6789) or a single long integer value.

The **community-access-list** *name* parameter filters the display using the specified community ACL.

The **community-filter** *number* parameter lets you display routes that match a specific community filter.

The **detail** *option* parameter lets you display more details about the routes. You can refine your request by also specifying one of the other parameters after the **detail** keyword.

The **local** keyword displays routes that are local to the device.

The **neighbor** *ipv6-address* parameter displays routes learned from a specified BGP4+ neighbor.

The **nexthop** *ipv6-address* parameter displays the routes for a specified next-hop IPv6 address. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **no-best** keyword displays the routes for which none of the routes to a given prefix were selected as the best route. The IPv6 route table does not contain a BGP4+ route for any of the routes listed using this option.

The **prefix-list** *name* parameter filters the display using the specified IPv6 prefix list.

The **regular-expression** *regular-expression* parameter filters the display based on a regular expression.

The **route-map** *name* parameter filters the display using the specified route map. The software displays only the routes that match the match statements in the route map. The software disregards the route map's set statements.

The **summary** keyword displays summary information for the routes.

The **unreachable** keyword displays the routes that are unreachable because the device does not have a valid RIPng, OSPFv3, or static IPv6 route to the next hop.

To display details about BGP4+ routes, enter the following command at any level of the CLI.

```
device# show ipv6 bgp route detail
Total number of BGP Routes: 4
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH m:NOT-INSTALLED-MULTIPATH
       S:SUPPRESSED F:FILTERED s:STALE
1 Prefix: 2001:db8:1::/64,  Status: BL,  Age: 0h1m14s
    NEXT_HOP: 2001:db8:1111::2, Learned from Peer: Local Router
    In-Label: 794624
    LOCAL_PREF: 100,  MED: 1,  ORIGIN: incomplete,  Weight: 32768
    AS_PATH:
    Adj_RIB_out count: 1,  Admin distance 1
2 Prefix: 2001:db8:2::/64,  Status: BI,  Age: 0h0m8s
    NEXT_HOP: 2001:db8::ffff:30:1, Metric: 1,  Learned from Peer: 10.30.30.1 (1)
    Out-Label: 794624
    LOCAL_PREF: 100,  MED: 1,  ORIGIN: incomplete,  Weight: 0
    AS_PATH:
3 Prefix: 2001:db8:1111::/64,  Status: BL,  Age: 0h2m26s
    NEXT_HOP: ::, Learned from Peer: Local Router
    In-Label: 794624
    LOCAL_PREF: 100,  MED: 0,  ORIGIN: incomplete,  Weight: 32768
    AS_PATH:
    Adj_RIB_out count: 1,  Admin distance 1
4 Prefix: 2001:db8:2222::/64,  Status: BI,  Age: 0h0m35s
    NEXT_HOP: 2001:db8::ffff:30:1, Metric: 1,  Learned from Peer: 10.30.30.1 (1)
```

```
Out-Label: 794624
LOCAL_PREF: 100,  MED: 0,  ORIGIN: incomplete,  Weight: 0
AS_PATH:
```

TABLE 95 show ipv6 bgp route detail output descriptions

Field	Description
Number of BGP4+ Routes advertised to specified neighbor (appears only in display for all routes)	For information about this field, refer to Displaying BGP4+ route information on page 510.
Status codes	For information about this field, refer to Displaying BGP4+ route information on page 510.
Prefix	For information about this field, refer to Displaying BGP4+ route information on page 510.
Status	For information about this field, refer to Displaying BGP4+ route information on page 510.
Age	The age of the advertised route, in seconds.
Next Hop	For information about this field, refer to Displaying BGP4+ route information on page 510.
Learned from Peer	The IPv6 address of the neighbor from which this route is learned. "Local Router" indicates that the device itself learned the route.
In-Label	The MPLS inner label in the 6PE packet received from the MPLS network.
Out-Label	The MPLS inner label in the 6PE packet sent to the MPLS network.
LOCAL_PREF	For information about this field, refer to Displaying BGP4+ route information on page 510.
MED	The value of the advertised route's MED attribute. If the route does not have a metric, this field is blank.

TABLE 95 show ipv6 bgp route detail output descriptions (Continued)

Field	Description
Origin	The source of the route information. The origin can be one of the following: • A - AGGREGATE. The route is an aggregate route for multiple networks. • B - BEST. BGP4+ has determined that this is the optimal route to the destination. • b - NOT-INSTALLED-BEST - BGP4+ has determined that this is the optimal route to the destination but did not install it in the IPv6 route table because the device received better routes from other sources (such as OSPFv3, RIPng, or static IPv6 routes). • C - CONFED_EBGP. The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D - DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • EGP - The routes with this set of attributes came to BGP4+ through EGP. • H - HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • IGP - The routes with this set of attributes came to BGP4+ through IGP. • L - LOCAL. The route originated on this device. • M - MULTIPATH. BGP4+ load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE** If the "m" is shown in lowercase, the software was not able to install the route in the IPv6 route table. • S - SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors.
Weight	For information about this field, refer to Displaying BGP4+ route information on page 510.
AS-PATH	For information about this field, refer to Displaying BGP4+ route information on page 510.
Adj_RIB_out count	The number of neighbors to which the route has been or will be advertised. This is the number of times the route has been selected as the best route and placed in the Adj-RIB-Out (outbound queue) for a BGP4+ neighbor.
Admin Distance	The administrative distance of the route.

Syntax: show ipv6 bgp routes detail [*ipv6-prefix*|*prefix-length* | *table-entry-number* | **age** *seconds* | **as-path-access-list** *name* | **as-path-filter** *number* | **best** | **cidr-only** | [**community** *number* | **no-export** | **no-advertise** | **internet** | **local-as**] | **community-access-list** *name* | **community-filter** *number* | **local** | **neighbor** *ipv6-address* | **nexthop** *ipv6-address* | **no-best** | **prefix-list** *name* | **regular-expression** *regular-expression* | **route-map** *name* | **summary** | **unreachable**]

You can use the following options with the **show ipv6 bgp routes detail** command to determine the content of the display.

The *ipv6-prefix and prefix-length* options display details about routes for a specific network. You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in

RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The *table-entry-number* parameter specifies the table entry with which you want the display to start. For example, if you specify 100, the display shows entry 100 and all entries subsequent to entry 100.

The **age***seconds* parameter displays only the routes that have been received or updated more recently than the number of seconds you specify.

The **as-path-access-list** *name* parameter filters the display using the specified AS-path ACL.

The **as-path-filter** *number* parameter filters the display using the specified AS-path filter.

The **best** keyword displays the routes received from neighbors that the device selected as the best routes to their destinations.

The **cidr-only** keyword lists only the routes whose network masks do not match their class network length.

The **community** *number* parameter lets you display routes for a specific community. You can specify **local-as**, **no-export**, **no-advertise**, **internet**, or a private community number. You can specify the community number as either two five-digit integer values of up to 1- 65535, separated by a colon (for example, 12345:6789) or a single long integer value.

The **community-access-list** *name* parameter filters the display using the specified community ACL.

The **community-filter** *number* parameter lets you display routes that match a specific community filter.

The **detail** keyword lets you display more details about the routes. You can refine your request by also specifying one of the other parameters after the **detail** keyword.

The **local** keyword displays routes that are local to the device.

The **neighbor** *ipv6-address* parameter displays routes learned from a specified BGP4+ neighbor.

The **nexthop** *ipv6-address* option displays the routes for a specified next-hop IPv6 address. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **no-best** keyword displays the routes for which none of the routes to a given prefix were selected as the best route. The IPv6 route table does not contain a BGP4+ route for any of the routes listed using this option.

The **prefix-list** *name* parameter filters the display using the specified IPv6 prefix list.

The **regular-expression** *regular-expression* parameter filters the display based on a regular expression.

The **route-map** *name* parameter filters the display using the specified route map. The software displays only the routes that match the match statements in the route map. The software disregards the route map's set statements.

The **summary** keyword displays summary information for the routes.

The **unreachable** keyword displays the routes that are unreachable because the device does not have a valid RIPng, OSPFv3 or static IPv6 route to the next hop.

Displaying BGP4+ route information

You can display all BGP4+ routes known by a device, only those routes that match a specified prefix, or routes that match a specified or longer prefix.

To display all BGP4+ routes known by the device, enter the following command at any level of the CLI.

```
device# show ipv6 bgp
```

```
Total number of BGP Routes: 2
Status codes: s suppressed, d damped, h history, * valid, > best, i internal
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network            Next Hop        Metric LocPrf Weight Path
*>  2001:db8::/32      ::              1      100    32768  ?
*>  2001:db8:1234::/48 ::              1      100    32768  ?
```

Syntax: show ipv6 bgp *ipv6-prefix*|*prefix-length* [**longer-prefixes**]

The *ipv6-prefix* and *prefix-length* parameters allow you to display routes that match a specified BGP prefix only. You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The **longer-prefixes** keyword allows you to display routes that match a specified or longer BGP prefix. For example, if you specify **2001:db8::/32 longer-prefixes**, then all routes with the prefix 2001:db8::/32 or that have a longer prefix (such as 2001:db8:e016::/48) are displayed.

To display only those routes that match prefix 2001:db8::/32, enter the following command at any level of the CLI.

```
device# show ipv6 bgp 2001:db8::/32
Number of BGP Routes matching display condition : 1
Status codes: s suppressed, d damped, h history, * valid, > best, i internal
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network            Next Hop        MED LocPrf Weight Path
*>  2001:db8::/32      ::              1      100    32768  ?
       Route is advertised to 1 peers:
        2001:db8:4::110(65002)
```

For example, to display routes that match prefix 2001:db8::/32 or longer, enter the following command at any level of the CLI.

```
device# show ipv6 bgp 2001:db8::/32 longer-prefixes
Number of BGP Routes matching display condition : 3
Status codes: s suppressed, d damped, h history, * valid, > best, i internal
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network            Next Hop        MED LocPrf Weight Path
*>  2001:db8::/32      ::              1      100    32768  ?
*>  2001:db8:1234::/48 ::              1      100    32768  ?
*>  2001:db8:e0ff::/48 ::              1      100    32768  ?
       Route is advertised to 1 peers:
        2001:db8:4::110(65002)
```

TABLE 96 show ipv6 bgp output descriptions

This field...	Displays...
Total number of BGP Routes (appears in display of all BGP routes only)	The number of routes known by the device.
Number of BGP Routes matching display condition (appears in display that matches specified and longer prefixes)	The number of routes that matched the display parameters you entered. This is the number of routes displayed by the command.
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the left column of the display, to the left of each route. The status codes are described in the command's output.

TABLE 96 show ipv6 bgp output descriptions (Continued)

This field...	Displays...
Origin codes	A character the display uses to indicate the route's origin. The origin code appears to the right of the AS path (Path field). The origin codes are described in the command's output.
Network	The network prefix and prefix length.
Next Hop	The next-hop router for reaching the network from the device.
MED	The value of the route's MED attribute. If the route does not have a metric, this field is blank.
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 - 4294967295.
Weight	The value that this device associates with routes from a specific neighbor. For example, if the receives routes to the same destination from two BGP4+ neighbors, the prefers the route from the neighbor with the larger weight.
Path	The route's AS path.

Displaying BGP4+ route-attribute entries

The route-attribute entries table lists sets of BGP4+ attributes stored in the device's memory. Each set of attributes is unique and can be associated with one or more routes. In fact, the typically has fewer route attribute entries than routes.

To display the IPv6 route-attribute entries table, enter the following command.

```
device# show ipv6 bgp attribute-entries
Total number of BGP Attribute Entries: 378
1       Next Hop  :::                MED :1               Origin:INCOMP
        Originator:0.0.0.0           Cluster List:None
        Aggregator:AS Number :0      Router-ID:0.0.0.0          Atomic:None
        Local Pref:100               Communities:Internet
        AS Path   :(65002) 65001 4355 2548 3561 5400 6669 5548
        Address: 0x27a4cdb0  Hash:877 (0x03000000) Reference Counts: 2:0:2
...
```

NOTE
Portions of this display are truncated for brevity. The purpose of this display is to show all possible fields that might display rather than to show complete output.

Syntax: show ipv6 bgp attribute-entries

TABLE 97 show ipv6 bgp attribute-entries output descriptions

This field...	Displays...
Total number of BGP Attribute Entries	The number of entries contained in the device's BGP4+ route-attribute entries table.
Next Hop	The IPv6 address of the next hop router for routes that have this set of attributes.
MED	The cost of the routes that have this set of attributes.
Origin	The source of the route information. The origin can be one of the following: • EGP - The routes with this set of attributes came to BGP4+ through EGP. • IGP - The routes with this set of attributes came to BGP4+ through IGP. • INCOMPLETE - The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPFv3 or RIPng. When BGP4+ compares multiple routes to a destination to select the best route, IGP is preferred over EGP, and both are preferred over INCOMPLETE.
Originator	The originator of the route in a route-reflector environment.
Cluster List	The route-reflector clusters through which this set of attributes has passed.
Aggregator	Aggregator information: • AS Number shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise 0. • Router-ID shows the router that originated this aggregator.
Atomic	Whether the network information in this set of attributes has been aggregated and this aggregation has resulted in information loss: • TRUE - Indicates information loss has occurred • FALSE - Indicates no information loss has occurred • None - Indicates this attribute is not present. **NOTE** Information loss under these circumstances is a normal part of BGP4+ and does not indicate an error.
Local Pref	The degree of preference for routes that use this set of attributes relative to other routes in the local AS.
Communities	The communities that routes with this set of attributes are in.
AS Path	The ASs through which routes with this set of attributes have passed. The local AS is shown in parentheses.
Address	For debugging purposes only.
Hash	For debugging purposes only.

TABLE 97 show ipv6 bgp attribute-entries output descriptions (Continued)

This field...	Displays...
Reference Counts	For debugging purposes only.

Displaying the BGP4+ running configuration

To view the active BGP4+ configuration information contained in the running configuration without displaying the entire running configuration, enter the following command at any level of the CLI.

```
device# show ipv6 bgp config
Current BGP configuration:
router bgp
 local-as 1000
 neighbor peer_group1 peer-group
 neighbor 2001:db8:e0ff:783a::3 remote-as 1001
 neighbor 2001:db8:edd3:8389::1 remote-as 1002
 neighbor 2001:db8:80::23 peer-group peer_group1
 neighbor 2001:db8:80::23 remote-as 1003
 address-family ipv6 unicast
 no neighbor 2001:db8:e0ff:783a::3 activate
 no neighbor 2001:db8:edd3:8389::1 activate
 no neighbor 2001:db8:80::23 activate
 exit-address-family

 address-family vpnv4
 exit-address-family

 address-family l2vpn
 network 2001:db8::/32
 neighbor peer_group1 activate
 neighbor 2001:db8:edd3:8389::1 activate
 exit-address-family

end
```

Syntax: show ipv6 bgp config

Displaying dampened BGP4+ paths

To display BGP4+ paths that have been dampened (suppressed) by route flap dampening, enter the following command at any level of the CLI.

```
device# show ipv6 bgp dampened-paths
Status Code >:best d:damped h:history *:valid
    Network                From                    Flaps     Since     Reuse     Path
*d     2001:db8:8::/45     2001:db8:1::1     1     0 :1 :14     0 :2 :20     100
1002 1000
*d     2001:db8:1::/48     2001:db8:1::1     1     0 :1 :14     0 :2 :20     100
1002 1000
*d     2001:db8:4::/46     2001:db8:1::1     1     0 :1 :14     0 :2 :20     100
1002 1000
*d     2001:db8:2::/47     2001:db8:1::1     1     0 :1 :14     0 :2 :20     100
1002 1000
*d     2001:db8:0:8000::/49     2001:db8:1::1     1     0 :1 :14     0 :2 :20
100 1002 1000
*d     2001:db8:17::/64    2001:db8:1::1     1     0 :1 :18     0 :2 :20     100
```

Syntax: show ipv6 bgp dampened-paths

TABLE 98 show ipv6 bgp dampened-paths output descriptions

This field...	Displays...
Status codes	A list of the characters the display uses to indicate the path's status. The status code appears in the left column of the display, to the left of each route. The status codes are described in the command's output. The status column displays a "d" for each dampened route.
Network	The destination network of the route.
From	The IPv6 address of the advertising peer.
Flaps	The number of times the path has flapped.
Since	The amount of time (in hh:mm:ss) since the first flap of this route.
Reuse	The amount of time (in hh:mm:ss) after which the path is available again.
Path	The AS path of the route.

Displaying filtered-out BGP4+ routes

When you enable the soft reconfiguration feature, the device saves all updates received from the specified neighbor or peer group. The saved updates include those that contain routes that are filtered out by the BGP4+ route policies.

You can display a summary or more detailed information about routes that have been filtered out by BGP4+ route policies.

To display a summary of the routes that have been filtered out by BGP4+ route policies, enter the following command at any level of the CLI.

```
device# show ipv6 bgp filtered-routes
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix              Next Hop        MED LocPrf     Weight Status
1      2001:db8:3000::/48 2001:db8::110       100        0        EF
         AS_PATH: 65001 4355 701 80
2      2001:db8:4000::/48 2001:db8::110       100        0        EF
         AS_PATH: 65001 4355 1
3      2001:db8:5000::/48 2001:db8::110       100        0        EF
         AS_PATH: 65001 4355 701 1 189
```

The routes displayed by the command are the routes that the device's BGP policies filtered out. The did not place the routes in the BGP4+ route table, but did keep the updates. If a policy change causes these routes to be permitted, the user does not need to request the route information from the neighbor, but instead uses the information in the updates.

Syntax: show ipv6 bgp filtered-routes [*ipv6-prefix*|*prefix-length* [**longer-prefixes**] | [**as-path-access-list** *name*] | [**prefix-list** *name*]

The *ipv6-prefix* and *prefix-length* parameters display the specified IPv6 prefix of the destination network only. You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The **longer-prefixes** keyword allows you to display routes that match a specified or longer IPv6 prefix. For example, if you specify **2001:db8::/32 longer-prefixes**, then all routes with the prefix 2001:db8::/32 or that have a longer prefix (such as 2001:db8:e016::/48) are displayed.

The **as-path-access-list** *name* parameter specifies an AS-path ACL. Specify an ACL name. Only the routes permitted by the AS-path ACL are displayed.

The **prefix-list** *name* parameter specifies an IPv6 prefix list. Only the routes permitted by the prefix list are displayed.

TABLE 99 show ipv6 bgp filtered-routes output descriptions

This field...	Displays...
Number of BGP4+ Routes matching display condition	The number of routes that matched the display parameters you entered. This is the number of routes displayed by the command.
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the left column of the display, to the left of each route. The status codes are described in the command's output. The status column displays an "IF" for each filtered route.
Prefix	The network address and prefix.
Next Hop	The next-hop router for reaching the network from the device.
MED	The value of the route's MED attribute. If the route does not have a metric, this field is blank.
LocPrf	The degree of preference for this route relative to other routes in the local AS. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 - 4294967295.
Weight	The value that this device associates with routes from a specific neighbor. For example, if the receives routes to the same destination from two BGP4+ neighbors, the prefers the route from the neighbor with the larger weight.

TABLE 99 show ipv6 bgp filtered-routes output descriptions (Continued)

This field...	Displays...
Status	The route's status, which can be one or more of the following: • A - AGGREGATE - The route is an aggregate route for multiple networks. • B - BEST - BGP4+ has determined that this is the optimal route to the destination. • b - NOT-INSTALLED-BEST - BGP4+ has determined that this is the optimal route to the destination but did not install it in the IPv6 route table because the device received better routes from other sources (such as OSPFv3, RIPng, or static IPv6 routes). • C - CONFED_EBGP - The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D - DAMPED - This route has been dampened (by the route dampening feature), and is currently unusable. • E - EBGP - The route was learned through a in another AS. • H - HISTORY - Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I - IBGP - The route was learned through a in the same AS. • L - LOCAL - The route originated on this device. • M - MULTIPATH - BGP4+ load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE** If the "m" is shown in lowercase, the software was not able to install the route in the IPv6 route table. • S - SUPPRESSED - This route was suppressed during aggregation and thus is not advertised to neighbors. • F - FILTERED - This route was filtered out by BGP4+ route policies on the device, but the device saved updates containing the filtered routes.

To display detailed information about the routes that have been filtered out by BGP4+ route policies, enter the following command at any level of the CLI.

```
device# show ipv6 bgp filtered-routes detail
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
1     Prefix: 2001:db8:1::/64, Status: EF, Age: 0h0m10s
    NEXT_HOP: 2001:db8:1::1, Learned from Peer: 2001:db8:1::1 (100)
    LOCAL_PREF: 100, MED: 0, ORIGIN: incomplete, Weight: 0
    AS_PATH: 100
2     Prefix: 2001:db8:18::/64, Status: EF, Age: 0h0m10s
    NEXT_HOP: 2001:db8:1::1, Learned from Peer: 2001:db8:1::1 (100)
    LOCAL_PREF: 100, MED: 0, ORIGIN: incomplete, Weight: 0
    AS_PATH: 100
3     Prefix: 2001:db8:1::/64, Status: EF, Age: 0h0m10s
    NEXT_HOP: 2001:db8:1::1, Learned from Peer: 2001:db8:1::1 (100)
    LOCAL_PREF: 100, MED: 0, ORIGIN: incomplete, Weight: 0
    AS_PATH: 100
4     Prefix: 2001:db8:1::/64, Status: EF, Age: 0h0m10s
    NEXT_HOP: 2001:db8:1::1, Learned from Peer: 2001:db8:1::1 (100)
    LOCAL_PREF: 100, MED: 0, ORIGIN: incomplete, Weight: 0
    AS_PATH: 100
5     Prefix: 2001:db8:11::1/128, Status: EF, Age: 0h0m10s
    NEXT_HOP: 2001:db8:1::1, Learned from Peer: 2001:db8:1::1 (100)
    LOCAL_PREF: 100, MED: 0, ORIGIN: igp, Weight: 0
    AS_PATH: 100
6     Prefix: 2001:db8:17::/64, Status: EF, Age: 0h0m10s
    NEXT_HOP: 2001:db8:1::1, Learned from Peer: 2001:db8:1::1 (100)
```

```
LOCAL_PREF: 100, MED: 0, ORIGIN: incomplete, Weight: 0
AS_PATH: 100
```

Syntax: show ipv6 bgp filtered-routes detail [*ipv6-prefix*|*prefix-length* [**longer-prefixes**] | [**as-path-access-list** *name*] | [**prefix-list** *name*]

The *ipv6-prefix* and *prefix-length* parameters display the specified IPv6 prefix of the destination network only. You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The **longer-prefixes** keyword allows you to display routes that match a specified or longer IPv6 prefix. For example, if you specify 2001:db8::/32 longer-prefixes, then all routes with the prefix 2001:db8::/32 or that have a longer prefix (such as 2001:db8:e016::/48) are displayed.

The **as-path-access-list** *name* parameter specifies an AS-path ACL. Only the routes permitted by the AS-path ACL are displayed.

The **prefix-list** *name* parameter specifies an IPv6 prefix list. Only the routes permitted by the prefix list are displayed.

This display shows the following information.

TABLE 100 show ipv6 bgp filtered-routes detail output descriptions

This field...	Displays...
Status codes	A list of the characters the display uses to indicate the route's status. The Status field display an "F" for each filtered route.
Prefix	For information about this field, refer to Displaying filtered-out BGP4+ routes.
Status	For information about this field, refer to Displaying filtered-out BGP4+ routes.
Age	The age of the route, in seconds.
Next hop	For information about this field, refer to Displaying filtered-out BGP4+ routes.
Learned from peer	The IPv6 address of the neighbor from which this route is learned. "Local Router" indicates that the device itself learned the route.
Local pref	For information about this field, refer to Displaying filtered-out BGP4+ routes.
MED	The value of the advertised route's MED attribute. If the route does not have a metric, this field is blank.

TABLE 100 show ipv6 bgp filtered-routes detail output descriptions (Continued)

This field...	Displays...
Origin	The source of the route information. The origin can be one of the following: • A - AGGREGATE - The route is an aggregate route for multiple networks. • B - BEST - BGP4+ has determined that this is the optimal route to the destination. • b - NOT-INSTALLED-BEST - BGP4+ has determined that this is the optimal route to the destination but did not install it in the IPv6 route table because the device received better routes from other sources (such as OSPFv3, RIPng, or static IPv6 routes). • C - CONFED_EBGP - The route was learned from a neighbor in the same confederation and AS, but in a different sub-AS within the confederation. • D - DAMPED - This route has been dampened (by the route dampening feature), and is currently unusable. • E - EBGP - The route was learned through a in another AS. • H - HISTORY - Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I - IBGP - The route was learned through a in the same AS. • L - LOCAL - The route originated on this device. • M - MULTIPATH - BGP4+ load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE** If the "m" is shown in lowercase, the software was not able to install the route in the IPv6 route table. • S - SUPPRESSED - This route was suppressed during aggregation and thus is not advertised to neighbors. • F - FILTERED - This route was filtered out by BGP4+ route policies on the device, but the saved updates containing the filtered routes.
Weight	For information about this field, refer to Displaying filtered-out BGP4+ routes.
AS path	The ASs through which routes with this set of attributes have passed. The local AS is shown in parentheses.

Displaying route flap dampening statistics

To display route dampening statistics for all dampened routes, enter the following command at any level of the CLI.

```
device# show ipv6 bgp flap-statistics
Total number of flapping routes: 14
    Status Code  >:best d:damped h:history *:valid
    Network          From            Flaps  Since    Reuse    Path
h>  2001:db8:2::/48    2001:db8:23::47  1    0 :0 :13 0 :0 :0  65001 4355 1 701
*>  2001:db8:34::/48   2001:db8:23::47  1    0 :1 :4  0 :0 :0  65001 4355 701 62
```

Syntax: show ipv6 bgp flap-statistics [*ipv6-prefix*/*prefix-length* [**longer-prefixes**] | **as-path-filter** *number* | **neighbor** *ipv6-address* | **regular-expression** *regular-expression*]

The *ipv6-prefix* and *prefix-length* parameters display statistics for the specified IPv6 prefix only. You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as

documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The **longer-prefixes** keyword allows you to display statistics for routes that match a specified or longer IPv6 prefix. For example, if you specify 2001:db8::/32 longer-prefixes, then all routes with the prefix 2001:db8::/32 or that have a longer prefix (such as 2001db8::e016::/48) are displayed.

The **as-path-filter** *number* parameter specifies an AS path filter to display. Specify a filter number.

The **neighbor** *ipv6-address* parameter displays statistics for routes learned from the specified neighbor only. You also can display route flap statistics for routes learned from a neighbor by entering the following command: **show ipv6 bgp neighbor flap-statistics** .

The **regular-expression** *regular-expression* parameter is a regular expression. The regular expressions are the same ones supported for BGP4 AS-path filters.

You can also display route flap dampening statistics for a specified IPv6 neighbor. For more information, refer to Displaying route flap dampening statistics for a BGP4+ neighbor on page 531.

TABLE 101 show ipv6 bgp flap-statistics output descriptions

This field...	Displays...
Total number of flapping routes	The total number of routes in the device's BGP4+ route table that have changed state and thus have been marked as flapping routes.
Status code	Indicates the dampening status of the route, which can be one of the following: • > - This is the best route among those in the BGP4+ route table to the route's destination. • d - This route is currently dampened, and thus unusable. • h - The route has a history of flapping and is unreachable now. • * - The route has a history of flapping but is currently usable.
Network	The destination network of the route.
From	The IPv6 address of the advertising peer.
Flaps	The number of flaps (state changes) the route has experienced.
Since	The amount of time (in hh:mm:ss) since the first flap of this route.
Reuse	The amount of time (in hh:mm:ss) after which the path is again available.
Path	The AS path of the route.

You also can display all the dampened routes by using the **show ipv6 bgp dampened-paths** command. For more information, refer to Displaying dampened BGP4+ paths on page 514.

Displaying BGP4+ neighbor information

You can display the following information about a device's BGP4+ neighbors:

Configuration information and statistics:

- Router advertisements.
- Route-attribute entries.
- Route flap dampening statistics.
- The last packet containing an error.
- Received Outbound Route Filters (ORFs).
- Routes received from a neighbor.
- BGP4+ Routing Information Base (RIB).
- Received best, not installed best, and unreachable routes.
- Route summary.

Displaying IPv6 neighbor configuration information and statistics

To display BGP4+ neighbor configuration information and statistics, enter the following command at any level of the CLI.

```
device# show ipv6 bgp neighbor 2001:db8::110
1   IP Address: 2001:db8::110, AS: 65002 (EBGP), RouterID: 10.1.1.1
    State: ESTABLISHED, Time: 5d20h38m54s, KeepAliveTime: 60, HoldTime: 180
       RefreshCapability: Received
    Messages:    Open    Update  KeepAlive Notification Refresh-Req
       Sent    : 1       2       8012      0            0
       Received: 1       0       7880      0            0
    Last Update Time: NLRI       Withdraw          NLRI       Withdraw
                  Tx: ---        ---           Rx: ---        ---
    Last Connection Reset Reason:Unknown
    Notification Sent:     Unspecified
    Notification Received: Unspecified
    Neighbor NLRI Negotiation:
      Peer Negotiated IPV6 unicast  capability
      Peer configured for IPV6 unicast  Routes
    TCP Connection state: ESTABLISHED
       Byte Sent:   152411, Received: 149765
       Local host:  2001:db8::106, Local  Port: 8222
       Remote host: 2001:db8::110, Remote Port: 179
       ISentSeq:  740437769  SendNext:  740590181  TotUnAck:          0
       TotSent:      152412  ReTrans:           0  UnAckSeq:  740590181
       IRcvSeq:   242365900  RcvNext:   242515666  SendWnd:       16384
       TotalRcv:     149766  DupliRcv:          0  RcvWnd:        16384
       SendQue:           0  RcvQue:            0  CngstWnd:       1440
...
```

NOTE
Portions of this display are truncated for brevity. The purpose of this display is to show all possible fields that might display rather than to show complete output.

The display shows all the configured parameters for the neighbor. Only the parameters that have values different from their defaults are shown.

In this example, the number in the far left column indicates the neighbor for which information is displayed. When you list information for multiple neighbors, this number makes the display easier to read.

The TCP statistics at the end of the display show status for the TCP session with the neighbor. Most of the fields show information stored in the device's Transmission Control Block (TCB) for the TCP session between the device and its neighbor. These fields are described in detail in section 3.2 of RFC 793, "Transmission Control Protocol Functional Specification".

Syntax: show ipv6 bgp neighbor [*ipv6-address*]

The *ipv6-address* parameter allows you to display information for a specified neighbor only. You must specify the *ipv6-address* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

TABLE 102 show ipv6 bgp neighbor output descriptions

This field...	Displays...
IP Address	The IPv6 address of the neighbor.
AS	The AS in which the neighbor resides.
EBGP or IBGP	Whether the neighbor session is an IBGP session, an EBGP session, or a confederation EBGP session: • EBGP - The neighbor is in another AS. • EBGP_Confed - The neighbor is a member of another sub-AS in the same confederation. • IBGP - The neighbor is in the same AS.
RouterID	The neighbor's router ID.
State	The state of the device's session with the neighbor. The states are from the perspective of the session, not the neighbor's perspective. The state values can be one of the following: • IDLE - The BGP4+ process is waiting to be started. Usually, enabling BGP4 or establishing a neighbor session starts the BGP4+ process. - A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • ADMND - The neighbor has been administratively shut down. - A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • CONNECT - BGP4+ is waiting for the connection process for the TCP neighbor session to be completed. • ACTIVE - BGP4+ is waiting for a TCP connection from the neighbor. **NOTE** If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection. • OPEN SENT - BGP4+ is waiting for an Open message from the neighbor. • OPEN CONFIRM - BGP4+4 has received an OPEN message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. If the device receives a KEEPALIVE message from the neighbor, the state changes to Established. If the message is a NOTIFICATION, the state changes to Idle. • ESTABLISHED - BGP4+ is ready to exchange UPDATE messages with the neighbor. - If there is more BGP data in the TCP receiver queue, a plus sign (+) is also displayed. **NOTE** If you display information for the neighbor using the **show ipv6 bgp neighbor**<*ipv6-address*> command, the TCP receiver queue value will be greater than 0.

TABLE 102 show ipv6 bgp neighbor output descriptions (Continued)

This field...	Displays...
Time	The amount of time this session has been in its current state.
KeepAliveTime	The keep alive time, which specifies how often this device sends keep alive messages to the neighbor.
HoldTime	The hold time, which specifies how many seconds the device will wait for a KEEPALIVE or UPDATE message from a BGP4+ neighbor before deciding that the neighbor is dead.
RefreshCapability	Whether the device has received confirmation from the neighbor that the neighbor supports the dynamic refresh capability.
Messages Sent and Received	The number of messages this device has sent to and received from the neighbor. The display shows statistics for the following message types: • Open • Update • KeepAlive • Notification • Refresh-Req
Last Update Time	Lists the last time updates were sent and received for the following: • NLRIs • Withdraws

TABLE 102 show ipv6 bgp neighbor output descriptions (Continued)

This field...	Displays...
Last Connection Reset Reason	The reason the previous session with this neighbor ended. The reason can be one of the following: • No abnormal error has occurred. • Reasons described in the BGP specifications: - Message Header Error - Connection Not Synchronized - Bad Message Length - Bad Message Type - OPEN Message Error - Unsupported Version Number - Bad Peer AS Number - Bad BGP Identifier - Unsupported Optional Parameter - Authentication Failure - Unacceptable Hold Time - Unsupported Capability - UPDATE Message Error - Malformed Attribute List - Unrecognized Well-known Attribute - Missing Well-known Attribute - Attribute Flags Error - Attribute Length Error - Invalid ORIGIN Attribute - Invalid NEXT_HOP Attribute - Optional Attribute Error - Invalid Network Field - Malformed AS_PATH - Hold Timer Expired - Finite State Machine Error - Rcv Notification
Last Connection Reset Reason (cont.)	• Reasons specific to the implementation: - Reset All Peer Sessions - User Reset Peer Session - Port State Down - Peer Removed - Peer Shutdown - Peer AS Number Change - Peer AS Confederation Change - TCP Connection KeepAlive Timeout - TCP Connection Closed by Remote - TCP Data Stream Error Detected

TABLE 102 show ipv6 bgp neighbor output descriptions (Continued)

This field...	Displays...
Notification Sent	If the device receives a NOTIFICATION message from the neighbor, the message contains an error code corresponding to one of the following errors. Some errors have subcodes that clarify the reason for the error. Where applicable, the subcode messages are listed underneath the error code messages. • Message Header Error - Connection Not Synchronized - Bad Message Length - Bad Message Type - Unspecified • Open Message Error - Unsupported Version - Bad Peer As - Bad BGP Identifier - Unsupported Optional Parameter - Authentication Failure - Unacceptable Hold Time - Unspecified • Update Message Error - Malformed Attribute List - Unrecognized Attribute - Missing Attribute - Attribute Flag Error - Attribute Length Error - Invalid Origin Attribute - Invalid NextHop Attribute - Optional Attribute Error - Invalid Network Field - Malformed AS Path - Unspecified • Hold Timer Expired • Finite State Machine Error • Cease • Unspecified
Notification Received	See above.
Neighbor NLRI Negotiation	The state of the device's NLRI negotiation with the neighbor. The states can include the following: • Peer negotiated IPv6 unicast capability. • Peer configured for IPv6 unicast routes. • Peer negotiated IPv4 unicast capability. • Peer negotiated IPv4 multicast capability.

TABLE 102 show ipv6 bgp neighbor output descriptions (Continued)

This field...	Displays...
TCP Connection state	The state of the connection with the neighbor. The connection can have one of the following states: • LISTEN - Waiting for a connection request. • SYN-SENT - Waiting for a matching connection request after having sent a connection request. • SYN-RECEIVED - Waiting for a confirming connection request acknowledgment after having both received and sent a connection request. • ESTABLISHED - Data can be sent and received over the connection. This is the normal operational state of the connection. • FIN-WAIT-1 - Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent. • FIN-WAIT-2 - Waiting for a connection termination request from the remote TCP. • CLOSE-WAIT - Waiting for a connection termination request from the local user. • CLOSING - Waiting for a connection termination request acknowledgment from the remote TCP. • LAST-ACK - Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request). • TIME-WAIT - Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request. • CLOSED - There is no connection state.
Byte Sent	The number of bytes sent.
Byte Received	The number of bytes received.
Local host	The IPv6 address of the device.
Local port	The TCP port the Brocade device is using for the BGP4+ TCP session with the neighbor.
Remote host	The IPv6 address of the neighbor.
Remote port	The TCP port the neighbor is using for the BGP4+ TCP session with the device.
ISentSeq	The initial send sequence number for the session.
SendNext	The next sequence number to be sent.
TotUnAck	The number of sequence numbers sent by the device that have not been acknowledged by the neighbor.
TotSent	The number of sequence numbers sent to the neighbor.
ReTrans	The number of sequence numbers that the device retransmitted because they were not acknowledged.
UnAckSeq	The current acknowledged sequence number.

TABLE 102 show ipv6 bgp neighbor output descriptions (Continued)

This field...	Displays...
IRcvSeq	The initial receive sequence number for the session.
RcvNext	The next sequence number expected from the neighbor.
SendWnd	The size of the send window.
TotalRcv	The number of sequence numbers received from the neighbor.
DupliRcv	The number of duplicate sequence numbers received from the neighbor.
RcvWnd	The size of the receive window.
SendQue	The number of sequence numbers in the send queue.
RcvQue	The number of sequence numbers in the receive queue.
CngstWnd	The number of times the window has changed.

Displaying routes advertised to a BGP4+ neighbor

You can display a summary or detailed information about the following:

- All routes a device has advertised to a neighbor.
- A specified route a device has advertised to a neighbor.

For example, to display a summary of all routes a device has advertised to neighbor 2001.db8::110, enter the following command at any level of the CLI.

```
device# show ipv6 bgp neighbor 2001:db8::110
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST E:EBGP I:IBGP L:LOCAL
       Prefix              Next Hop  MED LocPrf   Weight Status
1      2001:db8:1234::/48 ::        1             32768  BL
         AS_PATH:
2      2001:db8:2002::/48 ::        1             32768  BL
         AS_PATH:
```

Syntax: show ipv6 bgp neighbor *ipv6-address* **advertised-routes** [**detail**] *ipv6-prefix*/*prefix-length*

The *ipv6-address* parameter displays routes advertised to a specified neighbor. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **detail** keyword displays detailed information about the advertised routes. If you do not specify this keyword, a summary of the advertised routes displays.

The *ipv6-prefix* and *prefix-length* parameters display the specified route advertised to the neighbor only. You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

TABLE 103 show ipv6 bgp neighbor advertised-routes output descriptions

This field...	Displays...
Number of BGP4+ Routes advertised to specified neighbor (appears only in display for all routes)	The number of routes displayed by the command.
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the Status column of the display. The status codes are described in the command's output.
Prefix	The advertised route's prefix.
Next Hop	The next-hop for reaching the advertised route from the device.
MED	The value of the advertised route's MED attribute. If the route does not have a metric, this field is blank.
LocPrf	The degree of preference for the advertised route relative to other routes in the local autonomous system. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference range is 0 - 4294967295.
Weight	The value that this device associates with routes from a specific neighbor. For example, if the receives routes to the same destination from two BGP4+ neighbors, the prefers the route from the neighbor with the larger weight.
Status	The advertised route's status, which can be one or more of the following: • A - AGGREGATE. The route is an aggregate route for multiple networks. • B - BEST. BGP4+ has determined that this is the optimal route to the destination. • b - NOT-INSTALLED-BEST - BGP4+ has determined that this is the optimal route to the destination but did not install it in the IPv6 route table because the device received better routes from other sources (such as OSPFv3, RIPng, or static IPv6 routes). • E - EBGP. The route was learned through a in another AS. • I - IBGP. The route was learned through a in the same AS. • L - LOCAL. The route originated on this device.
AS-PATH	The AS-path information for the route.

For example, to display details about all routes a device has advertised to neighbor 2001:db8::110, enter the following command at any level of the CLI..

```
device# show ipv6 bgp neighbor 2001:db8::110 advertised-routes detail
There are 2 routes advertised to neighbor 2001:db8::110
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST E:EBGP I:IBGP L:LOCAL
1       Prefix: 2001:db8:1::/48,  Status: BL,  Age: 6d13h28m7s
         NEXT_HOP: 2001:db8::106,  Learned from Peer: Local Router
          LOCAL_PREF: none,  MED: 1,  ORIGIN: incomplete,  Weight: 32768
           AS_PATH:
            Adj_RIB_out count: 1,  Admin distance 190
2       Prefix: 2001:db8::/32,  Status: BL,  Age: 6d13h31m22s
         NEXT_HOP: 2001:db8::106,  Learned from Peer: Local Router
          LOCAL_PREF: none,  MED: 1,  ORIGIN: incomplete,  Weight: 32768
           AS_PATH:
```

```
Adj_RIB_out count: 1,  Admin distance 190
```

This display shows the following information.

TABLE 104 show ipv6 bgp neighbor advertised-routes detail output descriptions

This field...	Displays...
Number of BGP4+ Routes advertised to specified neighbor (appears only in display for all routes)	For information about this field, refer to the table above.
Status codes	For information about this field, refer to Displaying routes advertised to a BGP4+ neighbor.
Prefix	For information about this field, refer to the table above.
Status	For information about this field, refer to the table above.
Age	The age of the advertised route, in seconds.
Next Hop	For information about this field, refer to the table above.
Learned from Peer	The IPv6 address of the neighbor from which this route is learned. "Local Router" indicates that the device itself learned the route.
LOCAL_PREF	For information about this field, refer to the table above.
MED	The value of the advertised route's MED attribute. If the route does not have a metric, this field is blank.
Origin	The source of the route information. The origin can be one of the following: • EGP - The routes with this set of attributes came to BGP4+ through EGP. • IGP - The routes with this set of attributes came to BGP4+ through IGP. • INCOMPLETE - The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPFv3 or RIPng. When BGP4+ compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Weight	For information about this field, refer to the table above.
AS-PATH	The AS-path information for the route.
Adj RIB out count	The number of routes in the device's current BGP4+ Routing Information Base (Adj-RIB-Out) for a specified neighbor.
Admin distance	The administrative distance of the route.

Displaying routes advertised to a BGP4+ neighbor

The route-attribute entries table lists sets of BGP4+ attributes stored in the device's memory. Each set of attributes is unique and can be associated with one or more routes. In fact, the typically has fewer route attribute entries than routes.

For example, to display the route-attribute entries table for a BGP4+ neighbor 2001:db8::110, enter the following command.

```
device# show ipv6 bgp neighbor
Total number of BGP Attribute Entries: 1
1       Next Hop  :2001:db8::106       Metric   :1           Origin:INCOMP
        Originator:0.0.0.0           Cluster List:None
        Aggregator:AS Number :0     Router-ID:0.0.0.0       Atomic:None
        Local Pref:100              Communities:Internet
        AS Path   :65001
        Address: 0x26579354 Hash:332(0x0301fcd4) Reference Counts: 2:0:0
```

Syntax: show ipv6 bgp neighbor *ipv6-address* **attribute-entries**

The *ipv6-address* parameter displays the route attribute entries for a specified neighbor. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

TABLE 105 show ipv6 bgp neighbor attribute-entries output descriptions

This field...	Displays...
Total number of BGP Attribute Entries	The number of route attribute entries for the specified neighbor.
Next Hop	The IPv6 address of the next hop router for routes that have this set of attributes.
Metric	The cost of the routes that have this set of attributes.
Origin	The source of the route information. The origin can be one of the following: • E - EBGP. The routes with this set of attributes came to BGP4+ through EGP. • I - IBGP. The routes with this set of attributes came to BGP4+ through IGP. • I - INCOMPLETE. The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPFv3 or RIPng. When BGP4+ compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Originator	The originator of the route in a route reflector environment.
Cluster List	The route-reflector clusters through which this set of attributes has passed.
Aggregator	Aggregator information: • AS Number shows the AS in which the network information in the attribute set was aggregated. This value applies only to aggregated routes and is otherwise 0. • outer-ID shows the that originated this aggregator.

TABLE 105 show ipv6 bgp neighbor attribute-entries output descriptions (Continued)

This field...	Displays...
Atomic	Whether the network information in this set of attributes has been aggregated *and* this aggregation has resulted in information loss: • TRUE – Indicates information loss has occurred • FALSE – Indicates no information loss has occurred • None – Indicates the attribute is not present **NOTE** Information loss under these circumstances is a normal part of BGP4+ and does not indicate an error.
Local Pref	The degree of preference for routes that use this set of attributes relative to other routes in the local AS.
Communities	The communities that routes with this set of attributes are in.
AS Path	The ASs through which routes with this set of attributes have passed. The local AS is shown in parentheses.
Address	For debugging purposes only.
Hash	For debugging purposes only.
Reference Counts	For debugging purposes only.

Displaying route flap dampening statistics for a BGP4+ neighbor

To display route flap dampening statistics for a specified BGP4+ neighbor, enter the following command at any level of the CLI.

```
device# show ipv6 bgp neighbor 2001:db8::110 flap-statistics
Total number of flapping routes: 14
    Status Code  >:best d:damped h:history *:valid
    Network            From            Flaps Since    Reuse    Path
h>  2001:db8:2::/48    10.90.213.77    1     0 :0 :13 0 :0 :0  65001 4355 1 701
*>  2001:db8:34::/48   10.90.213.77    1     0 :1 :4  0 :0 :0  65001 4355 701 62
```

Syntax: show ipv6 bgp neighbor *ipv6-address* **flap-statistics**

The *ipv6-address* parameter displays the route flap dampening statistics for a specified neighbor. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

TABLE 106 show ipv6 bgp neighbor flap-statistics output descriptions

This field...	Displays...
Total number of flapping routes	The total number of routes in the neighbor's BGP4+ route table that have changed state and thus have been marked as flapping routes.

TABLE 106 show ipv6 bgp neighbor flap-statistics output descriptions (Continued)

This field...	Displays...
Status code	Indicates the status of the route, which can be one of the following: • > - This is the best route among those in the neighbor's BGP4+ route table to the route's destination. • d - This route is currently dampened, and thus unusable. • h - The route has a history of flapping and is unreachable now. • * - The route has a history of flapping but is currently usable.
Network	The destination network of the route.
From	The IPv6 address of the advertising peer.
Flaps	The number of flaps (state changes) the route has experienced.
Since	The amount of time (in hh:mm:ss) since the first flap of this route.
Reuse	The amount of time (in hh:mm:ss) after which the path is again available.
Path	The AS path of the route.

You also can display all the dampened routes by using the **show ipv6 bgp dampened-paths** command. For more information, refer to Displaying dampened BGP4+ paths on page 514.

Displaying last error packet from a BGP4+ neighbor

You can display information about the last packet that contained an error from any of a device's neighbors. The displayed information includes the error packet's contents decoded in a human-readable format.

For example, to display information about the last error packet from any of a device's neighbors, enter the following command.

```
device# show ipv6 bgp neighbor last-packet-with-error
Total number of BGP Neighbors: 266
No received packet with error logged for any neighbor
```

Syntax: show ipv6 bgp neighbor last-packet-with-error

TABLE 107 show ipv6 bgp neighbor last-packet-with-error output descriptions

This field...	Displays...
Total number of BGP Neighbors	The total number of configured neighbors for a device.
Last error	The error packet's contents decoded in a human-readable format or notification that no packets with an error were received.

Displaying Outbound Route Filters received from a BGP4+ neighbor

You can display the Outbound Route Filters (ORFs) received from a BGP4+ neighbor. This option applies to cooperative route filtering feature.

For example, to display the ORFs received from neighbor 2001:db8::110, enter the following command.

```
device# show ipv6 bgp neighbor 2001:db8::110 received prefix-filter
ip prefix-list 2001:db8::110: 4 entries
     seq 5 permit 2001:db8:3::45/16 ge 18 le 28
     seq 10 permit 2001:db8::4::88/24
     seq 15 permit 2001:db8:5::37/8 le 32
     seq 20 permit 2001:db8:6::83/16 ge 18
```

Syntax: show ipv6 bgp neighbor *ipv6-address* **received prefix-filter**

The *ipv6-address* parameter displays the prefix filter learned from a specified neighbor. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

Displaying routes received from a BGP4+ neighbor

You can display a summary or detailed route information received in route updates from a specified BGP4+ neighbor since you enabled the soft reconfiguration feature.

For example, to display a summary of the route information received in route updates from neighbor 2001:db8::10, enter the following command at any level of the CLI.

```
Brocade# show ipv6 bgp neighbor 2001:db8::10 received-routes
There are 4 received routes from neighbor 2001:db8::10
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
    Prefix    Next Hop    Metric    LocPrf    Weight    Status
1     2001:db8:2002::/64    2001:db8::10     0    100    0    BE
AS_PATH: 400
2     2001:db8:2003::/64    2001:db8::10     1    100    0    BE
AS_PATH: 400
3     2001:db8:2004::/64    2001:db8::10     1    100    0    BE
AS_PATH: 400
4     2001:db8:2005::/64    2001:db8::10     1    100    0    BE
AS_PATH: 400
```

Syntax: show ipv6 bgp neighbor *ipv6-address* **received-routes** [**detail**]

The *ipv6-address* parameter displays route information received from a specified neighbor. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **detail** keyword displays detailed route information. If you do not specify this parameter, a summary of route information displays.

TABLE 108 show ipv6 bgp neighbor received-routes output descriptions

This field...	Displays...
Number of BGP4+ Routes received from a neighbor	The number of routes displayed by the command.
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the Status column of the display. The status codes are described in the command's output.
Prefix	The received route's prefix.

TABLE 108 show ipv6 bgp neighbor received-routes output descriptions (Continued)

This field...	Displays...
Next Hop	The IPv6 address of the next device that is used when forwarding a packet to the received route.
MED	The value of the route's MED attribute. If the route does not have a metric, this field is blank.
LocPrf	The degree of preference for the advertised route relative to other routes in the local autonomous system. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 - 4294967295.
Weight	The value that this device associates with routes from a specific neighbor. For example, if the receives routes to the same destination from two BGP4+ neighbors, the prefers the route from the neighbor with the larger weight.
Status	The advertised route's status, which can be one or more of the following: A - AGGREGATE. The route is an aggregate route for multiple networks. B - BEST. BGP4+ has determined that this is the optimal route to the destination. b - NOT-INSTALLED-BEST - BGP4+ has determined that this is the optimal route to the destination but did not install it in the IPv6 route table because the device received better routes from other sources (such as OSPFv3, RIPng, or static IPv6 routes). D - DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. E - EBGP. The route was learned through a in another AS. H - HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. I - IBGP. The route was learned through a in the same autonomous system. L - LOCAL. The route originated on this device. M - MULTIPATH. BGP4+ load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE** If the "m" is shown in lowercase, the software was not able to install the route in the IPv6 route table. S - SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors. F - FILTERED. This route was filtered out by BGP4+ route policies on the device, but the saved updates containing the filtered routes.

For example, to display details about routes received from neighbor 2001:db8:1::1, enter the following command at any level of the CLI.

```
device# show ipv6 bgp neighbor 2001:db8:1::1 received-routes detail
There are 4 received routes from neighbor 2001:db8:1::1
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
```

```
E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
1 Prefix: 2001:db8:1000:1::/64, Status: BI, Age: 0h17m25s
NEXT_HOP: 2001:db8:1::1, Learned from Peer: 2001:db8:1::1 (100)
LOCAL_PREF: 100, MED: 0, ORIGIN: incomplete, Weight: 0
AS_PATH:
Adj_RIB_out count: 1, Admin distance 200
2 Prefix: 2001:db8:1::/64, Status: I, Age: 0h17m25s
NEXT_HOP: 2001:db8:1::1, Learned from Peer: 2001:db8:1::1 (100)
LOCAL_PREF: 100, MED: 0, ORIGIN: incomplete, Weight: 0
AS_PATH:
3 Prefix: 2001:db8:11::1/128, Status: BI, Age: 0h17m25s
NEXT_HOP: 2001:db8:1::1, Learned from Peer: 2001:db8:1::1 (100)
LOCAL_PREF: 100, MED: 0, ORIGIN: igp, Weight: 0
AS_PATH:
Adj_RIB_out count: 1, Admin distance 200
4 Prefix: 2001:db8:17::/64, Status: BI, Age: 0h17m25s
NEXT_HOP: 2001:db8:1::1, Learned from Peer: 2001:db8:1::1 (100)
LOCAL_PREF: 100, MED: 0, ORIGIN: incomplete, Weight: 0
AS_PATH:
Adj_RIB_out count: 1, Admin distance 200
```

TABLE 109 show ipv6 bgp neighbor received-routes detail output descriptions

This field...	Displays...
Number of BGP4+ routes received from a neighbor	For information about this field, refer to the table above.
Status codes	For information about this field, refer to the table above.
Prefix	For information about this field, refer to the table above.
Status	For information about this field, refer to the table above.
Age	The age of the route, in seconds.
Next hop	The next-hop router for reaching the route from the device.
Learned from peer	The IPv6 address of the neighbor from which this route is learned. "Local Router" indicates that the device itself learned the route.
Local pref	For information about this field, refer to the table above.
MED	The value of the route's MED attribute. If the route does not have a metric, this field is blank.
Origin	The source of the route information. The origin can be one of the following: • EGP - The routes with this set of attributes came to BGP4+ through EGP. • IGP - The routes with this set of attributes came to BGP4+ through IGP. • INCOMPLETE - The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPFv3 or RIPng. When BGP4+ compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Weight	For information about this field, refer to the table above.
AS Path	For information about this field, refer to the table above.

TABLE 109 show ipv6 bgp neighbor received-routes detail output descriptions (Continued)

This field...	Displays...
Adj RIB out count	The number of routes in the device's current BGP4+ Routing Information Base (Adj-RIB-Out) for a specified neighbor.
Admin distance	The administrative distance of the route.

Displaying the Adj-RIB-Out for a BGP4+ neighbor

You can display a summary or detailed information about the following:

- All routes in a device's current BGP4+ Routing Information Base (Adj-RIB-Out) for a specified neighbor.
- A specified route in a device's current BGP4+ RIB for a specified neighbor.

The RIB contains the routes that the device either has most recently sent to the neighbor or is about to send to the neighbor.

For example, to display a summary of all routes in a device's RIB for neighbor 2001:db8::110, enter the following command at any level of the CLI.

```
device# show ipv6 bgp neighbor 2001:db8::110 rib-out-routes
       There are 2 RIB_out routes for neighbor 2001:db8::110
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST E:EBGP I:IBGP L:LOCAL
       Prefix              Next Hop     Metric    LocPrf     Weight Status
1      2001:db8:1234::/48  ::           1         100        32768  BL
         AS_PATH:
2      2001:db8:2002::/48  ::           1         100        32768  BL
         AS_PATH:
```

Syntax: show ipv6 bgp neighbor *ipv6-address* **rib-out-routes** [*ipv6-prefix*|*prefix-length* | **detail** [*ipv6-prefix*|*prefix-length network-mask*]]

The **ipv6-address** parameter displays the RIB routes for a specified neighbor. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The *ipv6-prefix* and *prefix-length* parameters display the specified RIB route for the neighbor. You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The **detail** *ipv6-prefix*, *prefix-length*, and *network-mask* parameters display detailed information about the specified RIB routes. If you do not specify this parameter, a summary of the RIB routes displays. You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter. You must specify the *network-mask* parameter using 8-bit values in dotted decimal notation.

TABLE 110 show ipv6 bgp neighbor rib-out-routesoutput descriptions

This field...	Displays...
Number of RIB_out routes for a specified neighbor (appears only in display for all RIB routes)	The number of RIB routes displayed by the command.

TABLE 110 show ipv6 bgp neighbor rib-out-routesoutput descriptions (Continued)

This field...	Displays...
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the Status column of the display. The status codes are described in the command's output.
Prefix	The RIB route's prefix.
Next Hop	The next-hop router for reaching the route from the device.
MED	The value of the advertised route's MED attribute. If the route does not have a metric, this field is blank.
LocPrf	The degree of preference for the route relative to other routes in the local autonomous system. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 - 4294967295.
Weight	The value that this device associates with routes from a specific neighbor. For example, if the receives routes to the same destination from two BGP4+ neighbors, the prefers the route from the neighbor with the larger weight.
Status	The RIB route's status, which can be one or more of the following: • A - AGGREGATE. The route is an aggregate route for multiple networks. • B - BEST. BGP4+ has determined that this is the optimal route to the destination. • E - EBGP. The route was learned through a in another autonomous system. • I - IBGP. The route was learned through a in the same autonomous system. • L - LOCAL. The route originated on this device.
AS-PATH	The AS-path information for the route.

For example, to display details about all RIB routes for neighbor 2001:db8::110,, enter the following command at any level of the CLI.

```
device# show ipv6 bgp neighbor 2001:db8::110 rib-out-routes detail
                    There are 2 RIB_out routes for neighbor 2001:db8::110
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST E:EBGP I:IBGP L:LOCAL
1       Prefix: 2001:db8:1234::/48,  Status: BL,  Age: 6d18h17m53s
         NEXT_HOP: ::,  Learned from Peer: Local Router
          LOCAL_PREF: 100,  MED: 1,  ORIGIN: incomplete,  Weight: 32768
           AS_PATH:
            Adj_RIB_out count: 1,  Admin distance 190
2       Prefix: 2001:db8:2002::/48,  Status: BL,  Age: 6d18h21m8s
         NEXT_HOP: ::,  Learned from Peer: Local Router
          LOCAL_PREF: 100,  MED: 1,  ORIGIN: incomplete,  Weight: 32768
           AS_PATH:

            Adj_RIB_out count: 1,  Admin distance 190
            Adj_RIB_out count: 1,  Admin distance 190
```

TABLE 111 show ipv6 bgp neighbor rib-out-routes detail output descriptions

This field...	Displays...
Number of RIB_out routes for a specified neighbor (appears only in display for all routes)	For information about this field, refer to the table above.
Status codes	For information about this field, refer to the table above.
Prefix	For information about this field, refer to the table above.
Status	For information about this field, refer to the table above.
Age	The age of the RIB route, in seconds.
Next Hop	For information about this field, refer to the table above.
Learned from Peer	The IPv6 address of the neighbor from which this route is learned. "Local Router" indicates that the device itself learned the route.
LOCAL_PREF	For information about this field, refer to the table above.
MED	The value of the RIB route's MED attribute. If the route does not have a metric, this field is blank.
Origin	The source of the route information. The origin can be one of the following: • EGP - The routes with this set of attributes came to BGP4+ through EGP. • IGP - The routes with this set of attributes came to BGP4+ through IGP. • INCOMPLETE - The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPFv3 or RIPng. When BGP4+ compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Weight	For information about this field, refer to the table above.
AS-PATH	For information about this field, refer to the table above.

Displaying the best and unreachable routes received from a BGP4+ neighbor

You can display a summary or detailed information about the following types of BGP4+ routes received from a specified neighbor:

- Best routes - The "best" routes to their destinations, which are installed in the device's IPv6 route table.
- Unreachable - The routes whose destinations are unreachable using any of the BGP4+ paths in the IPv6 route table.

For example, to display a summary of the best routes to a destination received from neighbor 2001:db8::106, enter the following command.

```
device# show ipv6 bgp neighbor 2001:db8::106 routes best
       There are 2 accepted routes from neighbor 2001:db8::106
```

```
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
       Prefix                  Next Hop        MED LocPrf     Weight Status
1      2001:db8:2002::/48      2001:db8::106   1      100         0      BE
         AS_PATH: 65001
2      2001:db8:2002:1234::/64 2001:db8::106   1      100         0      BE
         AS_PATH: 65001
```

Syntax: show ipv6 bgp neighbor *ipv6-address* **routes best** | **detail** [**best** | **unreachable**] | **unreachable**

The *ipv6-address* parameter displays the routes for a specified neighbor. You must specify this address in hexadecimal using 16-bit values between colons as documented in RFC 2373.

The **best** keyword displays the "best" routes, which are installed in the IPv6 route table.

The **unreachable** keyword displays the routes whose destinations are unreachable using any of the BGP4+ paths in the IPv6 route table.

The **detail** keyword displays detailed information about the routes. If you do not specify this parameter, a summary of the routes displays.

This display shows the following information.

TABLE 112 show ipv6 bgp neighbor routes best output descriptions

This field...	Displays...
Number of accepted routes from a specified neighbor	The number of routes displayed by the command.
Status codes	A list of the characters the display uses to indicate the route's status. The status code appears in the Status column of the display. The status codes are described in the command's output.
Prefix	The route's prefix.
Next Hop	The next-hop router for reaching the route from the device.
MED	The value of the route's MED attribute. If the route does not have a metric, this field is blank.
LocPrf	The degree of preference for the route relative to other routes in the local autonomous system. When the BGP4+ algorithm compares routes on the basis of local preferences, the route with the higher local preference is chosen. The preference can have a value from 0 - 4294967295.
Weight	The value that this device associates with routes from a specific neighbor. For example, if the receives routes to the same destination from two BGP4+ neighbors, the prefers the route from the neighbor with the larger weight.

TABLE 112 show ipv6 bgp neighbor routes best output descriptions (Continued)

This field...	Displays...
Status	The route's status, which can be one or more of the following: • A - AGGREGATE. The route is an aggregate route for multiple networks. • B - BEST. BGP4+ has determined that this is the optimal route to the destination. • C - CONFED_EBGP. The route was learned from a neighbor in the same confederation and autonomous system, but in a different sub-AS within the confederation. • D - DAMPED. This route has been dampened (by the route dampening feature), and is currently unusable. • E - EBGP. The route was learned through a in another autonomous system. • H - HISTORY. Route dampening is configured for this route, and the route has a history of flapping and is unreachable now. • I - IBGP. The route was learned through a in the same autonomous system. • L - LOCAL. The route originated on this device. • M - MULTIPATH. BGP4+ load sharing is enabled and this route was selected as one of the best ones to the destination. The best route among the multiple paths also is marked with "B". **NOTE** If the "m" is shown in lowercase, the software was not able to install the route in the IPv6 route table. • S - SUPPRESSED. This route was suppressed during aggregation and thus is not advertised to neighbors. • F - FILTERED. This route was filtered out by BGP4+ route policies on the device, but the saved updates containing the filtered routes.
AS-PATH	The AS-path information for the route.

For example, to display detailed information about the best routes to a destination received from neighbor 2001:db8::106, enter the following command.

```
device# show ipv6 bgp neighbor 2000:4::106 routes detail best
       There are 2 accepted routes from neighbor 2000:4::106
Searching for matching routes, use ^C to quit...
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED
       E:EBGP H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED F:FILTERED
1       Prefix: 2001:db8::/32,  Status: BE,  Age: 18h48m56s
         NEXT_HOP: 2001:db8::106,  Learned from Peer: 2001:db8::106 (65001)
          LOCAL_PREF: 100,  MED: 1,  ORIGIN: incomplete,  Weight: 0
           AS_PATH: 65001
2       Prefix: 2001:db8:1234::/48,  Status: BE,  Age: 18h48m56s
         NEXT_HOP: 2001:db8::106,  Learned from Peer: 2001:db8::106 (65001)
          LOCAL_PREF: 100,  MED: 1,  ORIGIN: incomplete,  Weight: 0
           AS_PATH: 65001
```

TABLE 113 show ipv6 bgp neighbor routes detail bestoutput descriptions

This field...	Displays...
Number of accepted routes from a specified neighbor (appears only in display for all routes)	For information about this field, refer to the table above.

TABLE 113 show ipv6 bgp neighbor routes detail bestoutput descriptions (Continued)

This field...	Displays...
Status codes	For information about this field, refer to the table above.
Prefix	For information about this field, refer to the table above.
Status	For information about this field, refer to the table above.
Age	The age of the route, in seconds.
Next Hop	For information about this field, refer to the table above.
Learned from Peer	The IPv6 address of the neighbor from which this route is learned. "Local Router" indicates that the device itself learned the route.
LOCAL_PREF	For information about this field, refer to the table above.
MED	The value of the RIB route's MED attribute. If the route does not have a metric, this field is blank.
Origin	The source of the route information. The origin can be one of the following: • EGP - The routes with this set of attributes came to BGP4+ through EGP. • IGP - The routes with this set of attributes came to BGP4+ through IGP. • INCOMPLETE - The routes came from an origin other than one of the above. For example, they may have been redistributed from OSPFv3 or RIPng. When BGP4+ compares multiple routes to a destination to select the best route, IGP is preferred over EGP and both are preferred over INCOMPLETE.
Weight	For information about this field, refer to the table above.
AS-PATH	For information about this field, refer to the table above.

Displaying IPv6 neighbor route summary information

You can display route summary information for all neighbors or a specified neighbor only.

For example, to display summary information for neighbor 2001:db8::110, enter the following command at any level of the CLI.

```
device# show ipv6 bgp neighbor 2001:db8::110 routes-summary
1   IP Address: 2001:db8::110
Routes Accepted/Installed:0,  Filtered/Kept:0,  Filtered:0
   Routes Selected as BEST Routes:0
      BEST Routes not Installed in IP Forwarding Table:0
   Unreachable Routes (no IGP Route for NEXTHOP):0
   History Routes:0

NLRIs Received in Update Message:0,  Withdraws:0 (0),  Replacements:0
   NLRIs Discarded due to
      Maximum Prefix Limit:0, AS Loop:0
      Invalid Nexthop:0, Invalid Nexthop Address:0.0.0.0
      Duplicated Originator_ID:0, Cluster_ID:0

Routes Advertised:2,  To be Sent:0,  To be Withdrawn:0
NLRIs Sent in Update Message:2,  Withdraws:0,  Replacements:0
```

```
Peer Out of Memory Count for:
   Receiving Update Messages:0, Accepting Routes(NLRI):0
   Attributes:0, Outbound Routes(RIB-out):0 Outbound Routes Holder:0
```

Syntax: show ipv6 bgp neighbor [*ipv6-address*] **routes-summary**

TABLE 114 show ipv6 bgp neighbor routes-summary output descriptions

This field...	Displays...
IP Address	The IPv6 address of the neighbor
Routes Received	How many routes the device has received from the neighbor during the current BGP4+ session: • Accepted or Installed - Indicates how many of the received routes the device accepted and installed in the BGP4+ route table. • Filtered or Kept - Indicates how many routes were filtered out, but were nonetheless retained in memory for use by the soft reconfiguration feature. • Filtered - Indicates how many of the received routes were filtered out.
Routes Selected as BEST Routes	The number of routes that the device selected as the best routes to their destinations.
BEST Routes not Installed in IPv6 Forwarding Table	The number of routes received from the neighbor that are the best BGP4+ routes to their destinations, but were nonetheless not installed in the IPv6 route table because the device received better routes from other sources (such as OSPFv3, RIPng, or static IPv6 routes).
Unreachable Routes	The number of routes received from the neighbor that are unreachable because the device does not have a valid RIPng, OSPFv3, or static IPv6 route to the next hop.
History Routes	The number of routes that are down but are being retained for route flap dampening purposes.
NLRIs Received in Update Message	The number of routes received in Network Layer Reachability (NLRI) format in UPDATE messages: • Withdraws - The number of withdrawn routes the device has received. • Replacements - The number of replacement routes the device has received.
NLRIs Discarded due to	Indicates the number of times the device discarded an NLRI for the neighbor due to the following reasons: • Maximum Prefix Limit - The device's configured maximum prefix amount had been reached. • AS Loop - An AS loop occurred. An AS loop occurs when the BGP4+ AS-path attribute contains the local AS number. • Invalid Nexthop Address - The next hop value was not acceptable. • Duplicated Originator_ID - The originator ID was the same as the local router ID. • Cluster_ID - The cluster list contained the local cluster ID, or contained the local router ID (see above) if the cluster ID is not configured.

TABLE 114 show ipv6 bgp neighbor routes-summary output descriptions (Continued)

This field...	Displays...
Routes Advertised	The number of routes the device has advertised to this neighbor: • To be Sent - The number of routes the device has queued to send to this neighbor. • To be Withdrawn - The number of NLRIs for withdrawing routes the device has queued up to send to this neighbor in UPDATE messages.
NLRIs Sent in Update Message	The number of NLRIs for new routes the device has sent to this neighbor in UPDATE messages: • Withdraws - The number of routes the device has sent to the neighbor to withdraw. • Replacements - The number of routes the device has sent to the neighbor to replace routes the neighbor already has.
Peer Out of Memory Count for	Statistics for the times the device has run out of BGP4+ memory for the neighbor during the current BGP4+ session: • Receiving Update Messages - The number of times UPDATE messages were discarded because there was no memory for attribute entries. • Accepting Routes(NLRI) - The number of NLRIs discarded because there was no memory for NLRI entries. This count is not included in the Receiving Update Messages count. • Attributes - The number of times there was no memory for BGP4+ attribute entries. • Outbound Routes (RIB-out) - The number of times there was no memory to place a "best" route into the neighbor's route information base (Adj-RIB-Out) for routes to be advertised. • Outbound Routes Holder - For debugging purposes only.

Displaying BGP4+ peer group configuration information

You can display configuration information for all peer groups or a specified peer group configured on a device.

For example, to display configuration information for a peer group named peer1, enter the following command at any level of the CLI.

```
device# show ipv6 bgp peer-group peer_group1
1   BGP peer-group is pg1, Remote AS: 65002
    Description: device group 1
       NextHopSelf: yes
      Address family : IPV4 Unicast
      Address family : IPV4 Multicast
      Address family : IPV6 Unicast
    Members:
       IP Address: 10.169.102.2
       IP Address: 10.169.100.2
       IP Address: 10.169.101.2
       IP Address: 10.169.103.2
       IP Address: 10.169.104.2
       IP Address: 10.169.105.2
       IP Address: 10.169.106.2
       IP Address: 10.169.107.2
       IP Address: 10.169.108.2
       IP Address: 10.169.109.2
       IP Address: 10.169.110.2
       IP Address: 10.169.111.2
       IP Address: 10.169.112.2
```

Syntax: show ipv6 bgp peer-group [*peer-group-name*]

The display shows only parameters that have values different from their default settings.

Displaying BGP4+ summary

To view summary BGP4+ information for the device, enter the following command at any level of the CLI.

```
device# show ipv6 bgp summary
BGP4 Summary
 Router ID: 10.223.223.223   Local AS Number : 65001
Confederation Identifier : not configured
Confederation Peers:
Maximum Number of Paths Supported for Load Sharing : 1
Number of Neighbors Configured : 1
Number of Routes Installed : 2
Number of Routes Advertising to All Neighbors : 2
Number of Attribute Entries Installed : 1
Neighbor Address  AS#   State   Time     Rt:Accepted Filtered Sent   ToSend
2001:db8::110     65002 ESTAB  21h32m32s    0        0        2      0
```

Syntax: show ipv6 bgp summary

TABLE 115 show ipv6 bgp summary output descriptions

This field...	Displays...
Router ID	The device's router ID.
Local AS Number	The BGP4+ AS number in which the device resides.
Confederation Identifier	The autonomous system number of the confederation in which the device resides.
Confederation Peers	The numbers of the local autonomous systems contained in the confederation. This list matches the confederation peer list you configure on the device.
Maximum Number of Paths Supported for Load Sharing	The maximum number of route paths across which the device can balance traffic to the same destination. The feature is enabled by default but the default number of paths is 1. You can increase the number from 2 - 8 paths.
Number of Neighbors Configured	The number of BGP4+ neighbors configured on this device.
Number of Routes Installed	The number of BGP4+ routes in the device's BGP4+ route table.
Number of Routes Advertising to All Neighbors	The total of the RtSent and RtToSend columns for all neighbors.
Number of Attribute Entries Installed	The number of BGP4+ route-attribute entries in the route-attributes table.
Neighbor Address	The IPv6 addresses of this BGP4+ neighbors.
AS#	The autonomous system number.

TABLE 115 show ipv6 bgp summary output descriptions (Continued)

This field...	Displays...
State	The state of this neighbor session with each neighbor. The states are from this perspective of the session, not the neighbor's perspective. The state values can be one of the following for each: • IDLE - The BGP4+ process is waiting to be started. Usually, enabling BGP4+ or establishing a neighbor session starts the BGP4+ process. - A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • ADMND - The neighbor has been administratively shut down. - A minus sign (-) indicates that the session has gone down and the software is clearing or removing routes. • CONNECT - BGP4+ is waiting for the connection process for the TCP neighbor session to be completed. • ACTIVE - BGP4+ is waiting for a TCP connection from the neighbor. **NOTE** If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection. • OPEN SENT - BGP4+ is waiting for an Open message from the neighbor. • OPEN CONFIRM - BGP4+ has received an OPEN message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. If the receives a KEEPALIVE message from the neighbor, the state changes to Established. If the message is a NOTIFICATION, the state changes to Idle. • ESTABLISHED - BGP4+ is ready to exchange UPDATE packets with the neighbor. - If there is more BGP data in the TCP receiver queue, a plus sign (+) is also displayed. **NOTE** If you display information for the neighbor using the **show ipv6 bgp neighbor**<*ipv6-address*> command, the TCP receiver queue value will be greater than 0.
Time	The time that has passed since the state last changed.
Accepted	The number of routes received from the neighbor that this installed in the BGP4+ route table. Usually, this number is lower than the RoutesRcvd number. The difference indicates that this filtered out some of the routes received in the UPDATE messages.
Filtered	The routes or prefixes that have been filtered out. • If soft reconfiguration is enabled, this field shows how many routes were filtered out (not placed in the BGP4+ route table) but retained in memory. • If soft reconfiguration is not enabled, this field shows the number of BGP4+ routes that have been filtered out.
Sent	The number of BGP4+ routes that the has sent to the neighbor.

TABLE 115 show ipv6 bgp summary output descriptions (Continued)

This field...	Displays...
ToSend	The number of routes the has queued to send to this neighbor.

Configuring BGP4+ graceful restart

BGP4+ Graceful Restart (GR) can be configured for a global routing instance or for a specified Virtual Routing and Forwarding (VRF) instance.

The following sections describe how to enable the BGP4+ Graceful Restart feature.

NOTE
Graceful restart is not supported for multicast. Only IPv4 and IPv6 are supported.

BGP4+ Graceful Restart is fully supported by Brocade MLX series and Brocade NetIron XMR devices. The Brocade NetIron CER and Brocade NetIron CES devices only support helper mode.

BGP4+ Graceful Restart can be executed in both IPv4 and IPv6 address families. Depending on the remote neighbor address family, the command and its parameters will be taken from the IPv4 family or IPv6 family.

When the **graceful restart** command is enabled, the BGP graceful restart capability is negotiated with neighbors in the BGP OPEN message when the session is established. If the neighbor also advertises support for graceful restart, then graceful restart is activated for that neighbor session. If the neighbor does not advertise support for graceful restart, then graceful restart is not activated for that neighbor session even though it is enabled locally. If the neighbor has not sent graceful restart parameters, the restarting device will not wait for the neighbor to start route-calculation, but graceful restart will be enabled.

Configuring BGP4+ graceful restart for the global routing instance

Use the following command to enable the BGP4+ graceful restart feature globally on a device.

```
device(config)# router bgp
device(config-bgp)# graceful-restart
```

Syntax: [no] graceful-restart

Configuring timers for BGP4+ graceful restart (optional)

You can optionally configure the following timers to change their values from the default values:

- Restart Timer
- Stale Routes Timer
- Purge Timer

Configuring the restart timer for BGP4+ graceful restart

Use the following command to specify the maximum amount of time a device will maintain routes from and forward traffic to a restarting device.

```
device(config-bgp)# graceful-restart restart-timer 150
```

Syntax: [no] graceful-restart restart-timer *seconds*

The *seconds* variable sets the maximum restart wait time advertised to neighbors. The allowable range is 1 to 3600 seconds. The default value is 120 seconds.

Configuring BGP4+ graceful restart stale routes timer

Use the following command to specify the maximum amount of time a helper device will wait for an end-of-RIB message from a peer before deleting routes from that peer.

```
device(config-bgp)# graceful-restart stale-routes-time 120
```

Syntax: [no] graceful-restart stale-routes-time *seconds*

The *seconds* variable sets the maximum time before a helper device cleans up stale routes. The allowable range is 1 to 3600 seconds. The default value is 360 seconds.

Configuring BGP4+ graceful restart purge timer

Use the following command to specify the maximum amount of time a device will maintain stale routes in its routing table before purging them.

```
device(config-bgp)# graceful-restart purge-time 900
```

Syntax: [no] graceful-restart purge-time *seconds*

The *seconds* variable sets the maximum time before a restarting device cleans up stale routes. The allowable range is 1 to 3600 seconds. The default value is 600 seconds.

Displaying BGP4+ graceful restart neighbor information

To display BGP4+ graceful restart information for BGP4 and BGP4+ neighbors, enter the **show ip bgp neighbors** command.

```
device# show ip bgp neighbors
    Total number of BGP Neighbors: 6
1   IP Address: 10.50.50.10, AS: 20 (EBGP), RouterID: 10.10.10.20, VRF: default
    State: ESTABLISHED, Time: 0h0m18s, KeepAliveTime: 60, HoldTime: 180
       KeepAliveTimer Expire in 34 seconds, HoldTimer Expire in 163 seconds
       Minimum Route Advertisement Interval: 0 seconds
       RefreshCapability: Received
       GracefulRestartCapability: Received
           Restart Time 120 sec, Restart bit 0
           afi/safi 1/1, Forwarding bit 0
       GracefulRestartCapability: Sent
           Restart Time 120 sec, Restart bit 0
           afi/safi 1/1, Forwarding bit 1
    Messages:    Open    Update  KeepAlive Notification Refresh-Req

....
```

OSPFv3

- OSPFv3 feature support 548
- OSPFv3 overview 549
- Link-state advertisement types for OSPFv3 550
- Configuring OSPFv3 550
- Displaying OSPFv3 information 581
- OSPFv3 clear commands 612

OSPFv3 feature support

Lists OSPFv3 features supported on NetIron devices.

The following features are documented in the *Multi-Service IronWare Routing Configuration Guide*.

Features supported	Brocade NetIron XMR Series	Brocade NetIron MLX Series	Brocade NetIron CES 2000 Series BASE Package	Brocade NetIron CES 2000 Series ME_PREM Package	Brocade NetIron CES 2000 Series L3_PREM Package	Brocade NetIron CER 2000 Series Base Package	Brocade NetIron CER 2000 Series Advanced Services Package
OSPFv3	Yes	Yes	No	Yes	Yes	Yes	Yes
Route redistribution into OSPFv3	Yes	Yes	No	Yes	Yes	Yes	Yes
OSPFv3 route filtering	Yes	Yes	No	Yes	Yes	Yes	Yes
SPF Timers	Yes	Yes	No	Yes	Yes	Yes	Yes
OSPFv3 LSA Pacing	Yes	Yes	No	Yes	Yes	Yes	Yes
Modify exit overflow intervals	Yes	Yes	No	Yes	Yes	Yes	Yes
Modify External LSA Database Limit	Yes	Yes	No	Yes	Yes	Yes	Yes
OSPFv3 Event Logging	Yes	Yes	No	Yes	Yes	Yes	Yes
OSPFv3 Debugging	Yes	Yes	No	Yes	Yes	Yes	Yes
IPsec for OSPFv3	Yes	Yes	No	Yes	Yes	Yes	Yes

Features supported	Brocade NetIron XMR Series	Brocade NetIron MLX Series	Brocade NetIron CES 2000 Series BASE Package	Brocade NetIron CES 2000 Series ME_PREM Package	Brocade NetIron CES 2000 Series L3_PREM Package	Brocade NetIron CER 2000 Series Base Package	Brocade NetIron CER 2000 Series Advanced Services Package
Encryption code for passwords, authentication keys and community strings	Yes	Yes	No	Yes	Yes	Yes	Yes
OSPFv3 Graceful Restart	No	No	No	No	No	No	No
OSPFv3 Graceful Restart Helper	Yes	Yes	Yes	Yes	Yes	Yes	Yes
OSPFv3 NSR	Yes	Yes	No	No	No	No	No

OSPFv3 overview

Open Shortest Path First (OSPF) is a link-state routing protocol. OSPF uses link-state advertisements (LSAs) to update neighboring routers about its interfaces and information on those interfaces. The device floods LSAs to all neighboring routers to update them about the interfaces. Each router maintains an identical database that describes its area topology to help a router determine the shortest path between it and any neighboring router.

IPv6 supports OSPF Version 3 (OSPFv3), which functions similarly to OSPF Version 2 (OSPFv2), the current version that IPv4 supports, except for the following enhancements:

- Support for IPv6 addresses and prefixes.
- Ability to configure several IPv6 addresses on a device interface. (While OSPF V2 runs per IP subnet, OSPFv3 runs per link. In general, you can configure several IPv6 addresses on a router interface, but OSPFv3 forms one adjacency per interface only, using the interface associated link-local address as the source for OSPF protocol packets. On virtual links, OSPFv3 uses the global IP address as the source.OSPFv3 imports all or none of the address prefixes configured on a router interface. You cannot select the addresses to import.)
- Ability to run one instance of OSPF Version 2 and one instance of OSPFv3 concurrently on a link.
- Support for IPv6 link-state advertisements (LSAs).

This section describes the commands that are specific to OSPFv3.

NOTE
Although OSPF Version 2 and 3 function similarly to each other, Brocade has implemented the user interface for each version independently of the other. Therefore, any configuration of OSPFv2 features will not affect the configuration of OSPFv3 features and vice versa.

NOTE
You are required to configure a router ID when running only IPv6 routing protocols.

Link-state advertisement types for OSPFv3

OSPFv3 supports the following types of LSAs:

- Router LSAs (Type 1)
- Network LSAs (Type 2)
- Interarea-prefix LSAs for ABRs (Type 3)
- Interarea-router LSAs for ASBRs (Type 4)
- Autonomous system external LSAs (Type 5)
- Link LSAs (Type 8)
- Intra-area-prefix LSAs (Type 9)

For more information about these LSAs, see RFC 2740.

Configuring OSPFv3

To configure OSPFv3, you must perform the following steps.

- Enable OSPFv3 globally.
- Assign OSPF areas.
- Assign device interfaces to an OSPF area.

The following configuration tasks are optional:

- Configure a virtual link between an Area Border Router (ABR) without a physical connection to a backbone area and the device in the same area with a physical connection to the backbone area.
- Change the reference bandwidth for the cost on OSPFv3 interfaces.
- Configure the redistribution of routes into OSPFv3.
- Configure default route origination.
- Modify the shortest path first (SPF) timers.
- Modify the administrative distances for OSPFv3 routes.
- Configure the OSPFv3 LSA pacing interval.
- Modify how often the Brocade device checks on the elimination of the database overflow condition.
- Modify the external link state database limit.
- Modify the default values of OSPFv3 parameters for device interfaces.
- Disable or re-enable OSPFv3 event logging.
- Set all the OSPFv3 interfaces to the passive state.

Enabling OSPFv3

Before enabling the device to run OSPFv3, you must perform the following steps.

- Enable the forwarding of IPv6 traffic on the device using the **ipv6 unicast-routing** command.
- Enable IPv6 on each interface over which you plan to enable OSPFv3. You enable IPv6 on an interface by configuring an IPv6 address or explicitly enabling IPv6 on that interface.

By default, OSPFv3 is disabled. To enable OSPFv3 for a default Virtual Routing and Forwarding (VRF), you must enable it globally.

To enable OSPFv3 globally, enter the following command.

```
device(config)# ipv6 router ospf
device(config-ospf6-router)#
```

After you enter this command, the Brocade device enters the IPv6 OSPF configuration level, where you can access several commands that allow you to configure OSPFv3.

Enabling OSPFv3 in a VRF

To enable OSPFv3 for a default Virtual Routing and Forwarding (VRF), enter a command such as the following.

device(config-ospf6-router)# ipv6 router ospf vrf red

Syntax: [no] ipv6 router ospf vrf *vrf-name*

The *vrf-name* parameter specifies the name of the VRF in which OSPFv3 is being initiated.

Disabling OSPFv3 in a VRF

To disable OSPFv3 for a default Virtual Routing and Forwarding (VRF), enter a command such as the following.

device(config-ospf6-router)# no ipv6 router ospf vrf red

Syntax: [no] ipv6 router ospf vrf *vrf-name*

The *vrf-name* parameter specifies the name of the VRF in which OSPFv3 is being initiated.

If you disable OSPFv3, the device removes all the configuration information for the disabled protocol from the running-configuration file. Moreover, when you save the configuration to the startup-config file after disabling one of these protocols, all the configuration information for the disabled protocol is removed from the startup-config file.

When you disable OSPFv3, the following warning message is displayed on the console.

```
device(config-ospf6-router)# no ipv6 router ospf
ipv6 router ospf mode now disabled. All ospf config data will be lost when writing to
flash!
```

If you have disabled the protocol but have not yet saved the configuration to the startup-config file and reloaded the software, you can restore the configuration information by re-entering the command to enable the protocol (for example, **ipv6 router ospf**). If you have already saved the configuration to the startup-config file and reloaded the software, the configuration information is gone. If you are testing an OSPF configuration and are likely to disable and re-enable the protocol, you should make a backup copy of the startup-config file containing the protocol configuration information. This way, if you remove the configuration information by saving the configuration after disabling the protocol, you can restore the configuration by copying the backup copy of the startup-config file onto the flash memory.

NOTE
All the configuration examples below are applicable for OSPFv3 configuration mode in VRFs as well.

Assigning OSPFv3 areas

After OSPFv3 is enabled, you can assign OSPFv3 areas. You can assign an IPv4 address or a number as the area ID for each area. The area ID is representative of all IPv4 addresses (subnets) on a device interface. Each device interface can support one area.

An area can be normal, a stub, or a Not-So-Stubby Area (NSSA) :

- Normal - OSPFv3 devices within a normal area can send and receive External Link State Advertisements (LSAs).
- Stub - OSPFv3 devices within a stub area cannot send or receive External LSAs. In addition, OSPF devices in a stub area must use a default route to the area's Area Border Router (ABR) or Autonomous System Boundary Router (ASBR) to send traffic out of the area.
- NSSA - The ASBR of an NSSA can import external route information into the area.
    - ASBRs redistribute (import) external routes into the NSSA as type 7 LSAs. Type-7 External LSAs are a special type of LSA generated only by ASBRs within an NSSA, and are flooded to all the routers within only that NSSA.
    - ABRs translate type 7 LSAs into type 5 External LSAs, which can then be flooded throughout the AS. You can configure address ranges on the ABR of an NSSA so that the ABR converts multiple type-7 External LSAs received from the NSSA into a single type-5 External LSA.

When an NSSA contains more than one ABR, OSPFv3 elects one of the ABRs to perform the LSA translation for NSSA. OSPF elects the ABR with the highest router ID. If the elected ABR becomes unavailable, OSPFv3 automatically elects the ABR with the next highest router ID to take over translation of LSAs for the NSSA. The election process for NSSA ABRs is automatic.

For example, to set up OSPFv3 areas 2001:db8: :32, 2001:db8: :32, 2001:db8: :15, and 2001:db8:64, enter the following commands.

```
device(config-ospf6-router)# area 2001:db8::32
device(config-ospf6-router)# area 2001:db8::32
device(config-ospf6-router)# area 2001:db8::15
device(config-ospf6-router)# area 2001:db8::64
```

Syntax: [no] area {*number* | *ipv4-address*}

The *number* and *ipv4-address* parameters specify the area number, which can be a number or in IPv4 address format.

NOTE
You can assign only one area on a device interface.

Assigning a totally stubby area

By default, the device sends summary LSAs (type 3 LSAs) into stub areas. You can reduce the number of LSAs sent into a stub area by configuring the device to stop sending summary LSAs into the area. You can disable the summary LSAs when you are configuring the stub area or later after you have configured the area.

This feature disables origination of summary LSAs into a stub area, but the device still accepts summary LSAs from OSPF neighbors and floods them to other areas. The device can form adjacencies with other routers regardless of whether summarization is enabled or disabled for areas on each router.

When you disable the summary LSAs, the change takes effect immediately. If you apply the option to a previously configured area, the device flushes all of the summary LSAs it has generated (as an ABR) from the area.

NOTE
This feature applies only when the Brocade device is configured as an Area Border Router (ABR) for the area. To completely prevent summary LSAs from being sent to the area, disable the summary LSAs on each OSPF router that is an ABR for the area.

For example, to disable summary LSAs for stub area 40 and specify an additional metric of 99, enter the following command.

```
device(config-ospf6-router)# area 40 stub 99 no-summary
```

Syntax: [no] area {*number* | *ipv4-address*} **stub** *metric* [**no-summary**]

The *number* and *ipv4-address* parameters specify the area number, which can be a number or in IPv4 address format. If you specify a number, the number can be from 0 through 2,147,483,647.

The **stub** *metric* parameter specifies an additional cost for using a route to or from this area and can be from 1 through 16777215. There is no default. Normal areas do not use the cost parameter.

The **no-summary** parameter applies only to stub areas and disables summary LSAs from being sent into the area.

Assign a Not-So-Stubby Area (NSSA)

The OSPF Not So Stubby Area (NSSA) feature enables you to configure OSPF areas that provide the benefits of stub areas, but that also are capable of importing external route information. OSPF does not flood external routes from other areas into an NSSA, but does translate and flood route information from the NSSA into other areas such as the backbone.

NSSAs are especially useful when you want to summarize Type-5 External LSAs (external routes) before forwarding them into an OSPF area. The OSPF specification (RFC 2328) prohibits summarization of Type-5 LSAs and requires OSPF to flood Type-5 LSAs throughout a routing domain. When you configure an NSSA, you can specify an address range for aggregating the external routes that the NSSAs ABR exports into other areas.

Since the NSSA is partially "stubby" the ABR does not flood external LSAs from the backbone into the NSSA. To provide access to the rest of the Autonomous System (AS), the ABR generates a default Type-7 LSA into the NSSA.

Configuring an NSSA

Using the **area nssa** command, you can block the generation of type-3 and type-7 LSAs into an NSSA. This command also provides an option to configure the NSSA translator role.

Configuration examples

The following example creates an NSSA area with an area-id 100. If the router is an ABR then a type-3 summary LSA will be originated into the NSSA area and if the router is an ASBR then type-7 NSSA external LSA will be generated into NSSA area with a default external metric value of 10. The routers NSSA translator role will be set to candidate and it will participate in NSSA translation election.

```
device(config-ospf6-router)# area 100 nssa
```

The following example modifies the NSSA area 100 wherein type-7 NSSA external LSA will not be originated into NSSA area. But the type-3 summary LSAs will still be originated into NSSA area.

```
device(config-ospf6-router)# area 100 nssa no-redistribution
```

The following example modifies the NSSA area 100 wherein origination of type-3 summary LSAs (apart from type-3 default summary) will be blocked into NSSA area. The CLI works in incremental fashion and the origination of type-7 LSA will be continued to be blocked as 'no-redistribution' option was enabled in the previous command.

```
device(config-ospf6-router)#area 100 nssa no-summary
```

The following example modifies the NSSA area 100 wherein origination of the self-router acts as NSSA translator. The generation of type-3 & type-7 LSA will still be blocked into NSSA area.

```
device(config-ospf6-router)#area 100 nssa translator-always
```

The following example modifies the NSSA area 100 wherein origination of type-3 summary will be allowed, but origination of type-7 LSA will still be blocked. Also the self-router will still act as NSSA translator-always.

```
device(config-ospf6-router)#no area 100 nssa no-summary
```

Although the NSSA configuration can be done in an incremental fashion during show-run, all the configuration options will be displayed in just one line. For example, the output of the **show run** would be:

```
device(config-ospf6-router)#area 100 nssa no-redistribution translator-always
```

The following example deletes the NSSA area 100.

```
device(config-ospf6-router)#no area 100
```

Syntax: [no] area *area-id* **nssa** [[*stub-metric*] [**default-information-originate** [**metric** *metric-value* | **metric-type** *type-value*]] [**no-summary**] [**no-redistribution**] [**translator-always**] [**translator-interval** *stability-interval*]]

The *area-id* parameter specifies the area number, which can be a number or in IP address format. If you specify a number, the number can be from 0 to 2,147,483,647.

The **nssa** *stub-metric* parameter configures an area as a not-so-stubby-area (NSSA). The *stub-metric* will be the metric used for generating default LSA in a NSSA. The range of the value is 1 to 1048575. The default value is 10.

The **default-information-originate** parameter generates a default route into an NSSA. If no-summary option is enabled then a type-3 default LSA will be generated into NSSA else a type-7 LSA will be generated into NSSA. By default the **default-information-origiante** parameter is not set.

The **metric** *metric-value* parameter specifies the cost of the default LSA originated into the NSSA area. The range is 1 to 1048575. There is no default

The **metric-type** *type-value* parameter specifies the type of the default external LSA originated into the NSSA area. It can be either type-1 or type-2. The default is type-1.

The **no-summary** parameter prevents an NSSA ABR from generating a type-3 summary into an NSSA. By default the summary LSA is originated into NSSA.

The **no-redistribution** parameter prevents an NSSA ABR from generating external (type-7) LSA into an NSSA area. This is used in the case where an ASBR should generate type-5 LSA into normal areas and should not generate type-7 LSA into NSSA area. By default, redistribution is enabled in a NSSA.

The **translator-always** parameter configures the translator-role. When configured on an ABR, this causes the router to unconditionally assume the role of an NSSA translator. By default, translator-always is not set, the translator role by default is candidate.

The **translator-interval** *stability-interval* parameter configures the time interval for which an elected NSSA translator continues to perform its duties even after its NSSA translator role has been disposed by another router. By default the stability-interval is 40 seconds and its range will be 10 to 60 seconds.

Disabling the router to perform translations for NSSA LSAs

The **nssa-translator** command allows you to disable the router from performing translations for NSSA LSAs. When this command is used, type 7 NSSA external LSAs are not translated into type 5 external LSAs. This command is useful when the router is an area border router with many NSSA areas, and does not need to export the NSSA external routes into the backbone.

```
device(config)# router ospf
device(config-ospf6-router)# no nssa-translator
```

Syntax: [no] nssa-translator

Configuring an address range for the NSSA

If you want the ABR that connects the NSSA to other areas to summarize the routes in the NSSA before translating them into Type-5 LSAs and flooding them into the other areas, configure an address range. The ABR creates an aggregate value based on the address range. The aggregate value becomes the address that the ABR advertises instead of advertising the individual addresses represented by the aggregate. You can configure up to 32 ranges in an OSPF area.

To configure an address range in NSSA 10.1.1.1, enter the following commands. This example assumes that you have already configured NSSA 10.1.1.1.

```
device(config)# router ospf
device(config-ospf6-router)# area 10.1.1.1 range 11:11::0/32  //ipv6 address range
device(config-ospf6-router)# write memory
```

Syntax: [no] area {*num* | *ip-addr*} {**range** *ipv6-addr/ipv6-subnet-mask*} [**advertise** | **not-advertise**]

The *num* and *ip-addr* parameters specify the area number, which can be in IP address format. If you specify a number, the number can be from 0 - 2,147,483,647.

The **range** *ipv6-addr* parameter specifies the IP address portion of the range. The software compares the address with the significant bits in the mask. All network addresses that match this comparison are summarized in a single route advertised by the router.

The *ipv6-subnet-mask* parameter specifies the portions of the IPv6 address that a route must contain to be summarized in the summary route. In the example above, all networks that begin with 11:11::0 are summarized into a single route.

The **advertise** and **not-advertise** parameters specify whether you want the device to send type 3 LSAs for the specified range in this area. The default is **advertise**.

Assigning an area cost for OSPFv3 (optional parameter)

You can assign a cost for an area, but it is not required. To consolidate and summarize routes at an area boundary, use the **area range cost** command in router configuration mode.

If the **cost** parameter is specified, it will be used (overriding the computed cost) to generate the summary LSA. If the **cost** parameter is not specified, then the existing range metric computation max or min cost of routes falling under this range will be used to generate summary LSA.

NOTE
The area should be already configured before using this command.

Creates an area range entry with prefix 2001:db8::1/64 with the area-id 10.

```
device(config)# ipv6 router ospf
device(config-ospf6-router)# area 10 range 2001:db8::1/64
```

Modifies the address range status to DoNotAdvertise. Neither the individual intra-area routes falling under range nor the ranged prefix is advertised as summary LSA.

```
device(config)# ipv6 router ospf
device(config-ospf6-router)# area 10 range 2001:db8::1/64 not-advertise
```

Modifies the address range status to advertise and a Type 3 summary link-state advertisement (LSA) can be generated for this address range.

```
device(config)# ipv6 router ospf
device(config-ospf6-router)# area 10 range 2001:db8::1/64 advertise
```

Modifies the address range status to advertise and assign cost for this area range to 10.

```
device(config)# ipv6 router ospf
device(config-ospf6-router)# area 10 range 2001:db8::1/64 advertise cost 10
```

Modifies the address range status to not-advertise and cost from 10 to 5.

```
device(config)# ipv6 router ospf
device(config-ospf6-router)# area 10 range 2001:db8::1/64 not-advertise cost 5
```

Removes the cost from the area range. The area range will be advertised with computed cost which is the max/min (based on RFC 1583 compatibility) of all individual intra-area routes falling under this range.

```
device(config)# ipv6 router ospf
device(config-ospf6-router)# no area 10 range 2001:db8::1/64 cost 5
```

Removes the area range.

```
device(config)# ipv6 router ospf
device(config-ospf6-router)# no area 10 range 2001:db8::1/64
```

NOTE
This command does not work in incremental fashion. So both the optional parameters have to be configured each time. Otherwise it will take the default value.

Syntax: [no] area {**num** | **ipv6-addr**} **range** *ipv6-addr/ipv6-subnet-mask* [**advertise** | **not-advertise**] [**cost** *cost-value*]

The *num* and *ipv6-addr* parameters specify the area number, which can be in IP address format.

The **range** *ipv6-addr* parameter specifies the IP address portion of the range. The software compares the address with the significant bits in the mask. All network addresses that match this comparison are summarized in a single route advertised by the router.

The *ipv6-subnet-mask* parameter specifies the portions of the IPv6 address that a route must contain to be summarized in the summary route. In the example above, all networks that begin with 193.45 are summarized into a single route.

The **advertise** parameter sets the address range status to advertise and generates a Type 3 summary link-state advertisement (LSA). If at least a single route falls under the range, a ranged LSA will be advertised.

The not-advertise parameter sets the address range status to DoNotAdvertise. Neither the individual intra-area routes falling under range nor the ranged prefix is advertised as summary LSA.

The **cost** *cost-value* parameter specifies the cost-value to be used while generating type-3 summary LSA. If the cost value is configured, then configured cost is used while generating the summary LSA. If the cost value is not configured, then computed range cost will be used. The cost-value ranges from 1 to 16777215.

To disable this function, use the **no** form of this command.

Assigning interfaces to an area

After you define OSPFv3 areas, you must assign device interfaces to the areas. All device interfaces must be assigned to one of the defined areas on an OSPF router. When an interface is assigned to an area, all corresponding subnets on that interface are automatically included in the assignment.

For example, to assign Ethernet interface 3/1 to area 10.5.0.0, enter the following commands.

```
device(config)# interface Ethernet 3/1
device(config-if-e100-3/1)# ipv6 ospf area 10.5.0.0
```

Syntax: [no] ipv6 ospf area {*number* | *ipv4-address*}

The *number* and *ipv4-address* parameters specify the area number, which can be a number or in IPv4 address format. If you specify a number, the number can be from 0 through 2,147,483,647.

To remove the interface from the specified area, use the **no** form of this command.

Specifying a network type

You can specify a point-to-point or broadcast network type for any OSPF interface of the following types: Ethernet, or VE interface. To specify the network type for an OSPF interface, use the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 ospf network broadcast
```

Syntax: [no] ipv6 ospf network {**point-to-point** | **broadcast**}

The **point-to-point** parameter specifies that the OSPF interface will support point-to-point networking. This is the default setting for tunnel interfaces.

The **broadcast** parameter specifies that the OSPF interface will support broadcast networking. This is the default setting for Ethernet and VE interfaces.

The **no** form of the command disables the command configuration.

Configuring virtual links

All ABRs must have either a direct or indirect link to an OSPF backbone area (0.0.0.0 or 0). If an ABR does not have a physical link to a backbone area, you can configure a virtual link from the ABR to another router within the same area that has a physical connection to the backbone area.

The path for a virtual link is through an area shared by the neighbor ABR (router with a physical backbone connection) and the ABR requiring a logical connection to the backbone.

Two parameters must be defined for all virtual links -- transit area ID and neighbor router:

- The transit area ID represents the shared area of the two ABRs and serves as the connection point between the two routers. This number should match the area ID value.
- The neighbor router is the router ID (IPv4 address) of the router that is physically connected to the backbone when assigned from the router interface requiring a logical connection. The neighbor router is the router ID (IPv4 address) of the router requiring a logical connection to the backbone when assigned from the router interface with the physical connection.

NOTE
By default, the router ID is the IPv4 address configured on the lowest-numbered loopback interface. If the device does not have a loopback interface, the default router ID is the lowest-numbered IPv4 address configured on the device.

When you establish an area virtual link, you must configure it on both ends of the virtual link. For example, imagine that ABR1 in areas 1 and 2 is cut off from the backbone area (area 0). To provide backbone access to ABR1, you can add a virtual link between ABR1 and ABR2 in area 1 using area 1 as a transit area. To configure the virtual link, you define the link on the router that is at each end of the link. No configuration for the virtual link is required on the routers in the transit area.

To define the virtual link on ABR1, enter the following command on ABR1.

```
device(config-ospf6-router)# area 1 virtual-link 10.157.22.1
```

To define the virtual link on ABR2, enter the following command on ABR2.

```
device(config-ospf6-router)# area 1 virtual-link 10.0.0.1
```

Syntax: [no] area {*number* | *ipv4-address*} **virtual-link** *router-id*

The *number* and *ipv4-address* parameters specify the transit area ID, area number, which can be a number, or in IPv4 address format. If you specify a number, the number can be from 0 through 2,147,483,647.

The *router-id* parameter specifies the router ID of the OSPF router at the remote end of the virtual link. To display the router ID on a router, enter the **show ip** command.

Assigning a virtual link source address

When routers at both ends of a virtual link communicate with one another, the source address included in the packets must be a global IPv6 address. The Multi-Service IronWare software automatically selects a global IPv6 address for each transit area and advertises this address into the transit area of the Intra-area-prefix LSA. The automatically selected global IPv6 address for a transit area is the first global IPv6 address of any loopback interface in the transit area. If no global IPv6 address is available on a loopback interface in the area, then the first global IPv6 address of the lowest-numbered interface in the UP state (belonging to the transit area) will be assigned. If no global IPv6 address is configured on any of the OSPF interfaces in the transit area, then the virtual links in the transit area will not operate. The automatically selected IPv6 global address is updated whenever the previously selected IPv6 address of the interface changes, is removed, or if the interface goes down.

NOTE
The existing selected virtual link address will not change because the global IPv6 address is now available on a loopback interface or a lower-numbered interface in the transit area. To force the global IPv6 address for the virtual link to be the global IPv6 address of a newly configured loopback,or a

lower-numbered interface in the area, you will have to either disable the existing selected interface or remove the currently selected global IPv6 address from the interface.

Modifying virtual link parameters

You can modify the following virtual link parameters:

- Dead-interval: The number of seconds that a neighbor router waits for a hello packet from the device before declaring the router is down. The range is from 1 through 65535 seconds. The default is 40 seconds.
- Hello-interval: The length of time between the transmission of hello packets. The range is from 1 through 65535 seconds. The default is 10 seconds.
- Retransmit-interval: The interval between the retransmission of link state advertisements to router adjacencies for this interface. The range is from 0 through 3600 seconds. The default is 5 seconds.
- Transmit-delay: The period of time it takes to transmit Link State Update packets on the interface. The range is from 0 through 3600 seconds. The default is 1 second.

NOTE
The values of the **dead-interval** and **hello-interval** parameters must be the same at both ends of a virtual link. Therefore, if you modify the values of these parameters at one end of a virtual link, you must make the same modifications on the other end of the link.The values of the other virtual link parameters do not require synchronization.

For example, to change the **dead-interval** parameter to 60 seconds on the virtual links defined on ABR1 and ABR2, enter the following command on ABR1.

```
device(config-ospf6-router)# area 1 virtual-link 10.157.22.1
dead-interval 60
```

Enter the following command on ABR2.

```
device(config-ospf6-router)# area 1 virtual-link 10.0.0.1 dead-interval 60
```

Syntax: [no] area {*number* | *ipv4-address*} **virtual-link** *router-id* [**dead-interval** *seconds* | **hello-interval** *seconds* | **retransmit-interval** *seconds* | **transmit-delay** *seconds*]

The **area** *number* and *ipv4-address* parameters specify the transit area ID.

The *router-id* parameter specifies the router ID of the OSPF router at the remote end of the virtual link. To display the router ID on a device, enter the **show ip** command.

The **dead-interval**, **hello-interval**, **retransmit-interval**, and **transmit-delay** parameters are described earlier in this section.

Changing the reference bandwidth for the cost on OSPFv3 interfaces

Each interface on which OSPFv3 is enabled has a cost associated with it. The device advertises its interfaces and their costs to OSPFv3 neighbors. For example, if an interface has an OSPF cost of 10, the device advertises the interface with a cost of 10 to other OSPF routers.

By default, OSPF cost of an interface is based on the port speed of the interface. The software uses the following formula to calculate the cost.

Cost = reference-bandwidth/interface-speed

By default, the reference bandwidth is 100 Mbps. If the resulting cost is less than 1, the software rounds the cost up to 1. The default reference bandwidth results in the following costs:

- 10 Mbps port cost = 100/10 = 10
- 100 Mbps port cost = 100/100 = 1
- 1000 Mbps port cost = 100/1000 = 0.10, which is rounded up to 1
- 155 Mbps port cost = 100/155 = 0.65, which is rounded up to 1
- 622 Mbps port cost = 100/622 = 0.16, which is rounded up to 1
- 2488 Mbps port cost = 100/2488 = 0.04, which is rounded up to 1

The interfaces that consist of more than one physical port is calculated as follows:

- LAG group- The combined bandwidth of all the ports.
- Virtual (Ethernet) interface - The combined bandwidth of all the ports in the port-based VLAN that contains the virtual interface.

You can change the default reference bandwidth from 100 Mbps to a value from 1 through 4294967.

If a change to the reference bandwidth results in a cost change to an interface, the Brocade device sends a link-state update to update the costs of interfaces advertised by the Brocade device.

NOTE
If you specify a cost for an interface, your specified cost overrides the cost that the software calculates.

Some interface types are not affected by the reference bandwidth and always have the same cost regardless of the reference bandwidth in use:

- The cost of a loopback interface is always 1.
- The cost of a virtual link is calculated using the Shortest Path First (SPF) algorithm and is not affected by the auto-cost feature.
- The bandwidth for tunnel interfaces is 9 Kbps and is subject to the auto-cost feature.

For example, to change the reference bandwidth to 500, enter the following command.

```
device(config-ospf6-router)# auto-cost reference-bandwidth 500
```

The reference bandwidth specified in this example results in the following costs:

- 10 Mbps port cost = 500/10 = 50
- 100 Mbps port cost = 500/100 = 5
- 1000 Mbps port cost = 500/1000 = 0.5, which is rounded up to 1
- 155 Mbps port cost = 500/155 = 3.23, which is rounded up to 4
- 622 Mbps port cost = 500/622 = 0.80, which is rounded up to 1
- 2488 Mbps port cost = 500/2488 = 0.20, which is rounded up to 1

The costs for 10 Mbps, 100 Mbps, and 155 Mbps ports change as a result of the changed reference bandwidth. Costs for higher-speed interfaces remain the same.

Syntax: [no] auto-cost reference-bandwidth *number*

The *number* parameter specifies the reference bandwidth in the range from 1 through 4294967. The default is 100.

To restore the reference bandwidth to its default value and thus restore the default costs of the interfaces to their default values, enter the **no** form of this command.

Redistributing routes into OSPFv3

In addition to specifying which routes are redistributed into OSPFv3, you can configure the following aspects related to route redistribution:

- Default metric.
- Metric type.
- Advertisement of an external aggregate route.

Configuring route redistribution into OSPFv3

You can configure the device to redistribute routes from the following sources into OSPFv3:

- IPv6 static routes
- Directly connected IPv6 networks
- BGP4+
- IPv6 IS-IS
- RIPng

You can redistribute routes in the following ways:

- By route types, for example, the Brocade device redistributes all IPv6 static and RIPng routes.
- By using a route map to filter which routes to redistribute, for example, the device redistributes specified IPv6 static and RIPng routes only.

For example, to configure the redistribution of all IPv6 static, RIPng, and IPv6 IS-IS level-1 and level-2 routes, enter the following commands.

```
device(config-ospf6-router)# redistribute static
device(config-ospf6-router)# redistribute rip
device(config-ospf6-router)# redistribute isis level-1-2
```

Syntax: [no] redistribute {**bgp** | **connected** | **isis** [**level-1** | **level-1-2** | **level-2**] | **rip** | **static** [**metric** *number* | **metric-type** *type*]}

The **bgp**, **connected**, **rip**, and **static** keywords specify the route source.

The level-1, level-1-2, and level-2 keywords (for IPv6 IS-IS only) allow you to specify that the device redistributes level-1 routes only, level-2 routes only, or both level-1 and level-2 routes.

The **metric** *number* parameter specifies the metric used for the redistributed route. If a value is not specified for this option, and the value for the **default-metric** command is set to 0, its default metric, then routes redistributed from the various routing protocols will have the metric value of the protocol from which they are redistributed.

The **metric-type** *type* parameter specifies an OSPF metric type for the redistributed route. You can specify external type 1 or external type 2. If a value is not specified for this option, the device uses the value specified by the **metric-type** command.

For example, to configure a route map and use it for redistribution of routes into OSPFv3, enter commands such as the following.

```
device(config)# ipv6 route 2001:db8:1::/32 2001:db8:343e::23
device(config)# ipv6 route 2001:db8:2::/32 2001:db8:343e::23
device(config)# ipv6 route 2001:db8:3::/32 2001:db8:343e::23 metric 5
device(config)# route-map abc permit 1
device(config-routemap abc)# match metric 5
device(config-routemap abc)# set metric 8
device(config-routemap abc)# ipv6 router ospf
device(config-ospf6-router)# redistribute static route-map abc
```

The commands in this example configure some static IPv6 routes and a route map, and use the route map for redistributing the static IPv6 routes into OSPFv3.

The **ipv6 route** commands configure the static IPv6 routes.

The **route-map** command begins configuration of a route map called "abc". The number indicates the route map entry (called the "instance") you are configuring. A route map can contain multiple entries. The software compares packets to the route map entries in ascending numerical order and stops the comparison once a match is found.

NOTE
The default action rule for route-map is to deny all routes that are not explicitly permitted.

The **match** command in the route map matches on routes that have 5 for their metric value (cost). The **set** command changes the metric in routes that match the route map to 8.

The **redistribute** command configures the redistribution of static IPv6 routes into OSPFv3, and uses route map "abc" to control the routes that are redistributed. In this example, the route map allows a static IPv6 route to be redistributed into OSPF only if the route has a metric of 5, and changes the metric to 8 before placing the route into the OSPF route redistribution table.

Syntax: [no] redistribute {**bgp** | **connected** | **isis** | **rip** | **static** [**route-map** *map-name*]}

The **bgp**, **connected**, **isis**, **ip**, and **static** keywords specify the route source.

The **route-map** *map-name* parameter specifies the route map name. The following match parameters are valid for OSPFv3 redistribution:

- **match ipv6 address** | **next-hop***acl-number*
- **match metric** *number*
- **match tag** *tag-value*

The following set parameters are valid for OSPFv3 redistribution:

- **set ipv6 next-hop** *ipv6 address*
- **set metric** [+ | -] *number* | **none**
- **set metric-type type-1** | **type-2**
- **set tag** *tag-value*

NOTE
You must configure the route map before you configure a redistribution filter that uses the route map.

NOTE
When you use a route map for route redistribution, the software disregards the permit or deny action of the route map.

NOTE
For an external route that is redistributed into OSPFv3 through a route map, the metric value of the route remains the same unless the metric is set by a **set metric** command inside the route map or the **default-metric** command. For a route redistributed without using a route map, the metric is set by the metric parameter if set or the **default-metric** command if the metric parameter is not set.

Modifying default metric for routes redistributed into OSPF Version 3

The default metric is a global parameter that specifies the cost applied by default to routes redistributed into OSPFv3. The default value is 0.

If the **metric** parameter for the **redistribute** command is not set and the **default-metric** command is not set, the metric is set to 1, its default value, then routes redistributed from the various routing protocols will have the metric value of the protocol from which they are redistributed.

For information about the redistribute command, refer to Configuring route redistribution into OSPFv3 on page 561.

NOTE
You also can define the cost on individual interfaces. The interface cost overrides the default cost.

To assign a default metric of 4 to all routes imported into OSPFv3, enter the following command.

```
device(config-ospf6-router)# default-metric 4
```

Syntax: [no] default-metric *number*

You can specify a value from 0 - 65535. The default is 0.

To restore the default metric to the default value, use the **no** form of this command.

Modifying metric type for routes redistributed into OSPFv3

The device uses the **metric-type** parameter by default for all routes redistributed into OSPFv3 unless you specify a different metric type for individual routes using the **redistribute** command.

A type 1 route specifies a small metric (two bytes), while a type 2 route specifies a big metric (three bytes). The default value is type 2.

To modify the default value of type 2 to type 1, enter the following command.

```
device(config-ospf6-router)# metric-type type1
```

Syntax: [no] metric-type {**type1** | **type2**}

To restore the metric type to the default value, use the **no** form of this command.

Configuring external route summarization

When the Brocade device is an OSPF Autonomous System Boundary Router (ASBR), you can configure it to advertise one external route as an aggregate for all redistributed routes that are covered by a specified IPv6 address range.

When you configure an address range, the range takes effect immediately. All the imported routes are summarized according to the configured address range. Imported routes that have already been advertised and that fall within the range are flushed out of the AS and a single route corresponding to the range is advertised.

If a route that falls within a configured address range is imported by the device, no action is taken if the device has already advertised the aggregate route; otherwise, the device advertises the aggregate route. If an imported route that falls within a configured address range is removed by the device, no action is taken if there are other imported routes that fall with in the same address range; otherwise the aggregate route is flushed.

You can configure up to 32 address ranges. The device sets the forwarding address of the aggregate route to zero and sets the tag to zero.

If you delete an address range, the advertised aggregate route is flushed and all imported routes that fall within the range are advertised individually.

If an external link state database overflow (LSDB) condition occurs, all aggregate routes are flushed out of the AS, along with other external routes. When the device exits the external LSDB overflow condition, all the imported routes are summarized according to the configured address ranges.

NOTE
If you use redistribution filters in addition to address ranges, the Brocade device applies the redistribution filters to routes first, then applies them to the address ranges.

NOTE
If you disable redistribution, all the aggregate routes are flushed, along with other imported routes.

NOTE
This option affects only imported, type 5 external routes. A single type 5 LSA is generated and flooded throughout the AS for multiple external routes.

To configure the summary address 2001:db8::/24 for routes redistributed into OSPFv3, enter the following command.

```
device(config-ospf6-router)# summary-address 2001:db8::/24
```

In this example, the summary prefix 2001:db8::/24 includes addresses 2001:db8::/1 through 2001:db8::/24. Only the address FEC0::/24 is advertised in an external link-state advertisement.

Syntax: summary-address { *ipv6-prefix*|*prefix-length*}

You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

Filtering OSPFv3 routes

You can filter the routes to be placed in the OSPFv3 route table by configuring distribution lists. OSPFv3 distribution lists can be applied globally or to an interface.

The functionality of OSPFv3 distribution lists is similar to that of OSPFv2 distribution lists. However, unlike OSPFv2 distribution lists, which filter routes based on criteria specified in an Access Control List (ACL), OSPFv3 distribution lists can filter routes using information specified in an IPv6 prefix list or a route map.

Configuration examples

The following sections show examples of filtering OSPFv3 routes using prefix lists globally and for a specific interface, as well as filtering OSPFv3 routes using a route map.

You can configure the device to use all three types of filtering. When you do this, filtering using route maps has higher priority over filtering using global prefix lists. Filtering using prefix lists for a specific interface has lower priority than the other two filtering methods.

The examples in this section assume the following routes are in the OSPFv3 route table.

```
device# show ipv6 ospf route
  Current Route count: 5
    Intra: 3 Inter: 0 External: 2 (Type1 0/Type2 2)
    Equal-cost multi-path: 0
    Destination                   Options   Area             Cost Type2 Cost
    Next Hop Router               Outgoing Interface
*IA 2001:db8:1::/64               --------- 10.0.0.1             0 0
    ::                            ve 10
*E2 2001:db8:2::/64               --------- 0.0.0.0              10 0
    fe80::2e0:52ff:fe00:10        ve 10
*IA 2001:db8:3::/64               V6E---R-- 0.0.0.0              11 0
    fe80::2e0:52ff:fe00:10        ve 10
*IA 2001:db8:4::/64               --------- 0.0.0.0              10 0
    ::                            ve 11
*E2 2001:db8:5::/64               --------- 0.0.0.0              10 0
    fe80::2e0:52ff:fe00:10        ve 10
```

Configuring an OSPFv3 distribution list using an IPv6 prefix list as input

The following example illustrates how to use an IPv6 prefix list to filter OSPFv3 routes.

To specify an IPv6 prefix list called filterOspfRoutes that denies route 2001:db8:2::/64, enter the following commands.

```
device(config)# ipv6 prefix-list filterOspfRoutes seq 5 deny 2001:db8:2::/64
device(config)# ipv6 prefix-list filterOspfRoutes seq 7 permit ::/0 ge 1 le 128
```

Syntax: ipv6 prefix-list *name* [**seq** *seq-value*] [**description** *string*] {**deny** | **permit**} *ipv6-addr/mask-bits* [**ge** *ge-value*] [**le** *le-value*]

To configure a distribution list that applies the filterOspfRoutes prefix list globally.

```
device(config)# ipv6 router ospf
device(config-ospf6-router)# distribute-list prefix-list filterOspfRoutes in
```

Syntax: [no] distribute-list prefix-list *name* **in** [**ethernet** *slot/port* | **ve** *num* | **loopback** *num*]

After this distribution list is configured, route 2001:db8:2::/64 would be omitted from the OSPFv3 route table.

```
device# show ipv6 ospf route
  Current Route count: 4
    Intra: 3 Inter: 0 External: 1 (Type1 0/Type2 1)
    Equal-cost multi-path: 0
    Destination                   Options   Area             Cost Type2 Cost
    Next Hop Router               Outgoing Interface
*IA 2001:db8:1::/64               --------- 10.0.0.1              0 0
    ::                            ve 10
*IA 2001:db8:3::/64               V6E---R-- 0.0.0.0              11 0
    fe80::2e0:52ff:fe00:10        ve 10
*IA 2001:db8:4::/64               --------- 0.0.0.0              10 0
    ::                            ve 11
*E2 2001:db8:5::/64               --------- 0.0.0.0              10 0
    fe80::2e0:52ff:fe00:10        ve 10
```

The following commands specify an IPv6 prefix list called filterOspfRoutesVe that denies route 2001:db8:3::/64.

```
device(config)# ipv6 prefix-list filterOspfRoutesVe seq 5 deny 2001:db8:3::/64
device(config)# ipv6 prefix-list filterOspfRoutesVe seq 10 permit ::/0 ge 1 le 128
```

The following commands configure a distribution list that applies the filterOspfRoutesVe prefix list to routes pointing to virtual interface 10.

```
device(config)# ipv6 router ospf
device(config-ospf6-router)# distribute-list prefix-list filterOspfRoutesVe in ve 10
```

After this distribution list is configured, route 2001:db8:3::/64, pointing to virtual interface 10, would be omitted from the OSPFv3 route table.

```
device# show ipv6 ospf route
  Current Route count: 4
    Intra: 3 Inter: 0 External: 1 (Type1 0/Type2 1)
    Equal-cost multi-path: 0
    Destination                    Options   Area             Cost Type2 Cost
    Next Hop Router                Outgoing Interface
*IA 2001:db8:1::/64                --------- 10.0.0.1          0 0
    ::                             ve 10
*E2 2001:db8:2::/64                --------- 0.0.0.0           10 0
    fe80::2e0:52ff:fe00:10         ve 10
*IA 2001:db8:4::/64                --------- 0.0.0.0           10 0
    ::                             ve 11
*E2 2001:db8:5::/64                --------- 0.0.0.0           10 0
    fe80::2e0:52ff:fe00:10         ve 10
```

Configuring an OSPFv3 distribution list using a route map as input

The following commands configure a route map that matches internal routes.

```
device(config)# route-map allowInternalRoutes permit 10
device(config-routemap allowInternalRoutes)# match route-type internal
```

The following commands configure a distribution list that applies the allowInternalRoutes route map globally to OSPFv3 routes.

```
device(config)# ipv6 router ospf
device(config-ospf6-router)# distribute-list route-map allowinternalroutes in
```

Syntax: [no] distribute-list route-map *name* **in**

After this distribution list is configured, the internal routes would be included, and the external routes would be omitted from the OSPFv3 route table.

```
device# show ipv6 ospf route
  Current Route count: 3
    Intra: 3 Inter: 0 External: 0 (Type1 0/Type2 0)
    Equal-cost multi-path: 0
    Destination                    Options   Area             Cost Type2 Cost
    Next Hop Router                Outgoing Interface
*IA 2001:db8:3001::/64            --------- 10.0.0.1             0 0
    ::                             ve 10
*IA 2001:db8:3015::/64             V6E---R-- 0.0.0.0           11 0
    fe80::2e0:52ff:fe00:10         ve 10
*IA 2001:db8:3020::/64             --------- 0.0.0.0           10 0
    ::                             ve 11
```

Configuring default route origination

When the Brocade device is an OSPFv3 Autonomous System Boundary Router (ASBR), you can configure it to automatically generate a default external route into an OSPFv3 routing domain. This feature is called "default route origination" or "default information origination."

By default, the Brocade device does not advertise the default route into the OSPFv3 domain. If you want the device to advertise the OSPF default route, you must explicitly enable default route origination.

When you enable OSPFv3 default route origination, the device advertises a type 5 default route that is flooded throughout the AS (except stub areas).

The device advertises the default route into OSPF even if OSPF route redistribution is not enabled, and even if the default route is learned through an IBGP neighbor. The router will not, however, originate default if the active default route is learned from an OSPF router in the same domain.

NOTE
The Brocade device does not advertise the OSPFv3 default route, regardless of other configuration parameters, unless you explicitly enable default route origination.

If default route origination is enabled and you disable it, the default route originated by the device is flushed. Default routes generated by other OSPFv3 routers are not affected. If you re-enable the feature, the feature takes effect immediately and thus does not require you to reload the software.

For example, to create and advertise a default route with a metric of 2 and as a type 1 external route, enter the following command.

```
device(config-ospf6-router)# default-information-originate always metric 2 metric-type type1
```

Syntax: [no] default-information-originate [**always**] [**metric** *value*] [**metric-type** *type*]

The **always** keyword originates a default route regardless of whether the device has learned a default route. This option is disabled by default.

The **metric** *value* parameter specifies a metric for the default route. If this option is not used, the value of the **default-metric** command is used for the route.

The **metric-type** *type* parameter specifies the external link type associated with the default route advertised into the OSPF routing domain. The *type* can be one of the following:

- 1 - Type 1 external route
- 2 - Type 2 external route

If you do not use this option, the default redistribution metric type is used for the route type.

NOTE
If you specify a metric and metric type, the values are used even if you do not use the always option.

To disable default route origination, enter the **no** form of the command.

Modifying Shortest Path First timers

The Brocade device uses the following timers when calculating the shortest path for OSPFv3 routes:

- SPF delay - When the Brocade device receives a topology change, the software waits before it starts a Shortest Path First (SPF) calculation. By default, the software waits 5 seconds. You can configure the SPF delay to a value from 0 through 65535 seconds. If you set the SPF delay to 0 seconds, the software immediately begins the SPF calculation after receiving a topology change.
- SPF hold time - The device waits a specific amount of time between consecutive SPF calculations. By default, it waits 10 seconds. You can configure the SPF hold time to a value from 0 through 65535 seconds. If you set the SPF hold time to 0 seconds, the software does not wait between consecutive SPF calculations.

You can set the SPF delay and hold time to lower values to cause the device to change to alternate paths more quickly if a route fails. Note that lower values for these parameters require more CPU processing time.

You can change one or both of the timers.

NOTE
If you want to change only one of the timers, for example, the SPF delay timer, you must specify the new value for this timer as well as the current value of the SPF hold timer, which you want to retain. The device does not accept only one timer value.

NOTE
If you configure SPF timers between 0-100, they will default to 0 and be displayed incorrectly in the running configuration.

To change the SPF delay to 10 seconds and the SPF hold to 20 seconds, enter the following command.

```
device(config-ospf6-router)# timers spf 10 20
```

Syntax: [no] timers spf *delay hold-time*

For the *delay* and *hold-time* parameters, specify a value from 0 through 65535 seconds.

To set the timers back to their default values, enter the **no** version of this command.

Modifying administrative distance

The Brocade device can learn about networks from various protocols, including BGP4+, IPv6 IS-IS, RIPng, and OSPFv3. Consequently, the routes to a network may differ depending on the protocol from which the routes were learned. By default, the administrative distance for OSPFv3 routes is 110.

The device selects one route over another based on the source of the route information. To do so, the device can use the administrative distances assigned to the sources. You can influence the device's decision by changing the default administrative distance for OSPFv3 routes.

Configuring administrative distance based on route type

You can configure a unique administrative distance for each type of OSPFv3 route. For example, you can use this feature to influence the Brocade device to prefer a static route over an OSPF inter-area route and to prefer OSPF intra-area routes to static routes.

The distance you specify influences the choice of routes when the device has multiple routes to the same network from different protocols. The device prefers the route with the lower administrative distance.

You can specify unique default administrative distances for the following OSPFv3 route types:

- Intra-area routes
- Inter-area routes
- External routes

The default for all of these OSPFv3 route types is 110.

NOTE
This feature does not influence the choice of routes within OSPFv3. For example, an OSPF intra-area route is always preferred over an OSPF inter-area route, even if the intra-area route's distance is greater than the inter-area route's distance.

For example, to change the default administrative distances for intra-area routes to 80, inter-area routes to 90, and external routes to 100, enter the following commands.

```
device(config-ospf6-router)# distance intra-area 80
device(config-ospf6-router)# distance inter-area 90
device(config-ospf6-router)# distance external 100
```

Syntax: [no] distance {**external** | **inter-area** | **intra-area**} *distance*

The **external**, **inter-area**, and **intra-area** keywords specify the route type for which you are changing the default administrative distance.

The *distance* parameter specifies the new distance for the specified route type. You can specify a value from 1 through 255.

To reset the administrative distance of a route type to its system default, enter the **no** form of this command.

Configuring the OSPFv3 LSA pacing interval

The Brocade device paces OSPFv3 LSA refreshes by delaying the refreshes for a specified time interval instead of performing a refresh each time an individual LSA's refresh timer expires. The accumulated LSAs constitute a group, which the device refreshes and sends out together in one or more packets.

The pacing interval, which is the interval at which the Brocade device refreshes an accumulated group of LSAs, is configurable to a range from 10 through 1800 seconds (30 minutes). The default is 240 seconds (four minutes). Thus, every four minutes, the device refreshes the group of accumulated LSAs and sends the group together in the same packets.

The pacing interval is inversely proportional to the number of LSAs the Brocade device is refreshing and aging. For example, if you have approximately 10,000 LSAs, decreasing the pacing interval enhances performance. If you have a very small database (40 - 100 LSAs), increasing the pacing interval to 10 - 20 minutes might enhance performance only slightly.

To change the OSPFv3 LSA pacing interval to two minutes (120 seconds), enter the following command.

```
device(config)# ipv6 router ospf
device(config-ospf6-router)# timers lsa-group-pacing 120
```

Syntax: [no] timers lsa-group-pacing *seconds*

The *seconds* parameter specifies the number of seconds and can be from 10 through 1800 (30 minutes). The default is 240 seconds (four minutes).

To restore the pacing interval to its default value, use the **no** form of the command.

Modifying exit overflow interval

If a database overflow condition occurs on the Brocade device, the device eliminates the condition by removing entries that originated on the device. The exit overflow interval allows you to set how often a device checks to see if the overflow condition has been eliminated. The default value is 0. If the configured value of the database overflow interval is 0, then the device never leaves the database overflow condition.

For example, to modify the exit overflow interval to 60 seconds, enter the following command.

```
device(config-ospf6-router)# database-overflow-interval 60
```

Syntax: database-overflow-interval *seconds*

The *seconds* parameter can be a value from 0 through 86400 seconds (24 hours).

To reset the exit overflow interval to its system default, enter the **no** form of this command.

Modifying external link state database limit

By default, the link state database can hold a maximum of 2000 entries for external (type 5) LSAs. You can change the maximum number of entries from 500 - 8000. After changing this limit, make sure to save the running-config file and reload the software. The change does not take effect until you reload or reboot the software.

For example, to change the maximum number entries from the default of 2000 to 3000, enter the following command.

```
device(config-ospf6-router)# external-lsdb-limit 3000
```

Syntax: external-lsdb-limit *entries*

The *entries* parameter can be a numerical value from 500 through 8000 seconds.

To reset the maximum number of entries to its system default, enter the **no** form of this command.

Setting all OSPFv3 interfaces to the passive state

You can set all the Open Shortest Path First Version 3 (OSPFv3) interfaces to the default passive state using the **default-passive-interface** command**.** When you configure the interfaces as passive, the interfaces drop all the OSPFv3 control packets.

To set all the OSPFv3 interfaces to passive, enter the following commands.

```
device# configure terminal
device(config)# ipv6 router ospf vrf A
device(config-ospf6-router-vrf-A)# default-passive-interface
```

Syntax: [no] default-passive-interface

Modifying OSPFv3 interface defaults

OSPFv3 has interface parameters that you can configure. For simplicity, each of these parameters has a default value. No change to these default values is required except as needed for specific network configurations.

You can modify the default values for the following OSPF interface parameters:

- **cost:** Indicates the overhead required to send a packet across an interface. You can modify the cost to differentiate between 100 Mbps and 1000 Mbps (1 Gbps) links. The command syntax is **ipv6 ospf cost** *number*. The default cost is calculated by dividing 100 million by the bandwidth. For 10 Mbps links, the cost is 10. The cost for both 100 Mbps and 1000 Mbps links is 1, because the speed of 1000 Mbps was not in use at the time the OSPF cost formula was devised.
- **dead-interval:** Indicates the number of seconds that a neighbor router waits for a hello packet from the device before declaring the router down. The command syntax is **ipv6 ospf dead-interval** *seconds*. The value can be from 1 through 2147483647 seconds. The default is 40 seconds.
- **hello-interval:** Represents the length of time between the transmission of hello packets. The command syntax is **ipv6 ospf hello-interval** *seconds*. The value can be from 1 through 65535 seconds. The default is 10 seconds.

- **instance:** Indicates the number of OSPFv3 instances running on an interface. The command syntax is **ipv6 ospf instance** *number*. The value can be from 0 through 255. The default is 1.
- **MTU-ignore:** Allows you to disable a check that verifies the same MTU is used on an interface shared by neighbors. The command syntax is **ipv6 ospf mtu-ignore**. By default, the mismatch detection is enabled.
- **network:** Allows you to configure the OSPF network type. The command syntax is **ipv6 ospf network** [**point-to-multipoint**]. The default setting of the parameter depends on the network type.
- **passive:** When you configure an OSPF interface to be passive, that interface does not send or receive OSPF route updates. This option affects all IPv6 subnets configured on the interface. The command syntax is **ipv6 ospf passive**. By default, all OSPF interfaces are active and thus can send and receive OSPF route information. Since a passive interface does not send or receive route information, the interface is in effect a stub network.
- **active:** When you configure an OSPFv3 interface to be active, that interface sends or receives all the control packets and forms the adjacency. By default, the **ipv6 ospf active** command is disabled. Whenever you configure the OSPFv3 interfaces to be passive using the **default-passive-interface** command, all the OSPFv3 interfaces stop sending and receiving control packets. To send and receive packets over specific interfaces, you can use the **ipv6 ospf active** command.
- **priority:** Allows you to modify the priority of an OSPF router. The priority is used when selecting the designated router (DR) and backup designated routers (BDRs). The command syntax is **ipv6 ospf priority** *number*. The value can be from 0 through 255. The default is 1. If you set the priority to 0, the router does not participate in DR and BDR election.
- **retransmit-interval:** The time between retransmissions of LSAs to adjacent routers for an interface. The command syntax is **ipv6 ospf retransmit-interval** *seconds* . The value can be from 0 through 3600 seconds. The default is 5 seconds.
- **Transmit-delay:** The time it takes to transmit Link State Update packets on this interface. The command syntax is **ipv6 ospf transmit-delay***seconds*. The range is 0 through 3600 seconds. The default is 1 second.

Disabling or re-enabling event logging

OSPFv3 supports the logging of OSPFv3 events. The log-status change command controls the generation of all OSPFv3 logs. You can disable or re-enable the logging of events related to OSPFv3, such as neighbor state changes and database overflow conditions. By default, the Brocade device does not logs these events.

To disable the logging of events, enter the following command.

```
device(config-ospf6-router)# no log-status-change
```

Syntax: [no] log-status-change

To re-enable the logging of events, enter the following command.

```
device(config-ospf6-router)# log-status-change
```

IPsec for OSPFv3

This section describes the implementation of Internet Protocol Security (IPsec) for securing OSPFv3 traffic.

IPsec is available for OSPFv3 traffic only and only for packets that are "for-us." A for-us packet is addressed to one of the IPv6 addresses on the device or to an IPv6 multicast address. Packets that are just forwarded by the line card do not receive IPsec scrutiny.

Brocade devices support the following components of IPsec for IPv6-addressed packets:

- Authentication through Encapsulating Security Payload (ESP) in transport mode
- HMAC-SHA1-96 as the authentication algorithm
- Manual configuration of keys
- Configurable rollover timer

IPsec can be enabled on the following logical entities:

- Interface
- Area
- Virtual link

With respect to traffic classes, this implementation of IPSec uses a single security association (SA) between the source and destination to support all traffic classes and so does not differentiate between the different classes of traffic that the DSCP bits define.

IPsec on a virtual link is a global configuration. Interface and area IPsec configurations are more granular.

Among the entities that can have IPsec protection, the interfaces and areas can overlap. The interface IPsec configuration takes precedence over the area IPsec configuration when an area and an interface within that area use IPsec. Therefore, if you configure IPsec for an interface and an area configuration also exists that includes this interface, the interface's IPsec configuration is used by that interface. However, if you disable IPsec on an interface, IPsec is disabled on the interface even if the interface has its own, specific authentication.

Refer to Disabling IPsec on an interface on page 577 for more information.

For IPsec, the system generates two types of databases. The *security association database* (SAD) contains a security association for each interface or one global database for a virtual link. Even if IPsec is configured for an area, each interface that uses the area's IPsec still has its own security association in the SAD. Each SA in the SAD is a generated entry that is based on your specifications of an authentication protocol (ESP in the current release), destination address, and a security policy index (SPI). The SPI number is user-specified according to the network plan. Consideration for the SPI values to specify must apply to the whole network.

The system-generated security policy databases (SPDs) contain the security policies against which the system checks the for-us packets. For each for-us packet that has an ESP header, the applicable security policy in the security policy database (SPD) is checked to see if this packet complies with the policy. The IPsec task drops the non-compliant packets. Compliant packets continue on to the OSPFv3 task.

Configuring IPsec for OSPFv3

This section describes how to configure IPsec for an interface, area, and virtual link. It also describes how to change the key rollover timer if necessary and how to disable IPsec on a particular interface for special purposes.

By default, OSPFv3 IPsec authentication is disabled. The following IPsec parameters are configurable:

- ESP security protocol
- Authentication
- HMAC-SHA1-96 authentication algorithm
- Security parameter index (SPI)
- A 40-character key using hexadecimal characters
- An option for not encrypting the keyword when it appears in **show** command output
- Key rollover timer
- Specifying the key add remove timer

NOTE
In the current release, certain keyword parameters must be entered even though only one keyword choice is possible for that parameter. For example, the only authentication algorithm in the current release is HMAC-SHA1-96, but you must nevertheless enter the keyword for this algorithm. Also, ESP currently is the only authentication protocol, but you must still enter the **esp** keyword. This section describes all keywords.

General considerations

The IPsec component generates security associations and security policies based on certain user-specified parameters. The parameters are described with the syntax of each command in this section. User-specified parameters and their relation to system-generated values are as follows:

- **Security association:** based on your entries for *security policy index* (SPI), *destination address*, and *security protocol* (currently ESP), the system creates a security association for each interface or virtual link.
- **Security policy database:** based on your entries for SPI, *source address*, *destination addresses*, and *security protocol* , the system creates a security policy database for each interface or virtual link.
- You can configure the same SPI and key on multiple interfaces and areas, but they still have unique IPsec configurations because the SA and policies are added to each separate security policy database (SPD) that is associated with a particular interface. If you configure an SA with the same SPI in multiple places, the rest of the parameters associated with the SA—such as key, cryptographic algorithm, and security protocol, and so on—must match. If the system detects a mismatch, it displays an error message.
- IPsec authentication for OSPFv3 requires the use of multiple SPDs, one for each interface. A virtual link has a separate, global SPD. The authentication configuration on a virtual link must be different from the authentication configuration for an area or interface, as required by RFC4552. The interface number is used to generate a non-zero security policy database identifier (SPDID), but for the global SPD for a virtual link, the system-generated SPDID is always zero. As a hypothetical example, the SPD for interface eth 1/1 might have the system-generated SPDID of 1, and so on.
- If you change an existing key, you must also specify a different SPI value. For example, in an interface context where you intend to change a key, you must type a different SPI value—which occurs before the key parameter on the command line—before you type the new key.
- The old key is active for twice the current configured key-rollover-interval for the inbound direction. In the outbound direction, the old key remains active for a duration equal to the key-rollover-interval. If the key-rollover-interval is set to 0, the new key immediately takes effect for both directions.

Interface and area IPsec considerations

This section describes the precedence of interface and area IPsec configurations.

If you configure an interface IPsec by using the **ipv6 ospf authentication** command in the context of a specific interface, that interface's IPsec configuration overrides the area configuration of IPsec.

If you configure IPsec for an area, all interfaces that utilize the area-wide IPsec (where interface-specific IPsec is not configured) nevertheless receive an SPD entry (and SPDID number) that is unique for the interface.

The area-wide SPI that you specify is a constant for all interfaces in the area that use the area IPsec, but the use of different interfaces results in an SPDID and an SA that are unique to each interface. The security policy database depends partly on the source IP address, so a unique SPD for each interface results.

Considerations for IPsec on virtual links

The IPsec configuration for a virtual link is global, so only one security association database and one security policy database exist for virtual links if you choose to configure IPsec for virtual links.

The virtual link IPsec SAs and policies are added to all interfaces of the transit area for the outbound direction. For the inbound direction, IPsec SAs and policies for virtual links are added to the global database.

NOTE
The security association (SA), security protocol index (SPI), security protocol database (SPD), and key have mutual dependencies, as the subsections that follow describe.

Specifying the key rollover timer

Configuration changes for authentication takes effect in a controlled manner through the key rollover procedure as specified in RFC 4552, Section 10.1. The key rollover timer controls the timing of the existing configuration changeover. The key rollover timer can be configured in the IPv6 router OSPF context, as the following example illustrates.

```
device(config-ospf6-router)#key-rollover-interval 200
```

Syntax: key-rollover-interval *time*

The range for the key-rollover-interval is 0 through 14400 seconds. The default is 300 seconds.

Specifying the key add remove timer

The **key-add-remove** timer is used in an environment where interoperability with other vendors is required on a specific interface. This parameter is used to determine the interval time when authentication addition and deletion will take effect.

The **key-add-remove-interval** timer can be used to set the required value globally, or on a specific interface as needed. Interface configuration takes preference over system level configuration.

By default, the **key-add-remove-interval** is set to 300 seconds to smoothly interoperate with Brocade routers.

To set the **key-add-remove-interval** globally to 100 seconds, enter the following commands:

```
device(config-ospf6-router)# key-add-remove-interval 100
```

To set the **key-add-remove-interval** to 100 seconds at a specific interface, enter the following commands:

```
Brocade (config-if-e1000-1/10)#ipv6 ospf authentication ipsec key-add-remove-
interval 100
```

Syntax: [no] ipv6 ospf authentication ipsec key-add-remove-interval *range*

The **no** form of this command sets the key-add-remove-interval back to a default of 300 seconds.

The **ipv6** command is available in the configuration interface context for a specific interface.

The **ospf** keyword identifies OSPFv3 as the protocol to receive IPsec security.

The **authentication** keyword enables authentication.

The **ipsec** keyword specifies IPsec as the authentication protocol.

The *range* is a value between 0 and 14400 seconds.

This command is not set by default and **key-add-remove-interval** is set to the same value as **key-rollover-interval**.

NOTE
This command will not resolve the issue completely on a network where Brocade Routers running software that does not support **key-add-remove-interval** (earlier versions of NetIron R05.3.00) and other vendor's routers are present. In this case, disabling and enabling the interface or setting **key-rollover-interval** to 0 will resolve the issue.

Configuring IPsec on a interface

For IPsec to work, the IPsec configuration must be the same on all the routers to which an interface connects.

For multicast, IPsec does not need or use a specific destination address, the destination address is "do not care," and this status is reflected by the lone pair of colons (::) for destination address in the **show** command output.

To configure IPsec on an interface, proceed as in the following example.

NOTE
The IPsec configuration for an interface applies to the inbound and outbound directions. Also, the same authentication parameters must be used by all devices on the network to which the interface is connected, as described in section 7 of RFC 4552.

```
device(config-if-e10000-1/2)#ipv6 ospf auth ipsec spi 429496795 esp sha1
abcdef12345678900987654321fedcba12345678
```

Syntax: [no] ipv6 ospf authentication ipsec spi *spi-num* **esp sha1** [**no-encrypt**] *key*

The **no** form of this command deletes IPsec from the interface.

The **ipv6** command is available in the configuration interface context for a specific interface.

The **ospf** keyword identifies OSPFv3 as the protocol to receive IPsec security.

The **authentication** keyword enables authentication.

The **ipsec** keyword specifies IPsec as the authentication protocol.

The **spi** keyword and the *spi-num* variable specify the security parameter that points to the security association. The near-end and far-end values for *spi-num* must be the same. The range for *spi-num* is decimal 256 through 4294967295.

The mandatory **esp** keyword specifies ESP (rather than authentication header) as the protocol to provide packet-level security. In the current release, this parameter can be **esp** only.

The **sha1** keyword specifies the HMAC-SHA1-96 authentication algorithm. This mandatory parameter can be only the **sha1** keyword in the current release.

Including the optional **no-encrypt** keyword means that when you display the IPsec configuration, the key is displayed in its unencrypted form and also saved as unencrypted.

The *key* variable must be 40 hexadecimal characters. To change an existing key, you must also specify a different SPI value. You cannot just change the key without also specifying a different SPI, too. For example, in an interface context where you intend to change a key, you must type a different SPI value -- which occurs before the key parameter on the command line -- before you type the new key.

If **no-encrypt** is not entered, then the key will be encrypted. This is the default. The system adds the following in the configuration to indicate that the key is encrypted:

- encrypt = the key string uses proprietary simple cryptographic 2-way algorithm (only for Brocade NetIron CES and Brocade NetIron CER devices)
- encryptb64 = the key string uses proprietary base64 cryptographic 2-way algorithm (only for Brocade NetIron XMR and Brocade MLX series devices)

This example results in the configuration shown in the screen output that follows. Note that because the optional **no-encrypt** keyword was omitted, the display of the key has the encrypted form by default.

```
interface ethernet 1/2
 enable
 ip address 10.3.3.1/8
 ipv6 address 2001:db8:3::1/64
 ipv6 ospf area 1
 ipv6 ospf authentication ipsec spi 429496795 esp sha1 encryptb64
$ITJkQG5HWnw4M09tWVd
```

Configuring IPsec for an area

This application of the **area** command (for IPsec) applies to all of the interfaces that belong to an area unless an interface has its own IPsec configuration. The interface IPsec can be operationally disabled if necessary.) To configure IPsec for an area in the IPv6 router OSPF context, proceed as in the following example.

```
device(config-ospf6-router)#area 2 auth ipsec spi 400 esp sha1
abcef12345678901234fedcba098765432109876
```

Syntax: [no] area *area-id* **authentication ipsec spi** *spi-num* **esp sha1** [**no-encrypt**] *key*

The **no** form of this command deletes IPsec from the area.

The **area** command and the *area-id* variable specify the area for this IPsec configuration. The *area-id* can be an integer in the range 0 through 2,147,483,647 or have the format of an IP address.

The **authentication** keyword specifies that the function to specify for the area is packet authentication.

The **ipsec** keyword specifies that IPsec is the protocol that authenticates the packets.

The **spi** keyword and the *spi-num* variable specify the index that points to the security association. The near-end and far-end values for *spi-num* must be the same. The range for *spi-num* is decimal 256 through 4294967295.

The mandatory **esp** keyword specifies ESP (rather than authentication header) as the protocol to provide packet-level security. In the current release, this parameter can be **esp** only.

The **sha1** keyword specifies the HMAC-SHA1-96 authentication algorithm. This mandatory parameter can be only the **sha1** keyword in the current release.

Including the optional **no-encrypt** keyword means that the 40-character key is not encrypted upon either its entry or its display. The key must be 40 hexadecimal characters.

If **no-encrypt** is not entered, then the key will be encrypted. This is the default. The system adds the following in the configuration to indicate that the key is encrypted:

- encrypt = the key string uses proprietary simple cryptographic 2-way algorithm (only for Brocade NetIron CES and Brocade NetIron CER devices)
- encryptb64 = the key string uses proprietary base64 cryptographic 2-way algorithm (only for Brocade NetIron XMR and Brocade MLX series devices)

The configuration in the preceding example results in the configuration for area 2 that is illustrated in the following.

```
ipv6 router ospf
 area 0
 area 1
 area 2
 area 2 auth ipsec spi 400 esp sha1 abcef12345678901234fedcba098765432109876
```

Configuring IPsec for a virtual link

IPsec on a virtual link has a global configuration.

To configure IPsec on a virtual link, enter the IPv6 router OSPF context of the CLI and proceed as the following example illustrates. (Note the **no-encrypt** option in this example.)

```
device(config-ospf6-router)#area 1 vir 10.2.2.2 auth ipsec spi 360 esp sha1 no-
encrypt 12345678900987654321123456789090987654321
```

Syntax: [no] area *area-id* **virtual** *nbr-id* **authentication ipsec spi** *spi-num* **esp sha1** [**no-encrypt**] *key*

The **no** form of this command deletes IPsec from the virtual link.

The **area** command and the *area-id* variable specify the area is to be configured. The *area-id* can be an integer in the range 0 through 2,147,483,647 or have the format of an IP address.

The **virtual** keyword indicates that this configuration applies to the virtual link identified by the subsequent variable *nbr-id*. The variable *nbr-id* is in dotted decimal notation of an IP address.

The **authentication** keyword specifies that the function to specify for the area is packet authentication.

The **ipsec** keyword specifies that IPsec is the protocol that authenticates the packets.

The **spi** keyword and the *spi-num* variable specify the index that points to the security association. The near-end and far-end values for *spi-num* must be the same. The range for *spi-num* is decimal 256 through 4294967295.

The mandatory **esp** keyword specifies ESP (rather than authentication header) as the protocol to provide packet-level security. In the current release, this parameter can be **esp** only.

The **sha1** keyword specifies the HMAC-SHA1-96 authentication algorithm. This mandatory parameter can be only the **sha1** keyword in the current release.

Including the optional **no-encrypt** keyword means that the 40-character key is not encrypted in **show** command displays. If **no-encrypt** is not entered, then the key will be encrypted. This is the default. The system adds the following in the configuration to indicate that the key is encrypted:

- encrypt = the key string uses proprietary simple cryptographic 2-way algorithm (only for Brocade NetIron CES and Brocade NetIron CER devices)
- encryptb64 = the key string uses proprietary base64 cryptographic 2-way algorithm (only for Brocade NetIron XMR and Brocade MLX series devices)

This example results in the following configuration.

```
area 1 virtual-link 10.2.2.2
area 1 virtual-link 10.2.2.2 authentication ipsec spi 360 esp sha1 no-encrypt 12
34567890098765432112345678990987654321
```

Disabling IPsec on an interface

For the purpose of troubleshooting, you can operationally disable IPsec on an interface by using the **ipv6 ospf authentication ipsec disable** command in the CLI context of a specific interface. This command disables IPsec on the interface whether its IPsec configuration is the area's IPsec

configuration or is specific to that interface. The output of the **show ipv6 ospf interface command** shows the current setting for the disable command.

To disable IPsec on an interface, go to the CLI context of the interface and proceed as in the following example.

```
device(config-if-e10000-1/2)#ipv6 ospf auth ipsec disable
```

Syntax: [no] ipv6 ospf authentication ipsec disable

The **no** form of this command restores the area and interface-specific IPsec operation.

Changing the key rollover timer

Configuration changes for authentication takes effect in a controlled manner through the key rollover procedure as specified in RFC 4552, Section 10.1. The key rollover timer controls the timing of the configuration changeover. The key rollover timer can be configured in the IPv6 router OSPF context, as the following example illustrates.

```
device(config-ospf6-router)#key-rollover-interval 200
```

Syntax: key-rollover-interval *time*

The range for the key-rollover-interval is 0 through 14400 seconds. The default is 300 seconds.

Clearing IPsec statistics

This section describes the **clear ipsec statistics** command for clearing statistics related to IPsec. The command resets to 0 the counters (which you can view as a part of IP Security Packet Statistics). The counters hold IPsec packet statistics and IPsec error statistics. The following example illustrates the **show ipsec statistics** output.

```
device#show ipsec statistics
                    IPSecurity Statistics
secEspCurrentInboundSAs 1          ipsecEspTotalInboundSAs:  2
secEspCurrentOutboundSA 1          ipsecEspTotalOutboundSAs: 2
                  IPSecurity Packet Statistics
secEspTotalInPkts:      20         ipsecEspTotalInPktsDrop:  0
secEspTotalOutPkts:     84
                  IPSecurity Error Statistics
secAuthenticationErrors 0
secReplayErrors:        0          ipsecPolicyErrors:        13
secOtherReceiveErrors:  0          ipsecSendErrors:          0
secUnknownSpiErrors:    0
```

To clear the statistics, enter the **clear ipsec statistics** command as in the following example.

```
device#clear ipsec statistics
```

Syntax: clear ipsec statistics

This command takes no parameters.

Configuring OSPFv3 Graceful Restart Helper mode

To enable the graceful restart (GR) helper capability, use the **graceful-restart helper** command in the OSPFv6 interface mode. Graceful restart for OSPFv3 helper mode is enabled by default.

```
device(config-ospf6-router)# graceful-restart helper strict-lsa-checking
```

Syntax: [no] graceful-restart helper {**disable** | **strict-lsa-checking**}

The **disable** keyword is used to disable the graceful-restart helper capability. By default, it is enabled.

The **strict-lsa-checking** keyword is used to enable the graceful-restart helper device to terminate restart supporting any topology change. By default, it is disabled.

TABLE 116 OSPFv3 area information fields

Task	Configuration example
Disabling graceful-restart-helper on a device	`device(config-ospf6-router)#graceful-restart helper disable`
NOTE Graceful restart for OSPFv3 helper mode is enabled by default.	
Enabling graceful-restart-helper on a device	`device(config-ospf6-router)#no graceful-restart helper disable`
Enabling LSA checking option on the helper	`device(config-ospf6-router)#graceful-restart helper strict-lsa-checking`
Enabling graceful-restart-helper per VRF	`device(config-ospf6-router-vrf-red)#graceful-restart helper strict-lsa-checking`
NOTE Graceful-restart-helper option can be enabled or disabled per VRF in OSPFv3. If configured outside VRF, then it is applicable to the default VRF instance of OSPFv3.	

Configuring OSPFv3 Non-stop routing (NSR)

In graceful restart, the restarting neighbors need to help build the routing information during the failover, but the graceful restart helper may not be supported by all devices in a network. Hence to eliminate this dependency, the non-stop routing (NSR) feature is supported on Brocade devices. NSR does not require support from neighboring devices to perform hitless failover. NSR does not support IPv6-over-IPv4 tunnel and vitual link, so traffic loss is expected while performing hitless failover.

To enable NSR for OSPFv3, use the **nonstop-routing** command in the OSPFv6 interface mode.

```
device(config)# ipv6 router ospf
device(config-ospf6-router)# nonstop-routing
```

To disable NSR for OSPFv3, use the **no** form of the **nonstop-routing** command.

Syntax: [no] nonstop-routing

Configuring OSPFv3 max-metric router LSA

The OSPFv3 max-metric router LSA feature enables OSPFv3 to advertise its locally generated router LSAs with a maximum metric to direct transit traffic away from the router, while still routing for directly connected networks. By advertising the maximum metric, the router will not attract transit traffic. A router which does not handle transit traffic and only forwards packets destined for its directly connected

links is known as a stub router. In OSPFv3 networks, a device could be placed in a stub router role by advertising large metrics for its connected links, so that the cost of a path through the device becomes larger than that of an alternative path.

You can configure OSPFv3 max-metric router LSA in either startup or non-startup mode. Configuring max-metric on startup may be helpful on ASBRs where protocols such as BGP converge after OSPF converges. Configuring max-metric on non-startup may be helpful in database overflow scenarios.

Max metric router LSA is configured in an incremental fashion. To configure OSPFv3 to advertise router LSAs with the cost of point-to-point and transit links set to 65535, enter the following command.

```
device(config-ospf6-router)# max-metric router-lsa
```

To modify OSPFv3 to advertise intra-area-prefix LSAs with the cost of stubs set to 65535 and the cost of external LSAs set to 16711680, enter the following command:

```
device(config-ospf6-router)# max-metric router-lsa include-stub external-lsa
```

To modify OSPFv3 to advertise summary type-3 and type-4 LSAs with the cost set to 10000, enter the following command. Executing this command will not alter the existing include-stub and external-lsa configuration.

```
device(config-ospf6-router)# max-metric router-lsa summary-lsa 10000
```

Although max-metric router LSA configuration is done in an incremental fashion, the show run command displays the configuration in just one line. For example, after executing the three configuration commands above, the output of the show run command would be:

```
device(config-ospf6-router)# max-metric router-lsa include-stub summary-lsa 10000
external-lsa
```

To remove the include-stub and summary-lsa options from this configuration, enter the following command.

```
device(config-ospf6-router)# no max-metric router-lsa include-stub summary-lsa
```

The output of show run would then be:

```
device(config-ospf6-router)# max-metric router-lsa external-lsa
```

By default, when max-metric router LSA is configured, OSPFv3 always advertises the maximum metric for router LSAs and external LSAs. To modify OSPFv3 to advertise the max-metric for a period of 60 seconds only on startup, enter the following command.

```
device(config-ospf6-router)# max-metric router-lsa on-startup 60
```

The output of show run would then be:

```
device(config-ospf6-router)# max-metric router-lsa external-lsa on-startup 60
```

OSPFv3 can also be configured to advertise the max-metric for stub on startup using the following command:

```
device(config-ospf6-router)# max-metric router-lsa include-stub
```

OSPFv3 would then be configured to advertise the max-metric for stub, router LSA and summary LSA on startup and the output of show run would be:

```
device(config-ospf6-router)# max-metric router-lsa include-stub external-lsa on-
startup 60
```

To remove the on-startup option alone, so that OSPFv3 will always advertise the max-metric for router LSA, intra-area-prefix LSA and summary LSA, enter the following command:

```
device(config-ospf6-router)# no max-metric router-lsa on-startup
```

The output of show run would then be:

```
device(config-ospf6-router)# max-metric router-lsa include-stub external-lsa
```

Syntax: [no] max-metric router-lsa [**include-stub**][**summary-lsa** [*max-metric-value*]] [**external-lsa** [*max-metric-value*]] [**on-startup** {*seconds* | **wait-for-bgp**}]

The **router-lsa** parameter configures the device to advertise the maximum metric for point-to-point and transit links. The maximum metric value is 65535.

The **include-stub** parameter specifies the advertisement of the maximum metric value (65535) forpoint-to-point and broadcast stub links in the intra-area-prefix LSA.

NOTE
You cannot specify a maximum metric value with the **include-stub** parameter. If you specify **include-stub**, point-to-point and broadcast stub links in the intra-area-prefix LSA are advertised at a value of 65535.

The **summary-lsa** [*max-metric-value*] parameter configures the maximum metric value for inter-area-prefix type-3 and type-4 LSAs. The range is from 1 through 16777215. The default value is 16711680.

NOTE
Setting the *max-metric-value* for **summary-lsa** to 16777215 makes the route unreachable.

The **external-lsa** [*max-metric-value*] parameter configures the maximum metric value for external type-5 and type-7 LSAs. The range is from 1 through 16777215. The default value is 16711680.

NOTE
Setting the *max-metric-value* for **external-lsa** to 16777215 makes the route unreachable.

The **on-startup** {*seconds* | **wait-for-bgp**} parameter specifies the advertisement of the maximum metric for a limited period only, on startup. The *seconds* variable specifies the length of time in seconds. The range is from 5 through 86400. The **wait-for-bgp** parameter specifies that the maximum metric is advertised until BGP converges or for 600 seconds. When the **on-startup** option is not specified, a device configured with **max-metric router-lsa** always advertises the max-metric.

The **no** form of the command removes the configuration.

NOTE
The **on-startup** configuration does not apply to NSR restarts.

Displaying OSPFv3 information

You can display the information for the following OSPFv3 parameters:

- Areas
- Link state databases
- Interfaces
- Memory usage
- Neighbors
- Redistributed routes
- Routes
- SPF
- Virtual links
- Virtual neighbors
- IPsec
- key-add-remove interval

General OSPFv3 configuration information

To indicate whether the Brocade device is operating as ASBR or not, enter the following command at any CLI level.

```
device#show ipv6 ospf
OSPFv3 Process number 0 with Router ID 0x10010101(10.1.1.1)
    Running 0 days 0 hours 1 minutes 53 seconds
    Number of AS scoped LSAs is 3
    Sum of AS scoped LSAs Checksum is fabdd4de
 External LSA Limit is 250000
 Database Overflow Interval is 10
 Database Overflow State is NOT OVERFLOWED
 Route calculation executed 0 times
 Pending outgoing LSA count 0
 Authentication key rollover interval 30 seconds
    Authentication key add/remove interval 0 seconds
 Number of areas in this router is 3
 Router is operating as ABR
 Router is operating as ASBR, Redistribute: CONNECTED
 High Priority Message Queue Full count: 0
 BFD is disabled
```

The output of the **show ipv6 ospf** command indicates if the Brocade device is operating as ASBR. If the device is not operating as ASBR, then there is no information about redistribution in the output.

Displaying OSPFv3 area information

To display global OSPFv3 area information for the device, enter the following command at any CLI level.

```
device# show ipv6 ospf area
Area 0:
  Interface attached to this area: loopback 2 ethe 3/2 tunnel 2
        Number of Area scoped LSAs is 6
        Statistics of Area 0:
            SPF algorithm executed 16 times
            SPF last updated: 335256 sec ago
            Current SPF node count: 3
                Router: 2 Network: 1
                Maximum of Hop count to nodes: 2
```

Syntax: show ipv6 ospf area [*area-id*]

You can specify the *area-id* parameter in the following formats:

- As an IPv4 address, for example, 192.168.1.1.
- As a numerical value from 0 through 2,147,483,647.

The *area-id* parameter restricts the display to the specified OSPF area.

TABLE 117 show ipv6 ospf area output descriptions

This field	Displays
Area	The area number.
Interface attached to this area	The device interfaces attached to the area.
Number of Area scoped LSAs is *N*	Number of LSAs (*N*) with a scope of the specified area.

TABLE 117 show ipv6 ospf area output descriptions (Continued)

This field	Displays
SPF algorithm executed is *N*	The number of times (*N*) the OSPF Shortest Path First (SPF) algorithm is executed within the area.
SPF last updated	The interval in seconds that the SPF algorithm was last executed within the area.
Current SPF node count	The current number of SPF nodes in the area.
Router	Number of router LSAs in the area.
Network	Number of network LSAs in the area.
Indx	The row number of the entry in the routers's OSPF area table.
Statistics of Area	The number of the area whose statistics are displayed.
Maximum hop count to nodes.	The maximum number of hop counts to an SPF node within the area.

Displaying OSPFv3 database information

You can display a summary of the device's link state database or detailed information about a specified LSA type.

To display a summary of a device's link state database, enter the following command at any CLI level.

```
device# show ipv6 ospf database
LSA Key - Rtr:Router Net:Network Inap:InterPrefix Inar:InterRouter
          Extn:ASExternal Grp:GroupMembership Typ7:Type7 Link:Link
          Iap:IntraPrefix Grc:Grace
Area ID     Type LSID     Adv Rtr      Seq(Hex)  Age  Cksum Len   Sync
0           Iap  0        10.1.1.1     80000001  3    343d  52    Yes
0           Iap  0        10.2.2.2     80000001  8    c61d  58    No
```

Syntax: show ipv6 ospf database [**advrtr** *ipv4-address* | **as-external** [**advrtr** *ipv4-address* | **link-id** *number*] | **extensive** | **inter-prefix** [**advrtr** *ipv4-address* | **link-id** *number*] | **inter-router** [**advrtr** *ipv4-address* | **link-id** *number*] | **intra-prefix** [**advrtr** *ipv4-address* | **link-id** *number*] | **link** [**advrtr** *ipv4-address* | **link-id** *number*] | **link-id** *number* | **network** [**advrtr** *ipv4-address* | **link-id** *number*] | **router** [**advrtr** *ipv4-address* | **link-id** *number*]]

The **advrtr** *ipv4-address* parameter displays detailed information about the LSAs for a specified advertising router only.

The **as-external** keyword displays detailed information about the AS externals LSAs only.

The **extensive** keyword displays detailed information about all LSAs in the database.

The **inter-prefix** keyword displays detailed information about the inter-area prefix LSAs only.

The **inter-router** keyword displays detailed information about the inter-area router LSAs only.

The **intra-prefix** keyword displays detailed information about the intra-area prefix LSAs only.

The **link** keyword displays detailed information about the link LSAs only.

The **link-id** *number* parameter displays detailed information about the specified link LSAs only.

The **network** *number* displays detailed information about the network LSAs only.

The **router** *number* displays detailed information about the router LSAs only.

The **scope** *area-id* parameter displays detailed information about the LSAs for a specified area, AS, or link.

TABLE 118 show ipv6 ospf database output descriptions

This field	Displays
Area ID	The OSPF area in which the device resides.
Type	Type of LSA. LSA types can be the following: • Rtr - Router LSAs (Type 1). • Net - Network LSAs (Type 2). • Inap - Inter-area prefix LSAs for ABRs (Type 3). • Inar - Inter-area router LSAs for ASBRs (Type 4). • Extn - AS external LSAs (Type 5). • Link - Link LSAs (Type 8). • Iap - Intra-area prefix LSAs (Type 9).
LS ID	The ID of LSA in Decimal.
Adv Rtr	The device that advertised the route.
Seq(Hex)	The sequence number of the LSA. The OSPF neighbor that sent the LSA stamps it with a sequence number to enable the device and other OSPF routers to determine which LSA for a given route is the most recent.
Age	The age of the LSA, in seconds.
Chksum	A checksum for the LSA packet. The checksum is based on all the fields in the packet except the age field. The device uses the checksum to verify that the packet is not corrupted.
Len	The length, in bytes, of the LSA.
Sync	Sync status with the slave management processor (MP).

To display the **show ipv6 ospf database advr** command output, enter the following command at any CLI level.

```
device# show ipv6 ospf database advr 10.4.4.4
LSA Key - Rtr:Router Net:Network Inap:InterPrefix Inar:InterRouter
          Extn:ASExternal Grp:GroupMembership Typ7:Type7 Link:Link
          Iap:IntraPrefix
Area ID     Type LSID    Adv Rtr     Seq(Hex)  Age  Cksum  Len
1           Iap     0      10.4.4.4    80000001  1085  99fa  44

   Number of Prefix: 1
             Referenced LS Type: Router
             Referenced LS ID: 0
             Referenced Advertising Router: 10.4.4.4
             Prefix Options: Metric: 1
             Prefix: 2001:db8:11::/64

LSA Key - Rtr:Router Net:Network Inap:InterPrefix Inar:InterRouter
```

```
          Extn:ASExternal Grp:GroupMembership Typ7:Type7 Link:Link
          Iap:IntraPrefix
Area ID      Type LSID    Adv Rtr     Seq(Hex)  Age  Cksum  Len
1            Typ7     1     10.4.4.4    80000001  394  a8a6   36

   Bits: N--
           Metric: 1
   Prefix Options:
           Referenced LSType: 0
           Prefix: 2001:db8:11::/64
```

To display detailed information about all LSAs in the database, enter the following command at any CLI level.

```
device# show ipv6 ospf database extensive
LSA Key - Rtr:Router Net:Network Inap:InterPrefix Inar:InterRouter
          Extn:ASExternal Grp:GroupMembership Typ7:Type7 Link:Link
          Iap:IntraPrefix Grc:Grace
Area ID    Type LS ID     Adv Rtr     Seq(Hex)  Age  Cksum  Len
0          Link 00000031  10.1.1.1    80000001  35   6db9   56
   Router Priority: 1
   Options: V6E---R--
   LinkLocal Address: fe80::1
   Number of Prefix: 1
   Prefix Options:
   Prefix: 2001:db8:3002::/64

Area ID   Type LS ID      Adv Rtr         Seq(Hex)  Age  Cksum  Len
0         Iap     00000159  10.223.223.223  800000ab  357  946b   56
   Number of Prefix: 2
   Referenced LS Type: Network
   Referenced LS ID: 00000159
   Referenced Advertising Router: 10.223.223.223
   Prefix Options: Metric: 0
   Prefix: 2001:db8:2000:4::/64
   Prefix Options: Metric: 0
   Prefix: 2001:db8:46a::/64

Area ID   Type LS ID      Adv Rtr         Seq(Hex)  Age  Cksum  Len
0         Rtr 00000039    10.223.223.223  800000b1  355   8f2d   40
   Capability Bits: --EOptions:
   V6E---R--
   Type: Transit Metric: 1
   Interface ID: 00000058 Neighbor Interface ID: 00000058
   Neighbor Router ID: 10.223.223.223

Area ID  Type LS ID     Adv Rtr         Seq(Hex) Age  Cksum  Len
0        Net     000001f4 10.223.223.223  800000ab 346  190a   32
   Options: V6E---R--
   Attached Router: 10.223.223.223
   Attached Router: 10.1.1.1

Area ID   Type LS ID     Adv Rtr         Seq(Hex) Age  Cksum  Len
N/A       Extn 000001df  10.223.223.223  800000af 368  0aa8   32
   Bits: E
   Metric: 00000001
   Prefix Options:
   Referenced LSType: 0
   Prefix: 2001:db8::/32

Area ID  Type LS ID     Adv Rtr      Seq(Hex) Age  Cksum  Len
1        Inap 0000011d  10.1.1.188   80000001 124  25de   36
   Metric: 2
   Prefix Options:
   Prefix: 2001:db8:2::/64

Area ID  Type  LS ID     Adv Rtr     Seq(Hex)  Age  Cksum  Len
0        Inar  0000005b  10.1.1.198  80000001  990  dbad   32
   Options: V6E---R--
   Metric: 1
   Destination Router ID:10.1.1.188
```

NOTE
Portions of this display are truncated for brevity. The purpose of this display is to show all possible fields that might display rather than to show complete output.

The fields that display depend upon the LSA type as shown in the following.

TABLE 119 OSPFv3 detailed database information fields

This field	Displays
Router LSA (Type 1) (Rtr) Fields	
Capability Bits	A bit that indicates the capability of the device. The bit can be set to one of the following: B - The device is an area border router. E - The device is an AS boundary router. V - The device is a virtual link endpoint. W - The device is a wildcard multicast receiver.
Options	A 24-bit field that enables IPv6 OSPF routers to support the optional capabilities. When set, the following bits indicate the following: V6 - The device should be included in IPv6 routing calculations. E - The device floods AS-external-LSAs as described in RFC 2740. MC - The device forwards multicast packets as described in RFC 1586. N - The device handles type 7 LSAs as described in RFC 1584. R - The originator is an active router. DC -The device handles demand circuits.
Type	The type of interface. Possible types can be the following: Point-to-point - A point-to-point connection to another router. Transit - A connection to a transit network. Virtual link - A connection to a virtual link.
Metric	The cost of using this router interface for outbound traffic.
Interface ID	The ID assigned to the router interface.
Neighbor Interface ID	The interface ID that the neighboring router has been advertising in hello packets sent on the attached link.
Neighbor Router ID	The router ID (IPv4 address) of the neighboring router that advertised the route. (By default, the router ID is the IPv4 address configured on the lowest numbered loopback interface. If the device does not have a loopback interface, the default router ID is the lowest numbered IPv4 address configured on the device.)

TABLE 119 OSPFv3 detailed database information fields (Continued)

This field	Displays
Network LSA (Type 2) (Net) Fields	
Options	A 24-bit field that enables IPv6 OSPF routers to support the optional capabilities. When set, the following bits indicate the following: V6 - The device should be included in IPv6 routing calculations. E - The device floods AS-external-LSAs as described in RFC 2740. MC - The device forwards multicast packets as described in RFC 1586. N - The device handles type 7 LSAs as described in RFC 1584. R - The originator is an active router. DC -The device handles demand circuits.
Attached Router	The address of the neighboring router that advertised the route.
Inter-Area Prefix LSA (Type 3) (Inap) Fields	
Metric	The cost of the route.
Prefix Options	An 8-bit field describing various capabilities associated with the prefix.
Prefix	The IPv6 prefix included in the LSA.
Inter-Area Router LSA (Type 4) (Inar) Fields	
Options	A 24-bit field that enables IPv6 OSPF routers to support the optional capabilities. When set, the following bits indicate the following: V6 - The device should be included in IPv6 routing calculations. E - The device floods AS-external-LSAs as described in RFC 2740. MC - The device forwards multicast packets as described in RFC 1586. N - The device handles type 7 LSAs as described in RFC 1584. R - The originator is an active router. DC -The device handles demand circuits.
Metric	The cost of the route.
Destination Router ID	The ID of the router described in the LSA.
AS External LSA (Type 5) (Extn) Fields	
Bits	The bit can be set to one of the following: • E - If bit E is set, a Type 2 external metric. If bit E is zero, a Type 1 external metric. • F - A forwarding address is included in the LSA. • T - An external route tag is included in the LSA.

TABLE 119 OSPFv3 detailed database information fields (Continued)

This field	Displays
Metric	The cost of this route, which depends on bit E.
Prefix Options	An 8-bit field describing various capabilities associated with the prefix.
Referenced LS Type	If non-zero, an LSA with this LS type is associated with the LSA.
Prefix	The IPv6 prefix included in the LSA.
Link LSA (Type 8) (Link) Fields	
Router Priority	The router priority of the interface attaching the originating router to the link.
Options	The set of options bits that the router would like set in the network LSA that will be originated for the link.
Link Local Address	The originating router's link-local interface address on the link.
Number of Prefix	The number of IPv6 address prefixes contained in the LSA.
Prefix Options	An 8-bit field of capabilities that serve as input to various routing calculations: • NU - The prefix is excluded from IPv6 unicast calculations. • LA - The prefix is an IPv6 interface address of the advertising router. • MC - The prefix is included in IPv6 multicast routing calculations. • P - NSSA area prefixes are readvertised at the NSSA area border.
Prefix	The IPv6 prefix included in the LSA.
Intra-Area Prefix LSAs (Type 9) (Iap) Fields	
Number of Prefix	The number of prefixes included in the LSA.
Referenced LS Type, Referenced LS ID	Identifies the router-LSA or network-LSA with which the IPv6 address prefixes are associated.
Referenced Advertising Router	The address of the neighboring router that advertised the route.
Prefix Options	An 8-bit field describing various capabilities associated with the prefix.
Metric	The cost of using the advertised prefix.
Prefix	The IPv6 prefix included in the LSA.
Number of Prefix	The number of prefixes included in the LSA.

Displaying IPv6 interface information

You can use the following command to display a summary of IPv6 Interface information.

```
device# show ipv6 interface
Type Codes - I:ISIS  O:OSPF R:RIP
Interface   Stat/Prot IGPs IPv6 Address                                        VRF
eth 3/20    up/up          fe80::2c0:12ff:fe34:5073                default-vrf
                           2001:db8:1000::1/64
                           2001:db8:1000::/64[Anycast]
```

Syntax: show ipv6 interface [**ethernet** *port* | **loopback** *number* | **tunnel** *number* | **ve** *number*]

The **ethernet**, **loopback**, **tunnel**, and **ve** parameters specify the interface for which to display information. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a loopback, tunnel, or VE interface, also specify the number associated with the interface.

TABLE 120 show ipv6 interface output descriptions

Field	Description
Type Codes	Shows the routing protocol enabled on the interface. The routing protocol can be one of the following: • R - RIP • O - OSPF • I – IS-IS
Interface	Shows the type, slot, and port number of the interface.
Stat/Prot	Shows the status of the link and the protocol for the interface. The status can be one of the following: • Up • Down
IGPs	Shows the type of the Interior Gateway Protocols (IGPs) enabled on the interface.
IPv6 Address	Shows the link local IPv6 address configured for the interface.
VRF	Specifies the VRF to which the interface belongs.

Displaying IPv6 OSPFv3 interface information

IPv6 Interface information can be displayed in either a brief or full mode. The following sections describe the command to display these modes and the resulting output:

- Displaying IPv6 OSPFv3 Interface Information in Brief Mode
- Displaying IPv6 OSPFv3 Interface Information in Full Mode

Displaying IPv6 OSPFv3 interface information in brief mode

You can use the following command to display a summary of IPv6 Interface information.

```
device# show ipv6 ospf interface brief
Interface  Area           Status  Type Cost  State     Nbrs (F/C)
eth 1/1    0              up      BCST 1     DROther   1/1
loopback 1 0              up      BCST 1     Loopback  0/0
```

Syntax: show ipv6 ospf interface brief

TABLE 121 show ipv6 ospf interface brief output descriptions

This field	Displays
Interface	The interface type, and the port number or number of the interface.
Area	The OSPF area configured on the interface.
Status	The status of the link and the protocol. Possible status include the following: • Up. • Down.
Type	The type of OSPFv3 circuit running on the interface. Possible types include the following: • BCST- Broadcast interface type • P2P- Point-to-point interface type • UNK- The interface type is not known at this time
Cost	The overhead required to send a packet across an interface.
State	The state of the interface. Possible states include the following: • DR - The interface is functioning as the Designated Router for OSPFv3. • BDR - The interface is functioning as the Backup Designated Router for OSPFv3. • Loopback - The interface is functioning as a loopback interface. • P2P - The interface is functioning as a point-to-point interface. • Passive - The interface is up but it does not take part in forming an adjacency. • Waiting - The interface is trying to determine the identity of the BDR for the network. • None - The interface does not take part in the OSPF interface state machine. • Down - The interface is unusable. No protocol traffic can be sent or received on such a interface. • DR other - The interface is a broadcast or NBMA network on which another router is selected to be the DR.
Nbrs (F/C)	The number of adjacent neighbor routers. The number to the left of the "/" are the neighbor routers that are fully adjacent and the number to the right represents all adjacent neighbor routers.

Displaying IPv6 OSPFv3 interface information in full mode

You can display detailed information about all OSPFv3 interfaces by using the **show ipv6 ospf interface** command, as the following truncated example illustrates.

```
device#show ipv6 ospf interface
eth 1/3 is down, type BROADCAST
```

```
 Interface is disabled
eth 1/8 is up, type BROADCAST
 IPv6 Address:
  2001:db8:18:18:18::1/64
  2001:db8:18:18::/64
 Instance ID 255, Router ID 10.1.1.1
 Area ID 1, Cost 1
  State Active(default passive) DR, Transmit Delay 1 sec, Priority 1
Timer intervals :
  Hello 10, Hello Jitter 10 Dead 40, Retransmit 5
 Authentication: Enabled
  KeyRolloverTime(sec): Configured: 30 Current: 0
  KeyRolloverState: NotActive
  Outbound: SPI:121212, ESP, SHA1
   Key:12345678901234567890123456789012345678901234567890
  Inbound: SPI:121212, ESP, SHA1
   Key:12345678901234567890123456789012345678901234567890
 DR:10.2.2.2 BDR:10.1.1.1 Number of I/F scoped LSAs is 2
 DRElection: 1 times, DelayedLSAck: 83 times
 Neighbor Count = 1, Adjacent Neighbor Count= 1
  Neighbor:
   10.2.2.2 (DR)
 Statistics of interface eth 1/8:
  Type     tx    rx     tx-byte    rx-byte
  Unknown  0     0      0          0
  Hello    1415  1408   56592      56320
  DbDesc   3     3      804        804
  LSReq    1     1      28         28
  LSUpdate 193   121    15616      9720
  LSAck    85    109    4840       4924
  OSPF messages dropped,no authentication: 0
eth 2/2 is up, type POINT-TO-POINT
 IPv6 Address:
  2001:db8:22:22::1/64
  2001:db8:22:22::/64
  2001:db8:202:202::1/64
  2001:db8:202:202::/64
 Instance ID 0, Router ID 10.1.1.1
 Area ID 100, Cost 1
 State P2P, Transmit Delay 1 sec, Priority 1
 Timer intervals:
  Hello 10, Hello Jitter 10 Dead 40, Retransmit 5
 Authentication: Enabled
  KeyRolloverTime(sec): Configured: 30 Current: 0
  KeyRolloverState: NotActive
  Outbound: SPI:11022, ESP, SHA1
   Key:12345678901234567890123456789012345678901234567890
  Inbound: SPI:11022, ESP, SHA1
   Key:12345678901234567890123456789012345678901234567890
  DR:0.0.0.0 BDR:0.0.0.0 Number of I/F scoped LSAs is 2
  ....................................
```

You can display detailed OSPFv3 information about a specific interface using the following command at any level of the CLI.

Syntax: show ipv6 ospf interface [**ethernet** *slot/port* | **loopback** *number* | **tunnel** *number* | **ve** *number*]

The **ethernet**, **loopback** , **tunnel**, and **ve** parameter specify the interface for which to display information. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a loopback, tunnel, or VE interface, also specify the number associated with the interface.

TABLE 122 show ipv6 ospf interface output descriptions

This field	Displays
Interface status	The status of the interface. Possible status includes the following: • Up. • Down.

TABLE 122 show ipv6 ospf interface output descriptions (Continued)

This field	Displays
Type	The type of OSPFv3 circuit running on the interface. Possible types include the following: • BROADCAST • POINT TO POINT UNKNOWN • POINT TO POINT
IPv6 Address	The IPv6 address assigned to the interface.
Instance ID	An identifier for an instance of OSPFv3.
Router ID	The IPv4 address of the device. By default, the router ID is the IPv4 address configured on the lowest numbered loopback interface. If the device does not have a loopback interface, the default router ID is the lowest numbered IPv4 address configured on the device.
Area ID	The IPv4 address or numerical value of the area in which the interface belongs.
Cost	The overhead required to send a packet through the interface.
default	Shows whether or not the default passive state is set.
State	The state of the interface. Possible states include the following: • DR - The interface is functioning as the Designated Router for OSPFv3. • BDR - The interface is functioning as the Backup Designated Router for OSPFv3. • Loopback - The interface is functioning as a loopback interface. • P2P - The interface is functioning as a point-to-point interface. • Passive - The interface is up but it does not take part in forming an adjacency. • Waiting - The interface is trying to determine the identity of the BDR for the network. • None - The interface does not take part in the OSPF interface state machine. • Down - The interface is unusable. No protocol traffic can be sent or received on such a interface. • DR other - The interface is a broadcast or NBMA network on which another router is selected to be the DR. • Active - The interface sends or receives all the OSPFv3 control packets, and forms the adjacency.
Transmit delay	The amount of time, in seconds, it takes to transmit Link State Updates packets on the interface.
Priority	The priority used when selecting the DR and the BDR. If the priority is 0, the interface does not participate in the DR and BDR election.
Timer intervals	The interval, in seconds, of the hello-interval, dead-interval, and retransmit-interval timers.
DR	The router ID (IPv4 address) of the DR.
BDR	The router ID (IPv4 address) of the BDR.

TABLE 122 show ipv6 ospf interface output descriptions (Continued)

This field	Displays
Number of I/F scoped LSAs	The number of interface LSAs scoped for a specified area, AS, or link.
DR Election	The number of times the DR election occurred.
Delayed LSA Ack	The number of the times the interface sent a delayed LSA acknowledgement.
Neighbor Count	The number of neighbors to which the interface is connected.
Adjacent Neighbor Count	The number of neighbors with which the interface has formed an active adjacency.
Neighbor	The router ID (IPv4 address) of the neighbor. This field also identifies the neighbor as a DR or BDR, if appropriate.
Interface statistics	The following statistics are provided for the interface: • Unknown - The number of Unknown packets transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received Unknown packets. • Hello - The number of Hello packets transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received Hello packets. • DbDesc - The number of Database Description packets transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received Database Description packets. • LSReq - The number of link-state requests transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received link-state requests. • LSUpdate - The number of link-state updates transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received link-state requests. • LSAck - The number of link-state acknowledgements transmitted and received by the interface. Also, the total number of bytes associated with transmitted and received link-state acknowledgements.

Displaying OSPFv3 memory usage

To display information about OSPFv3 memory usage, enter the following command at any level of the CLI.

```
device# show ipv6 ospf memory
Total Static Memory Allocated : 5829 bytes
Total Dynamic Memory Allocated : 0 bytes
 Memory Type               Size     Allocated  Max-alloc   Alloc-Fails
 MTYPE_OSPF6_TOP           0          0          0          0
 MTYPE_OSPF6_LSA_HDR       0          0          0          0
 MTYPE_OSPF6_RMAP_COMPILED 0          0          0          0
 MTYPE_OSPF6_OTHER         0          0          0          0
 MTYPE_THREAD_MASTER       0          0          0          0
 MTYPE_OSPF6_AREA          0          0          0          0
 MTYPE_OSPF6_AREA_RANGE    0          0          0          0
 MTYPE_OSPF6_SUMMARY_ADDRE 0          0          0          0
 MTYPE_OSPF6_IF            0          0          0          0
 MTYPE_OSPF6_NEIGHBOR      0          0          0          0
 MTYPE_OSPF6_ROUTE_NODE    0          0          0          0
 MTYPE_OSPF6_ROUTE_INFO    0          0          0          0
```

```
MTYPE_OSPF6_PREFIX         0          0          0          0
MTYPE_OSPF6_LSA            0          0          0          0
MTYPE_OSPF6_VERTEX         0          0          0          0
MTYPE_OSPF6_SPFTREE        0          0          0          0
MTYPE_OSPF6_NEXTHOP        0          0          0          0
MTYPE_OSPF6_EXTERNAL_INFO  0          0          0          0
MTYPE_THREAD               0          0          0          0
```

Syntax: show ipv6 ospf memory

TABLE 123 show ipv6 ospf memory output descriptions

This field	Displays
Total Static Memory Allocated	A summary of the amount of static memory allocated, in bytes, to OSPFv3.
Total Dynamic Memory Allocated	A summary of the amount of dynamic memory allocated, in bytes, to OSPFv3.
Memory Type	The type of memory used by OSPFv3. (This information is for use by Brocade technical support in case of a problem.)
Size	The size of a memory type.
Allocated	The amount of memory currently allocated to a memory type.
Max-alloc	The maximum amount of memory that was allocated to a memory type.
Alloc-Fails	The number of times an attempt to allocate memory to a memory type failed.

Displaying OSPFv3 neighbor information

You can display a summary of OSPFv3 neighbor information for the device or detailed information about a specified neighbor.

To display a summary of OSPFv3 neighbor information for the device, enter the following command at any CLI level.

```
device# show ipv6 ospf neighbor
RouterID     Pri State    DR              BDR         Interface [State]
10.1.1.1     1 Full       10.223.223.223  10.1.1.1    ethe 3/2  [DR]
```

Syntax: show ipv6 ospf neighbor [**router-id** *ipv4-address*]

The **router-id** *ipv4-address* parameter displays only the neighbor entries for the specified router.

TABLE 124 show ipv6 ospf neighbor output descriptions

Field	Description
Router ID	The IPv4 address of the neighbor. By default, the router ID is the IPv4 address configured on the lowest numbered loopback interface. If the device does not have a loopback interface, the default router ID is the lowest numbered IPv4 address configured on the device.
Pri	The OSPFv3 priority of the neighbor. The priority is used during election of the DR and BDR.

TABLE 124 show ipv6 ospf neighbor output descriptions (Continued)

Field	Description
State	The state between the device and the neighbor. The state can be one of the following: • Down • Attempt • Init • 2-Way • ExStart • Exchange • Loading • Full
DR	The router ID (IPv4 address) of the DR.
BDR	The router ID (IPv4 address) of the BDR.
Interface [State]	The interface through which the router is connected to the neighbor. The state of the interface can be one of the following: • DR - The interface is functioning as the Designated Router for OSPFv3. • BDR - The interface is functioning as the Backup Designated Router for OSPFv3. • Loopback - The interface is functioning as a loopback interface. • P2P - The interface is functioning as a point-to-point interface. • Passive - The interface is up but it does not take part in forming an adjacency. • Waiting - The interface is trying to determine the identity of the BDR for the network. • None - The interface does not take part in the OSPF interface state machine. • Down - The interface is unusable. No protocol traffic can be sent or received on such an interface. • DR other - The interface is a broadcast or NBMA network on which another router is selected to be the DR.

For example, to display detailed information about a neighbor with the router ID of 10.1.1.1, enter the **show ipv6 ospf neighbor router-id** command at any CLI level.

```
device# show ipv6 ospf neighbor router-id 10.3.3.3
RouterID     Pri State    DR          BDR         Interface [State]
10.3.3.3     1 Full       10.3.3.3    10.1.1.1    ve 10     [BDR]
   DbDesc bit for this neighbor: --s
   Nbr Ifindex of this router: 1
   Nbr DRDecision: DR 10.3.3.3, BDR 10.1.1.1
   Last received DbDesc: opt:xxx ifmtu:0 bit:--s seqnum:0
   Number of LSAs in DbDesc retransmitting: 0
   Number of LSAs in SummaryList: 0
   Number of LSAs in RequestList: 0
   Number of LSAs in RetransList: 0
   SeqnumMismatch 0 times, BadLSReq 0 times
   OnewayReceived 0 times, InactivityTimer 0 times
   DbDescRetrans 0 times, LSReqRetrans 0 times
   LSUpdateRetrans 1 times
   LSAReceived 12 times, LSUpdateReceived 6 times
```

TABLE 125 show ipv6 ospf neighbor router-id output descriptions

Field	Description
Router ID	The IPv4 address of the neighbor. By default, the router ID is the IPv4 address configured on the lowest numbered loopback interface. If the device does not have a loopback interface, the default router ID is the lowest numbered IPv4 address configured on the device.
Pri	The OSPFv3 priority of the neighbor. The priority is used during election of the DR and BDR.
State	The state between the device and the neighbor. The state can be one of the following: • Down • Attempt • Init • 2-Way • ExStart • Exchange • Loading • Full
DR	The router ID (IPv4 address) of the DR.
BDR	The router ID (IPv4 address) of the BDR.
Interface [State]	The interface through which the router is connected to the neighbor. The state of the interface can be one of the following: • DR - The interface is functioning as the Designated Router for OSPFv3. • BDR - The interface is functioning as the Backup Designated Router for OSPFv3. • Loopback - The interface is functioning as a loopback interface. • P2P - The interface is functioning as a point-to-point interface. • Passive - The interface is up but it does not take part in forming an adjacency. • Waiting - The interface is trying to determine the identity of the BDR for the network. • None - The interface does not take part in the OSPF interface state machine. • Down - The interface is unusable. No protocol traffic can be sent or received on such a interface. • DR other - The interface is a broadcast or NBMA network on which another router is selected to be the DR.
DbDesc bit	The Database Description packet, which includes 3 bits of information: • The first bit can be "i" or "-". "i" indicates the inet bit is set. "-" indicates the inet bit is not set. • The second bit can be "m" or "-". "m" indicates the more bit is set. "-" indicates the more bit is not set. • The third bit can be "m" or "s". An "m" indicates the master. An "s" indicates standby.

TABLE 125 show ipv6 ospf neighbor router-id output descriptions (Continued)

Field	Description
Index	The ID of the LSA from which the neighbor learned of the router.
DR Decision	The router ID (IPv4 address) of the neighbor's elected DR and BDR.
Last Received Db Desc	The content of the last database description received from the specified neighbor.
Number of LSAs in Db Desc retransmitting	The number of LSAs that need to be retransmitted to the specified neighbor.
Number of LSAs in Summary List	The number of LSAs in the neighbor's summary list.
Number of LSAs in Request List	The number of LSAs in the neighbor's request list.
Number of LSAs in Retransmit List	The number of LSAs in the neighbor's retransmit list.
Seqnum Mismatch	The number of times sequence number mismatches occurred.
BadLSReq	The number of times the neighbor received a bad link-state request from the device.
One way received	The number of times a hello packet, which does not mention the router, is received from the neighbor. This omission in the hello packet indicates that the communication with the neighbor is not bidirectional.
Inactivity Timer	The number of times that the neighbor's inactivity timer expired.
Db Desc Retransmission	The number of times sequence number mismatches occurred.
LSReqRetrans	The number of times the neighbor retransmitted link-state requests to the device.
LSUpdateRetrans	The number of times the neighbor retransmitted link-state updates to the device.
LSA Received	The number of times the neighbor received LSAs from the device.
LS Update Received	The number of times the neighbor received link-state updates from the device.

Displaying routes redistributed into OSPFv3

You can display all IPv6 routes or a specified IPv6 route that the device has redistributed into OSPFv3.

To display all IPv6 routes that the device has redistributed into OSPFv3, enter the following command at any level of the CLI.

```
device# show ipv6 ospf redistribute route
Id     Prefix                          Protocol  Metric Type  Metric
snIpAsPathAccessListStringRegExpression
```

```
1       2001:db8::/32                   Static    Type-2       1
2       2001:db8:1234::/48              Static    Type-2       0
```

Syntax: show ipv6 ospf redistribute route [*ipv6-prefix*]

The *ipv6-prefix* parameter specifies an IPv6 network prefix. (You do not need to specify the length of the prefix.)

For example, to display redistribution information for the prefix 2001:db8::, enter the following command at any level of the CLI.

```
device# show ipv6 ospf redistribute route 2001:db8::
Id      Prefix                          Protocol  Metric Type  Metric
1       2001:db8::/32                   Static    Type-2       1
```

TABLE 126 show ipv6 ospf redistribute route output descriptions

This field	Displays
ID	An ID for the redistributed route.
Prefix	The IPv6 routes redistributed into OSPFv3.
Protocol	The protocol from which the route is redistributed into OSPFv3. Redistributed protocols can be the following: • BGP - BGP4+. • RIP - RIPng. • IS-IS – IPv6 IS-IS. • Static - IPv6 static route table. • Connected - A directly connected network.
Metric Type	The metric type used for routes redistributed into OSPFv3. The metric type can be the following: • Type-1 - Specifies a small metric (2 bytes). • Type-2 - Specifies a big metric (3 bytes).
Metric	The value of the default redistribution metric, which is the OSPF cost of redistributing the route into OSPFv3.

Displaying OSPFv3 route information

You can display the entire OSPFv3 route table for the Brocade device or only the route entries for a specified destination.

To display the entire OSPFv3 route table for the device, enter the following command at any level of the CLI.

```
device#show ipv6 ospf route
Current Route count: 4
    Intra: 4 Inter: 0 External: 0 (Type1 0/Type2 0)
    Equal-cost multi-path: 0
    OSPF Type: IA- Intra, OA - Inter, E1 - External Type1, E2 - External Type2
   Destination               Cost     E2Cost    Tag         Flags    Dis
IA 2001:db8:200:1::1/128     0        0         0           00000003 110
   Next_Hop_Router           Outgoing_Interface Adv_Router
   ::                        loopback 1         10.1.2.1
   Destination               Cost     E2Cost    Tag         Flags    Dis
IA 2001:db8:300:1::1/128     0        0         0           00000003 110
   Next_Hop_Router           Outgoing_Interface Adv_Router
```

```
   ::                          loopback 2         10.1.2.1
   Destination                 Cost     E2Cost    Tag        Flags    Dis
IA 2001:db8:400:1::1/128       0        0         0          00000003 110
   Next_Hop_Router             Outgoing_Interface Adv_Router
   ::                          loopback 1         10.1.2.1
   Destination                 Cost      E2Cost   Tag        Flags    Dis
IA 2001:db8:500:1::1/128       1         0        0          00000003 110
   Next_Hop_Router             Outgoing_Interface Adv_Router
   ::                          loopback 2         10.1.2.1
```

Syntax: show ipv6 ospf routes [*ipv6-prefix*]

The *ipv6-prefix* parameter specifies a destination IPv6 prefix. (You do not need to specify the length of the prefix.) If you use this parameter, only the route entries for this destination are shown.

For example, to display route information for the destination prefix 2000:4::, enter the following command at any level of the CLI.

```
device# show ipv6 ospf routes 2000::
Destination                 Cost      E2Cost    Tag       Flags    Dis
IA 2000::/64                1         0         0         00000003 110
   Next_Hop_Router           Outgoing_Interface Adv_Router
   ::                          eth 1/1            10.1.1.1
```

TABLE 127 OSPFv3 route information

This field	Displays
Current Route Count (Displays with the entire OSPFv3 route table only)	The number of route entries currently in the OSPFv3 route table.
Intra/Inter/External (Type1/Type2) (Displays with the entire OSPFv3 route table only)	The breakdown of the current route entries into the following route types: • Inter - The number of routes that pass into another area. • Intra - The number of routes that are within the local area. • External1 - The number of type 1 external routes. • External2 - The number of type 2 external routes.
Equal-cost multi-path (Displays with the entire OSPFv3 route table only)	The number of equal-cost routes to the same destination in the OSPFv3 route table. If load sharing is enabled, the device equally distributes traffic among the routes.
Destination	The IPv6 prefixes of destination networks to which the device can forward IPv6 packets. "*IA" indicates the next router is an intra-area router.
Cost	The type 1 cost of this route.
E2 Cost	The type 2 cost of this route.
Tag	The route tag for this route.
Flags	Flags associated with this route.
Dis	Administrative Distance for this route.
Next-Hop Router	The IPv6 address of the next router a packet must traverse to reach a destination.

TABLE 127 OSPFv3 route information (Continued)

This field	Displays
Outgoing Interface	The router interface through which a packet must traverse to reach the next-hop router.
Adv_Router	The IP address of the advertising router.

Displaying OSPFv3 SPF information

You can display the following OSPFv3 SPF information:

- SPF node information for a specified area.
- SPF table for a specified area.
- SPF tree for a specified area.

For example, to display information about SPF nodes in area 0, enter the **show ipv6 ospf spf node area** command at any level of the CLI.

```
device# show ipv6 ospf spf node area 0
SPF node for Area 0
SPF node 10.223.223.223,  cost: 0,  hops: 0
 nexthops to node:
 parent nodes:
 child nodes: 10.223.223.223:88
SPF node 10.223.223.223:88,  cost: 1,  hops: 1
 nexthops to node:    :: ethe 3/2
 parent nodes: 10.223.223.223
 child nodes: 10.1.1.1:0
SPF node 10.1.1.1:0,  cost: 1,  hops: 2
 nexthops to node:    fe80::2e0:52ff:fe91:bb37 ethe 3/2
 parent nodes: 10.223.223.223:88
 child nodes:
```

Syntax: show ipv6 ospf spf node area [*area-id*]

The **node** keyword displays SPF node information.

The **area** *area-id* parameter specifies a particular area. You can specify the *area-id* in the following formats:

- As an IPv4 address; for example, 192.168.1.1.
- As a numerical value from 0 through 2,147,483,647.

TABLE 128 show ipv6 ospf spf node area output descriptions

This field	Displays
SPF node	Each SPF node is identified by its device ID (IPv4 address). If the node is a child node, it is additionally identified by an interface on which the node can be reached appended to the router ID in the format *router-id* :*interface-id* .
Cost	The cost of traversing the SPF node to reach the destination.
Hops	The number of hops needed to reach the parent SPF node.
Next Hops to Node	The IPv6 address of the next hop-router or the router interface through which to access the next-hop router.

TABLE 128 show ipv6 ospf spf node area output descriptions (Continued)

This field	Displays
Parent Nodes	The SPF node's parent nodes. A parent node is an SPF node at the highest level of the SPF tree, which is identified by its router ID.
Child Nodes	The SPF node's child nodes. A child node is an SPF node at a lower level of the SPF tree, which is identified by its router ID and interface on which the node can be reached.

For example, to display the SPF table for area 0, enter the following command at any level of the CLI.

```
device# show ipv6 ospf spf table area 0
SPF table for Area 0
 Destination         Bits Options  Cost Nexthop                 Interface
R 10.1.1.1           ---- V6E---R-   1  fe80::2e0:52ff:fe91:bb37 ethe 3/2
N 10.223.223.223[88] ---- V6E---R-   1  ::                       ethe 3/2
```

Syntax: show ipv6 opsf spf table area *area-id*

The **table** parameter displays the SPF table.

The **area** *area-id* parameter specifies a particular area. You can specify the *area-id* in the following formats:

- As an IPv4 address, for example, 192.168.1.1.
- As a numerical value from 0 through 2,147,483,647.

TABLE 129 show ipv6 ospf spf table area output descriptions

This field	Displays
Destination	The destination of a route, which is identified by the following: • "R", which indicates the destination is a router. "N", which indicates the destination is a network. • An SPF node's device ID (IPv4 address). If the node is a child node, it is additionally identified by an interface on which the node can be reached appended to the router ID in the format *router-id* :*interface-id* .
Bits	A bit that indicates the capability of the device. The bit can be set to one of the following: • B - The device is an area border router. • E - The device is an AS boundary router. • V - The device is a virtual link endpoint. • W - The device is a wildcard multicast receiver.
Options	A 24-bit field that enables IPv6 OSPF routers to support the optional capabilities. When set, the following bits indicate the following: V6 - The router should be included in IPv6 routing calculations. E - The router floods AS-external-LSAs as described in RFC 2740. MC - The router forwards multicast packets as described in RFC 1586. N - The router handles type 7 LSAs as described in RFC 1584. R - The originator is an active router. DC -The router handles demand circuits.

TABLE 129 show ipv6 ospf spf table area output descriptions (Continued)

This field	Displays
Cost	The cost of traversing the SPF node to reach the destination.
Next hop	The IPv6 address of the next hop-router.
Interface	The router interface through which to access the next-hop router.

For example, to display the SPF tree for area 0, enter the following command at any level of the CLI.

```
device# show ipv6 ospf spf tree area 0
   SPF tree for Area 0
 +- 10.223.223.223 cost 0
     +- 10.223.223.223:88 cost 1
         +- 10.1.1.1:0 cost 1
```

Syntax: show ipv6 opsf spf tree area *area-id*

The **tree** keyword displays the SPF table.

The **area** *area-id* parameter specifies a particular area. You can specify the *area-id* in the following formats:

- As an IPv4 address; for example, 192.168.1.1.
- As a numerical value from 0 through 2,147,483,647.

In this sample output, consider the SPF node with the router ID 10.223.223.223 to be the top (root) of the tree and the local router. Consider all other layers of the tree (10.223.223.223:88 and 10.1.1.1:0) to be destinations in the network. Therefore, traffic destined from router 10.223.223.223 to router 10.1.1.1:0 must first traverse router 10.223.223.223:88.

Displaying OSPFv3 GR Helper mode information

Run the **show ipv6 ospf** command to display information about the graceful restart helper mode

```
device# (config-ospf6-router)#show ipv6 ospf
OSPFv3 Process number 0 with Router ID 0x10010101(10.1.1.1)
Running 0 days 0 hours 18 minutes 21 seconds
Number of AS scoped LSAs is 0
Sum of AS scoped LSAs Checksum is 00000000
External LSA Limit is 250000
Database Overflow Interval is 10
Database Overflow State is NOT OVERFLOWED
Nonstop-routing is ENABLED
Route calculation executed 0 times
Pending outgoing LSA count 0
Authentication key rollover interval 300 seconds
Number of areas in this router is 1
High Priority Message Queue Full count: 0
BFD is disabled
Graceful restart helper is enabled, strict lsa checking enabled
```

Displaying OSPFv3 NSR information

Run the **show ipv6 ospf** command to display information about the NSR support.

```
device# (config-ospf6-router)#show ipv6 ospf
OSPFv3 Process number 0 with Router ID 0x10010101(10.1.1.1)
    Running 0 days 0 hours 18 minutes 21 seconds
```

```
    Number of AS scoped LSAs is 0
    Sum of AS scoped LSAs Checksum is 00000000
    External LSA Limit is 250000
    Database Overflow Interval is 10
    Database Overflow State is NOT OVERFLOWED
    Nonstop-routing is ENABLED
    Route calculation executed 0 times
    Pending outgoing LSA count 0
    Authentication key rollover interval 300 seconds
    Number of areas in this router is 1
    High Priority Message Queue Full count: 0
    BFD is disabled
    Graceful restart helper is enabled, strict lsa checking enabled
```

Displaying OSPFv3 max-metric router LSA information

Run the **show ipv6 ospf** command to display information about the NSR support.

```
device# (config-ospf6-router)#show ipv6 ospf
 OSPFv3 Process number 0 with Router ID 0x01010101(1.1.1.1)
 Running 0 days 0 hours 11 minutes 56 seconds
 Number of AS scoped LSAs is 0
 Sum of AS scoped LSAs Checksum is 00000000
 External LSA Limit is 250000
 Database Overflow Interval is 10
 Database Overflow State is NOT OVERFLOWED
 Route calculation executed 1 times
 Pending outgoing LSA count 0
 Authentication key rollover interval 300 seconds
 Number of areas in this router is 1
 High Priority Message Queue Full count: 0
 BFD is disabled
 Graceful restart helper is disabled
 Originate LSAs with maximum metric on startup, rem time 58 sec
  Include stub
  Additional LSAs originated with maximum metric:
   LSA Type Metric Value
   Summary 16711680
   External 16711680
```

Displaying IPv6 OSPF virtual link information

To display OSPFv3 virtual link information on a Brocade device, enter the **show ipv6 ospf virtual-link** command at any level of the CLI.

```
device# show ipv6 ospf virtual-link
Transit Area ID  Router ID        Interface Address             State
 1       1        10.1.1.1         201:db8::2                    P2P
```

Syntax: show ipv6 opsf virtual-link

TABLE 130 show ipv6 ospf virtual-link output descriptions

This field	Displays
Index	An index number associated with the virtual link.
Transit Area ID	The ID of the shared area of two ABRs that serves as a connection point between the two routers.

TABLE 130 show ipv6 ospf virtual-link output descriptions (Continued)

This field	Displays
Router ID	Router ID of the router at the other end of the virtual link (virtual neighbor).
Interface Address	The local address used to communicate with the virtual neighbor.
State	The state of the virtual link. Possible states include the following: • P2P - The link is functioning as a point-to-point interface. • DOWN - The link is down.

Displaying OSPFv3 virtual neighbor information

To display OSPFv3 virtual neighbor information for the Brocade device, enter the following command at the enabled level of the CLI.

```
device#show ipv6 ospf virtual-neighbor
Index Router ID        Address                       State     Interface
1     10.14.14.14      2001:db8:44:44::4             Full      eth 1/8
                 Option: 00-00-00    QCount: 0    Timer: 408
2     10.14.14.14      2001:db8:44:44::4             Full      tunnel 256
                 Option: 00-00-00    QCount: 0    Timer: 43
```

Syntax: show ipv6 ospf virtual-neighbor [**brief**]

The **brief** option results in an output that omits the Option, QCount, and Timer fields.

TABLE 131 show ipv6 ospf virtual-neighbor output descriptions

This field	Displays
Index	An index number associated with the virtual neighbor.
Router ID	IPv4 address of the virtual neighbor.
Address	The IPv6 address to be used for communication with the virtual neighbor.
State	The state between the device and the virtual neighbor. The state can be one of the following: • Down • Attempt • Init • 2-Way • ExStart • Exchange • Loading • Full
Interface	The IPv6 address of the virtual neighbor.
Option	The bits set in the virtual-link hello or database descriptors.

TABLE 131 show ipv6 ospf virtual-neighbor output descriptions (Continued)

This field	Displays
QCount	The number of packets that are in the queue and ready for transmission. If the system is stable, this number should always be 0.
Timer	A timer that counts down until a hello packet should arrive. If "timers" elapses and a hello packet has not arrived, the VL neighbor is declared to be down.

IPsec examples

This section contains examples of IPsec configuration and the output from the IPsec-specific **show** commands. In addition, IPsec-related information appears in general **show** command output for interfaces and areas.

The **show** commands that are specific to IPsec are:

- **show ipsec sa**
- **show ipsec policy**
- **show ipsec statistics**

The other **show** commands with IPsec-related information are:

- **show ipv6 ospf area**
- **show ipv6 ospf interface**
- **show ipv6 ospf vrf**

Showing IPsec security association information

The **show ipsec sa** command displays the IPSec security association databases, as follows.

```
device#show ipsec sa
IPSEC Security Association Database(Entries:8)
SPDID(vrf:if) Dir Encap SPI        Destination          AuthAlg  EncryptAlg
1:ALL         in  ESP   512        2001:db8:1::1        sha1     Null
1:e1/1        out ESP   302        ::                   sha1     Null
1:e1/1        in  ESP   302        FE80::               sha1     Null
1:e1/1        out ESP   512        2001:db8:1::2        sha1     Null
2:ALL         in  ESP   512        2001:db8:1::1        sha1     Null
2:e1/2        out ESP   302        ::                   sha1     Null
2:e1/2        in  ESP   302        FE80::               sha1     Null
2:e1/2        out ESP   512        2001:db8:1::2        sha1     Null
```

Syntax: show ipsec sa

Showing IPsec policy

The **show ipsec policy** command displays the database for the IPsec security policies. The fields for this **show** command output appear in the screen output example that follows. However, you should understand the layout and column headings for the display before trying to interpret the information in the example screen.

Each policy entry consists of two categories of information:

- The policy information
- The SA used by the policy

The policy information line in the screen begins with the heading Ptype and also has the headings Dir, Proto, Source (Prefix:TCP.UDP Port), and Destination (Prefix:TCP/UDPPort). The SA line contains the SPDID, direction, encapsulation (always ESP in the current release), the user-specified SPI.

```
device#show ipsec policy
IPSEC Security Policy Database(Entries:8)
PType  Dir Proto Source(Prefix:TCP/UDP Port)
               Destination(Prefix:TCP/UDPPort)
  SA: SPDID(vrf:if) Dir Encap SPI        Destination
use    in  OSPF  FE80::/10:any
               ::/0:any
  SA: 2:e1/2       in  ESP   302        FE80::
use    out OSPF  FE80::/10:any
               ::/0:any
  SA: 2:e1/2       out ESP   302        ::
use    in  OSPF  FE80::/10:any
               ::/0:any
  SA: 1:e1/1       in ESP    302        FE80::
use    out OSPF  FE80::/10:any
               ::/0:any
  SA: 1:e1/1       out ESP   302        ::
use    in  OSPF  2001:db8:1:1::1/128:any
               2001:db8:1:1::2/128:any
  SA: 1:ALL        in  ESP   512        2001:db8:1:1::2
use    out OSPF  2001:db8:1:1::2/128:any
               2001:db8:1:1::1/128:any
  SA: 1:e1/1       out ESP   512        2001:db8:1:1::1
```

Syntax: show ipsec policy

TABLE 132 show ipsec policy output descriptions

This field	Displays
PType	This field contains the policy type. Of the existing policy types, only the "use" policy type is supported, so each entry can have only "use."
Dir	The direction of traffic flow to which the IPsec policy is applied. Each direction has its own entry.
Proto	The only possible routing protocol for the security policy in the current release is OSPFv3.
Source	The source address consists of the IPv6 prefix and the TCP or UDP port identifier.
Destination	The destination address consists of the IPv6 prefix. Certain logical elements have a bearing on the meaning of the destination address and its format, as follows: For IPsec on an interface or area, the destination address is shown as a prefix of 0xFE80 (link local). The solitary "::" (no prefix) indicates a "do not-care" situation because the connection is multicast. In this case, the security policy is enforced without regard for the destination address. For a virtual link (SPDID = 0), the address is required.

TABLE 133 SA used by the policy

This field	Displays
SA	This heading points at the SA-related headings for information used by the security policy. Thereafter, on each line of this part of the IPsec entry (which alternates with lines of policy information, "SA:" points at the fields under those SA-related headings. The remainder of this table describes each of the SA-related items.

TABLE 133 SA used by the policy (Continued)

This field	Displays
SPDID	The security policy database identifier (SPDID) consists of two parts; the first part is an VRF id and the second part is an interface ID. The SPDID 0/ALL is a global database for the default VRF that applies to all interfaces.
Dir	The Dir field is either ‘in" for inbound or "out" for outbound.
Encap	The type of encapsulation in the current release is ESP.
SPI	Security parameter index.
Destination	The IPv6 address of the destination endpoint. From the standpoint of the near interface and the area, the destination is not relevant and therefore appears as ::/0:any. For a virtual link, both the inbound and outbound destination addresses are relevant.

Showing IPsec statistics

The **show ipsec statistics** command displays the error and other counters for IPsec, as this example shows.

```
device#show ipsec statistics
                      IPSecurity Statistics
secEspCurrentInboundSAs 1          ipsecEspTotalInboundSAs:  2
secEspCurrentOutboundSA 1          ipsecEspTotalOutboundSAs: 2
                   IPSecurity Packet Statistics
secEspTotalInPkts:      19         ipsecEspTotalInPktsDrop:  0
secEspTotalOutPkts:     83
                   IPSecurity Error Statistics
secAuthenticationErrors 0
secReplayErrors:        0          ipsecPolicyErrors:        13
secOtherReceiveErrors:  0          ipsecSendErrors:          0
secAuthenticationErrors 0
secReplayErrors:        0          ipsecPolicyErrors:        13
secOtherReceiveErrors:  0          ipsecSendErrors:          0
secUnknownSpiErrors:    0
```

Syntax: show ipsec statistics

This command takes no parameters.

Displaying IPsec configuration for an area

The **show ipv6 ospf area** command includes information about IPsec for one area or all areas. In the following example, the IPsec information is in bold. IPsec is enabled in the first area (area 0) in this example but not in area 3. Note that in area 3, the IPsec key was specified as not encrypted.

```
device(config-ospf6-router)#show ipv6 ospf area
  Area 0:
  Authentication: Not Configured
  Active interface(s)attached to this area: eth 1/1
  Inactive interface(s)attached to this area: None
  Number of Area scoped LSAs is 2
  Sum of Area LSAs Checksum is 00021139
  Statistics of Area 1:
   SPF algorithm executed 1 times
   SPF last updated: 111 sec ago
   Current SPF node count: 1
    Router: 1 Network: 0
```

```
    Maximum of Hop count to nodes: 0
Area 1: Area is NSSA, no redistribution
 Authentication: Not Configured
 NSSA translator status: Enabled, NSSA translator interval: 40 sec
 Active interface(s)attached to this area: eth 1/1
 Inactive interface(s)attached to this area: None
 Number of Area scoped LSAs is 2
 Sum of Area LSAs Checksum is 00021139
 Statistics of Area 1:
  SPF algorithm executed 1 times
  SPF last updated: 111 sec ago
  Current SPF node count: 1
   Router: 1 Network: 0
   Maximum of Hop count to nodes: 0
Area 2: Area is stub
 Authentication: Not Configured
 Active interface(s)attached to this area: None
 Inactive interface(s)attached to this area: None
 Number of Area scoped LSAs is 0
 Sum of Area LSAs Checksum is 00000000
 Statistics of Area 2:
  SPF algorithm executed 0 times
  SPF last updated: 89562 sec ago
  Current SPF node count: 0
   Router: 0 Network: 0
   Maximum of Hop count to nodes: 0
Area 3: Area is NSSA, no summary
 Authentication: Not Configured
 NSSA translator status: Elected, NSSA translator interval: 60 sec
 Active interface(s)attached to this area: eth 1/4
 Inactive interface(s)attached to this area: None
 Number of Area scoped LSAs is 1
 Sum of Area LSAs Checksum is 00023299
 Statistics of Area 1:
  SPF algorithm executed 1 times
  SPF last updated: 111 sec ago
  Current SPF node count: 1
   Router: 1 Network: 0
   Maximum of Hop count to nodes: 0
```

Syntax: show ipv6 ospf area [*area-id*]

The *area-id* parameter restricts the display to the specified OSPF area. You can specify the *area-id* parameter in the following formats:

- An IPv6 address, for example, 2001:db8::10
- A numerical value in the range 0 through 2,147,483,647

TABLE 134 show ipv6 ospf area output descriptions

This field	Displays
Authentication	This field shows whether or not authentication is configured. If this field says "Not Configured," the IPsec-related fields (bold in example screen output) are not displayed at all.
KeyRolloverTime	The number of seconds between each initiation of a key rollover. This field shows the configured and current times.
KeyRolloverState	Can be: Not active: key rollover is not active. Active phase 1: rollover is in its first interval. Active phase 2: rollover is in its second interval.
Current	Shows current SPI, authentication algorithm (currently ESP only), encryption algorithm (currently SHA1 only), and the current key.

TABLE 134 show ipv6 ospf area output descriptions (Continued)

This field	Displays
New	Shows new SPI (if changed), authentication algorithm (currently ESP only), encryption algorithm (currently SHA1 only), and the new key.
Old	Shows old SPI (if changed), authentication algorithm (currently ESP only), encryption algorithm (currently SHA1 only), and the old key.

Displaying IPsec for an interface

To see IPsec configuration for a particular interface or all interfaces, use the **show ipv6 ospf interface** command as in the following example. IPsec information appears in bold.

```
device#show ipv6 ospf interface
eth 1/3 is down, type BROADCAST
   Interface is disabled
eth 1/8 is up, type BROADCAST
  IPv6 Address:
      2001:db8:18:18:18::1/64
      2001:db8:18:18::/64
  Instance ID 255, Router ID 10.1.1.1
  Area ID 1, Cost 1
  State BDR, Transmit Delay 1 sec, Priority 1
  Timer intervals :
    Hello 10, Hello Jitter 10  Dead 40, Retransmit 5
  Authentication: Enabled
   KeyRolloverTime(sec): Configured: 30 Current: 0
   KeyRolloverState: NotActive
   Outbound: SPI:121212, ESP, SHA1
      Key:12345678901234567890123456789012345678901234567890
   Inbound: SPI:121212, ESP, SHA1
      Key:12345678901234567890123456789012345678901234567890
  DR:10.2.2.2 BDR:10.1.1.1  Number of I/F scoped LSAs is 2
  DRElection:     1 times, DelayedLSAck:    83 times
  Neighbor Count = 1,   Adjacent Neighbor Count= 1
     Neighbor:
      10.2.2.2 (DR)
    Statistics of interface eth 1/8:
      Type     tx         rx         tx-byte    rx-byte
      Unknown  0          0          0          0
      Hello    1415       1408       56592      56320
      DbDesc   3          3          804        804
      LSReq    1          1          28         28
      LSUpdate 193        121        15616      9720
      LSAck    85         109        4840       4924
      OSPF messages dropped,no authentication: 0
```

Syntax: show ipv6 ospf interface [**ethernet** *slot/port* | **loopback** *number* | **tunnel** *number* | **ve** *number*]

TABLE 135 show ipv6 ospf interface output descriptions

This field	Displays
Authentication	This field shows whether or not authentication is configured. If this field says "Not Configured," the IPsec-related fields (bold in example screen output) are not displayed at all.
KeyRolloverTime	The number of seconds between each initiation of a key rollover. This field shows the configured and current times.

TABLE 135 show ipv6 ospf interface output descriptions (Continued)

This field	Displays
KeyRolloverState	Can be: Not active: key rollover is not active. Active phase 1: rollover is in its first interval. Active phase 2: rollover is in its second interval.
Current	Shows current SPI, authentication algorithm (currently ESP only), encryption algorithm (currently SHA1 only), and the current key.
New (Inbound or Outbound)	Shows new SPI (if changed), authentication algorithm (currently ESP only), encryption algorithm (currently SHA1 only), and the new key.
Old (Inbound or Outbound)	Shows old SPI (if changed), authentication algorithm (currently ESP only), encryption algorithm (currently SHA1 only), and the old key.
OSPF messages dropped	Shows the number of packets dropped because the packets failed authentication (for any reason).

Displaying IPsec for a virtual link

To display IPsec for a virtual link, run the **show ipv6 ospf virtual-link brief** or **show ipv6 ospf virtual-link** command, as the following examples illustrate.

```
device#show ipv6 ospf virtual-link brief
Index Transit Area ID  Router ID        Interface Address             State
1     1                10.14.14.14      2001:db8::1:1:1::1            P2P
device#show ipv6 ospf virtual-link
Transit Area ID  Router ID        Interface Address             State
1                10.14.14.14      2001:db8:1:1:1::1             P2P
  Timer intervals(sec) :
    Hello 10, Hello Jitter 10,  Dead 40, Retransmit 5, TransmitDelay 1
  DelayedLSAck:     5 times
  Authentication: Configured
   KeyRolloverTime(sec): Configured: 10 Current: 0
   KeyRolloverState: NotActive
   Outbound: SPI:100004, ESP, SHA1
      Key:12345678901234567890123456789012345678901234567890
   Inbound: SPI:100004, ESP, SHA1
      Key:12345678901234567890123456789012345678901234567890
  Statistics:
      Type     tx         rx         tx-byte    rx-byte
      Unknown  0          0          0          0
      Hello    65         65         2600       2596
      DbDesc   4          4          2752       2992
      LSReq    1          1          232        64
      LSUpdate 11         5          1040       1112
      LSAck    5          8          560        448
      OSPF messages dropped,no authentication: 0
Neighbor: State: Full Address: 2001:db8:44:44::4 Interface: eth 2/2
```

Syntax: show ipv6 ospf virtual-link [**brief**]

The optional **brief** keyword limits the display to the Transit, Area ID, Router ID, Interface Address, and State fields for each link.

Changing a key

In this example, the key is changed. Note that the SPI value is changed from 300 to 310 to comply with the requirement that the SPI is changed when the key is changed.

Initial configuration command.

```
device(config-if-e10000-1/3)#ipv6 ospf auth ipsec spi 300 esp sha1
no-encrypt 1234567890098765543123456789 0aabbccddef
```

Command for changing the key.

```
device(config-if-e10000-1/3)#ipv6 ospf auth ipsec spi 310 esp sha1
no-encrypt 9898989890098765543212345678 90aabbccddef
```

Displaying IPv6 OSPF information for a VRF

To display IPv6 OSPF information for a VRF or all VRF interfaces, use the **show ipv6 ospf vrf** command as in the following example.

```
device#show ipv6 ospf vrf red
OSPF V3 Process number 0 with Router ID 0x10020202(10.2.2.2)
Running 0 days 0 hours 5 minutes 49 seconds
Number of AS scoped LSAs is 0
Sum of AS scoped LSAs Checksum is 00000000
External LSA Limit is 250000
Database Overflow Interval is 10
Database Overflow State is NOT OVERFLOWED
Route calculation executed 0 times
Pending outgoing LSA count 0
Authentication key rollover interval 30 seconds
Number of areas in this router is 4
Router is operating as ABR
Router is operating as ASBR, Redistribute: CONNECTED
High Priority Message Queue Full count: 0
BFD is disabled
```

Syntax: show ipv6 ospf vrf *vrf-name* [**area** *area-id* | **virtual-links**]

The *vrf-name* parameter specifies the VRF that you want the OSPF area information for.

The *area-id* parameter shows information for the specified area.

The *virtual-link* parameter displays the entry that corresponds to the IP address you enter.

Use the **show ipv6 ospf vrf** command to display the currently selected IPv6 global address for use by the Virtual Links in each transit area.

```
device#show ipv6 ospf vrf red area
Area 3:
Authentication: Not Configured
Interface attached to this area:
Number of Area scoped LSAs is 3
Sum of Area LSAs Checksum is 0001a6c4
Statistics of Area 3:
SPF algorithm executed 3 times
SPF last updated: 302 sec ago
Current SPF node count: 1
Router: 1 Network: 0
Maximum of Hop count to nodes: 0
Area 2:
Authentication: Not Configured
Interface attached to this area:
Number of Area scoped LSAs is 3
Sum of Area LSAs Checksum is 000192d6
Statistics of Area 2:
SPF algorithm executed 3 times
SPF last updated: 302 sec ago
Current SPF node count: 1
Router: 1 Network: 0
```

```
Maximum of Hop count to nodes: 0
Area 1:
Authentication: Not Configured
Interface attached to this area: eth 1/1
Number of Area scoped LSAs is 6
Sum of Area LSAs Checksum is 00046630
Statistics of Area 1:
SPF algorithm executed 3 times
SPF last updated: 302 sec ago
Current SPF node count: 3
Router: 2 Network: 1
Maximum of Hop count to nodes: 2
Global IPv6 Address used by Virtual Links in this area:10:1:1::2
Area 0.0.0.0 :
Authentication: Not Configured
Interface attached to this area: VLink 1
Number of Area scoped LSAs is 6
```

Syntax: show ipv6 ospf vrf *vrf-name* [**area** *area-id* | **virtual-links**]

Use the **show ipv6 ospf vrf** *vrf-name* **neighbor** command to display the currently selected neighbor for use by the Virtual Links in each transit area.

```
device#show ipv6 ospf vrf red neighbor
Total number of neighbors in all states: 1
Number of neighbors in state Full      : 1
Type     tx         rx         tx-byte    rx-byte
      Unknown  0          0          0          0
      Hello    32         32         1276       1280
      DbDesc   2          2          116        116
      LSReq    1          1          52         52
      LSUpdate 2          2          184        200
      LSAck    2          2          112        112
      OSPF messages dropped,no authentication: 0
  Neighbor: State: Full Address: 2001:db8:1::1 Interface: eth 1/1
```

OSPFv3 clear commands

The following OSPFv3 clear commands are supported.

Clearing all OSPFv3 data

You can use the **clear ipv6 ospf all** command to clear all OSPF data by disabling and enabling the OSPFv3 processes as shown in the following.

```
device# clear ipv6 ospf all
```

Syntax: clear ipv6 ospf all

Clearing all OSPFv3 packet counters

You can use the **clear ipv6 ospf traffic** command to clear all OSPFv3 packet counters as shown in the following.

```
device# clear ipv6 ospf traffic
```

Syntax: clear ipv6 ospf traffic

Scheduling Shortest Path First (SPF) calculation

You can use the **clear ipv6 ospf force-spf** command to perform the SPF calculation without clearing the OSPF database, as shown in the following.

```
device# clear ipv6 ospf force-spf
```

Syntax: clear ipv6 ospf force-spf

Clearing all redistributed routes from OSPFv3

You can use the **clear ipv6 ospf redistribution** command to clear all redistributed routes from OSPF, as shown in the following.

```
device# clear ipv6 ospf redistribution
```

Syntax: clear ipv6 ospf redistribution

Clearing OSPFv3 neighbors

You can use the **clear ipv6 ospf neighbor** command to delete and relearn OSPF neighbors, as shown in the following:

- Clearing all OSPF Neighbors
- Clearing OSPF Neighbors Attached to a Specified Interface

Clearing all OSPF neighbors

You can use the **clear ipv6 ospf neighbor all** command to delete and relearn all OSPF neighbors, as shown in the following.

```
device# clear ipv6 ospf neighbor all
```

Syntax: clear ipv6 ospf neighborall

Clearing OSPF neighbors attached to a specified interface

You can use the **clear ipv6 ospf neighbor interface** command to delete and relearn the OSPF neighbors attached to a specified interface, as shown in the following.

```
device# clear ipv6 ospf neighbor interface ethernet 1/1
```

Syntax: clear ipv6 ospf neighbor interface [**ethernet** *slot/port* | **ve** *port-no* | **tunnel** *tunnel-port*] [*nbr-id*]

Specify the interface options as shown in the following options.

ethernet *slot/port* - clears OSPF neighbors on the specified Ethernet interface.

ve *port-no* - clears OSPF neighbors on the specified virtual interface.

tunnel *tunnel-port* - clears OSPF neighbors on the specified tunnel interface.

Specifying the *nbr-id* variable limits the **clear ipv6 ospf neighbor** command to an individual OSPF neighbor attached to the interface.

Clearing OSPFv3 counters

You can use the **ospf counts** command to clear OSPF neighbor's counters as described in the following:

- Clearing all OSPF Counters
- Clearing the OSPF Counters for a Specified Neighbor
- Clearing the OSPF Counters for a Specified Interface

Clearing all OSPFv3 counters

You can clear all OSPF counters using the **clear ipv6 ospf counts** command, as shown in the following.

```
device# clear ipv6 ospf counts
```

Syntax: clear ipv6 ospf counts

Clearing OSPFv3 counters for a specified neighbor

You can clear all OSPF counters for a specified neighbor using the **clear ipv6 counts neighbor** command, as shown in the following.

```
device# clear ipv6 ospf counts neighbor 10.10.10.1
```

Syntax: clear ipv6 ospf counts neighbor *nbr-id*

The *nbr-id* variable specifies the neighbor ID of the OSPF neighbor whose counters you want to clear.

Clearing OSPFv3 counters for a specified interface

You can clear all OSPFv3 counters for a specified interface using the **clear ipv6 counts neighbor interface** command, as shown in the following.

```
device# clear ipv6 ospf counts interface ethernet 3/1
```

Syntax: clear ipv6 ospf counts neighbor [**interface ethernet** *slot/port* | **ve** *port-no* | **tunnel** *tunnel-port*] [**nbr-id**]

Specify the interface options as shown in the following options.

ethernet *slot/port* - clears OSPFv3 counters for OSPFv3 neighbors on the specified Ethernet interface.

ve *port-no* - clears OSPFv3 counters for OSPFv3 neighbors on the specified virtual interface.

tunnel *tunnel-port* - clears OSPFv3 counters for OSPFv3 neighbors on the specified tunnel interface.

Using an *nbr-id* value limits the displayed output to an individual OSPFv3 neighbor attached to the interface.

IS-IS (IPv6)

- IPv6 IS-IS single-topology mode....616
- IS-IS CLI levels....617
- Configuring IPv6 IS-IS....619
- Configuring IPv6 IS-IS single topology....621
- Globally configuring IS-IS on a device....621
- Configuring IPv6 specific address family route parameters....621
- Configuring IS-IS properties on an interface....629
- IPv6 IS-IS Non-Stop Routing....630
- Displaying IPv6 IS-IS information....631
- IPv6 IS-IS Multi-Topology....649
- default-link-metric....655
- reverse-metric....657
- isis reverse-metric....660

Table 136 displays the individual Brocade devices and the IPv6 IS-IS features they support.

TABLE 136 Supported Brocade IPv6 IS-IS features

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
IPv6 IS-IS	Yes	Yes	No	Yes	Yes	Yes	Yes
Redistributing BGP4+ Routes into IPv6 IS-IS	Yes	Yes	No	Yes	Yes	Yes	Yes
Redistributing RIPng Routes into IPv6 IS-IS	Yes	Yes	No	Yes	Yes	Yes	Yes
Redistributing OSPFv3 Routes into IPv6 IS-IS	Yes	Yes	No	Yes	Yes	Yes	Yes
Redistributing Static IPv6 Routes into IPv6 IS-IS	Yes	Yes	No	Yes	Yes	Yes	Yes
Redistributing IPv6 routes learned from directly connected networks	Yes	Yes	No	Yes	Yes	Yes	Yes

TABLE 136 Supported Brocade IPv6 IS-IS features (Continued)

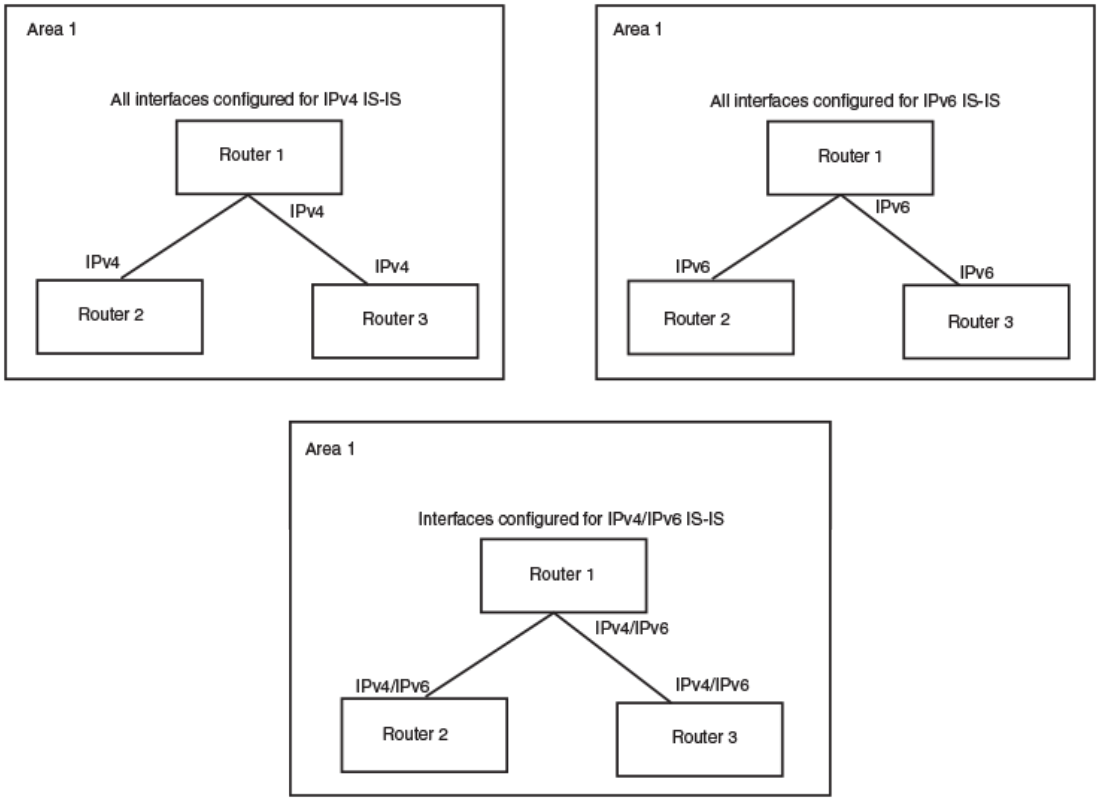
Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
Nonstop routing support	Yes	Yes	No	No	No	No	No
IPv6 Protocol-Support Consistency Checks	Yes	Yes	No	Yes	Yes	Yes	Yes
IS-IS interface metric enhancement	Yes	Yes	No	Yes	Yes	Yes	Yes
IS-IS reverse metric	Yes	Yes	No	Yes	Yes	Yes	Yes
IS-IS Multi-Topology	Yes	Yes	No	Yes	Yes	Yes	Yes

A description of the IS-IS protocol is provided in IS-IS (IPv6). This chapter describes the specific requirements for configuring a device for IPv6 IS-IS.

IPv6 IS-IS single-topology mode

IPv6 IS-IS supports single-topology mode, which means that you can run IPv6 IS-IS concurrently with other network protocols such as IPv4 IS-IS throughout a topology. However, when implementing a single topology, all routers in an area (Level 1 routing) or domain (Level 2 routing) must be configured with the same set of network protocols on all its interfaces, even on loopback interfaces. You can configure IPv4 IS-IS only, IPv6 IS-IS only, or both IPv4 IS-IS and IPv6 IS-IS (Figure 34). For example, to successfully implement both IPv4 and IPv6 IS-IS in an area, you must configure both IPv4 and IPv6 IS-IS on all router interfaces in the area.

FIGURE 34 IPv6 IS-IS in single-topology mode



A single shortest path first (SPF) per level computes the IPv4 and IPv6 routes. The use of a single SPF indicates that both IPv4 and IPv6 IS-IS routing protocols must share a common network topology

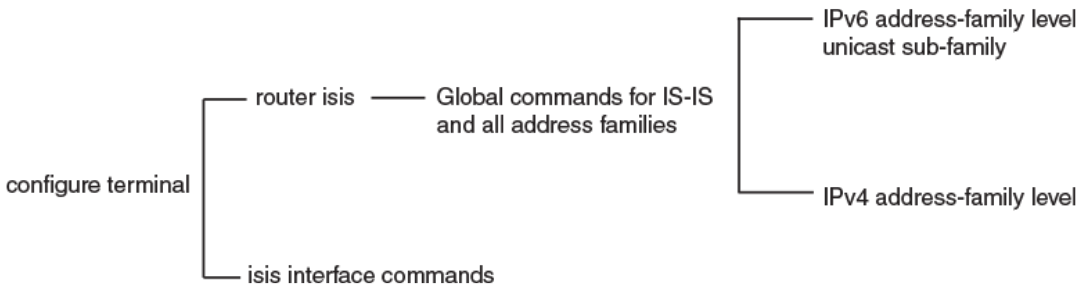
The implementation of IPv4 IS-IS supports type, length, and value (TLV) parameters to advertise reachability to IPv4 networks. The TLVs specify the types of data, the maximum length of the data, and the valid values for the data. IPv6 IS-IS advertises its information using new TLV parameters. The new TLV parameters for IPv6 support an extended default metric value.

In a single topology, if both IPv4 and IPv6 are configured on an interface, metric-style must be set to wide in both address families. Narrow is the default for IPv4. Wide is the default for IPv6.

IS-IS CLI levels

The CLI includes various levels of commands for IS-IS. Figure 35 diagrams these levels that includes the levels used for IPv6 IS-IS.

FIGURE 35 IPv6 IS-IS CLI levels



The IPv6 IS-IS CLI levels are as follows:

- A global level for the configuration of the IS-IS protocol. At this level, all IS-IS configurations at this level apply to IPv4 and IPv6. You enter this layer using the **router isis** command.
  - Under the global level, you specify an address family. Address families separate the IS-IS configuration IPv6 and IPv4. You enter configurations that are for a specific You enter this level by entering the **address-family** command at the router IS-IS level.
  - Under the address family level, you select a sub-address family, which is the type of routes for the configuration. For IS-IS, you specify **unicast** .
  - An interface level

Global configuration level

You enter the global configuration level of IS-IS by entering the following command:

```
device(config)#router isis
device(config-isis-router)#
```

Syntax: router isis

The **(config-isis-router)#** prompt indicates that you are at the global level for IS-IS. Configurations you enter at this level apply to both IS-IS IPv4 and IS-IS IPv6.

Address family configuration level

The implementation of IPv6 IS-IS includes a new configuration level: address family. You enter IS-IS definitions for IPv6 IS-IS under this level. Address-family allows you to create configurations for IPv6 IS-IS unicast routes that are separate and distinct from configurations for IPv4 IS-IS unicast routes.

Under the address family level, Brocade devices support the unicast address family configuration level only. The device enters the IPv6 IS-IS unicast address family configuration level when you enter the following command while at the global IS-IS configuration level:

```
device(config-isis-router)# address-family ipv6 unicast
device(config-isis-router-ipv6u)#
```

Syntax: address-family ipv6 unicast

The **(config-isis-router-ipv6u)#** prompt indicates that you are at the IPv6 IS-IS unicast address family configuration level. While at this level, you can access several commands that allow you to configure IPv6 IS-IS unicast routes.

NOTE
Each address family configuration level allows you to access commands that apply to that particular address family only. To enable a feature in a particular address family, you must specify any associated commands for that feature in that particular address family. You cannot expect the feature, which you may have configured in the IPv4 IS-IS unicast address family, to work in the IPv6 IS-IS unicast address family unless it is explicitly configured in the IPv6 IS-IS unicast address family.

To exit from the IPv6 IS-IS unicast address family configuration level, enter the following command:

```
device(config-isis-router-ipv6u)# exit-address-family
device(config-isis-router)#
```

Entering this command returns you to the global IS-IS configuration level.

Interface level

Some IS-IS definitions are entered at the interface level. To change to the interface level for IS-IS configuration, enter the following command.

```
device(config)# interface ethernet 2/3
device(config-if-e1000-2/3)#ipv6 router isis
```

Syntax: ipv6 router isis

Configuring IPv6 IS-IS

Enabling IPv6 IS-IS globally

Follow the steps listed below to configure IPv6 IS-IS globally.

1. You must enable the forwarding of IPv6 traffic on the device using the **ipv6 unicast-routing** command. Enter a command such as the following:

   ```
   device#configure terminal
   device(config)# ipv6 unicast-routing
   ```

   Syntax: [no] ipv6 unicast-routing
2. Globally enable IS-IS by entering the following command:

   ```
   device(config)# router isis
   ISIS: Please configure NET!
   ```

   Once you enter**router isis,** the device enters the IS-IS router configuration level.

   Syntax: [no] router isis

   To disable IS-IS, use the **no** form of this command.
3. If you have not already configured a NET for IS-IS, enter commands such as the following:

   ```
   device(config-isis-router)# net 49.2211.0000.00bb.cccc.00
   device(config-isis-router)#
   ```

   The commands in the example above configure a NET that has the area ID 49.2211, the system ID 0000.00bb.cccc (the device's base MAC address), and SEL value 00.

Syntax: [no] net area-id.system-id.sel

The *area-id* parameter specifies the area and has the format xx or xx.xxxx. For example, 49 and 49.2211 are valid area IDs.

The *system-id* parameter specifies the device's unique IS-IS router ID and has the format xxxx.xxxx.xxxx. You can specify any value for the system ID. A common practice is to use the device's base MAC address as the system ID. The base MAC address is also the MAC address of port 1. To determine the base MAC address, enter the following command at any level of the CLI: **show interfaces brief** . The base MAC address is listed in the first row of information, in the MAC column.

You must use the same system ID in all the NETs on the device.

NOTE
The parameter descriptions above are the recommended values for the NET. However, the CLI accepts any value that fits within the following lengths and formats:xx.xxxx.xxxx.xxxx.00 - minimum length of NETxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.00 - maximum length of NET

The *sel* parameter specifies the NSAP Selector (SEL). This value must always be 00 (two zeros). The value 00 indicates that this address is an NET.

To delete a NET, use the **no** form of this command.

4. Configure an IPv6 IS-IS single topology. Refer to Configuring IPv6 IS-IS single topology on page 621.
5. Configure IS-IS parameters. Refer to the sections Globally configuring IS-IS on a device on page 621, Configuring IPv6 specific address family route parameters on page 621, and Configuring IS-IS properties on an interface on page 629.

Enabling IS-IS and assigning an IPv6 address to an interface

To configure IPv6 IS-IS on the desired device interfaces, enter commands such as the following:

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 address 2001:db8:12d:1300::/64 eui-64
device(config-if-e100-3/1)# ipv6 router isis
```

The commands in this example assign the global IPv6 prefix 2001:db8:12d:1300::/64 to Ethernet interface 3/1 and enable IPv6 IS-IS on the interface.

Syntax: ipv6 address ipv6-prefix/prefix-length [**eui-64**]

You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The **eui-64** keyword configures the global or unique local address with an EUI-64 interface ID in the low-order 64 bits. The interface ID is automatically constructed in IEEE EUI-64 format using the interface's MAC address.

Syntax: [no] ipv6 router isis

To disable IPv6 IS-IS on an interface, use the **no** form of this command.

The following configuration tasks are optional:

- Configure IPv6 route parameters.
- Redistribute routes from other route sources into IPv6 IS-IS.

- Perform IPv6 IS-IS adjacency checks.
- Disable partial SPF calculations.

Configuring IPv6 IS-IS single topology

If your IS-IS single topology will support both IPv6 and IPv4, you can configure both IPv6 and IPv4 on an IS-IS interface for Level 1, Level 2, or both Level 1 and Level 2. However, if you configure both IPv6 and IPv4 on an IS-IS interface, they must be configured to run on the same level. For example, you can configure IPv6 to run on Level 1 on an interface and IPv4 to also run on Level 1 on the same interface. However, you cannot configure IPv6 to run on Level 1 on an interface and IPv4 to run to Level 2 on the same interface.

To configure an IPv6 IS-IS single topology, you must perform the tasks listed below.

1. Globally enable IS-IS and configure at least one Network Entity Title (NET). The NET is the device's network interface with IS-IS. You can configure up to three NETs on a device.
2. Configure the desired device interfaces with an IPv6 address and enable IPv6 IS-IS on the device interfaces.
3. Configure IS-IS parameters. Refer to the sections Globally configuring IS-IS on a device on page 621, Configuring IPv6 specific address family route parameters on page 621, and Configuring IS-IS properties on an interface on page 629.

Globally configuring IS-IS on a device

The following configuration tasks described in IS-IS (IPv4) on page 255, apply to IS-IS IPv6 configuration:

- Setting the Overload Bit
- Configuring Authentication
- Changing the IS-IS Level Globally
- Disabling or Re-enabling Display of Hostname
- Changing the Sequence Numbers PDU Interval
- Changing the Maximum LSP Lifetime
- Changing the LSP Refresh Interval
- Changing the LSP Generation Interval
- Changing the LSP Interval and Retransmit
- Changing the SPF Timer
- Globally Disabling or Re-Enabling Hello Padding
- Logging Adjacency Changes
- Disabling Partial SPF Calculations

Configuring IPv6 specific address family route parameters

This section describes how to modify the IS-IS the parameters for the IS-IS IPv6 address family.

Changing the maximum number of load sharing paths

By default, IPv6 IS-IS can calculate and install four equal-cost paths into the IPv6 forwarding table. You can change the number of paths IPv6 IS-IS can calculate and install in the IPv6 forwarding table to an amount from 1 - 8. If you change the number of paths to one, the device does not load share route paths learned from IPv6 IS-IS.

For example, to change the number of paths IPv6 IS-IS can calculate and install in the IPv6 forwarding table to three, enter the following command at the IPv6 IS-IS unicast address family configuration level:

```
device(config-isis-router-ipv6u)# maximum-paths 8
```

Syntax: [no] maximum-paths number

The *number* parameter specifies the number of paths IPv6 IS-IS can calculate and install in the IPv6 forwarding table. The *number* value range is 2 to 32 and the default is 1.

NOTE
The maximum number of paths supported by the BR-MLX-10Gx24-DM module is 16.

To return to the default number of maximum paths, enter the **no** form of this command.

Enabling advertisement of a default route

By default, the device does not generate or advertise a default route to its neighboring ISs. A default route is not advertised even if the device's IPv6 route table contains a default route. You can enable the device to advertise a default route to all neighboring ISs using one of the following methods. By default, the feature originates the default route at Level 2 only. However, you can apply a route map to originate the default route to Level 1 only or at both Level 1 and Level 2.

NOTE
This feature requires the presence of a default route in the IPv6 route table.

To enable the device to advertise a default route that is originated a Level 2, enter the following command at the IPv6 IS-IS unicast address family configuration level:

```
device(config-isis-router-ipv6u)# default-information-originate
```

This command enables the device to advertise a default route into the IPv6 IS-IS area to which the device is attached.

Syntax: [no] default-information-originate [**route-map name**]

The **route-map***name* parameter allows you to specify the level on which to advertise the default route. You can specify one of the following:

- Advertise to Level-1 ISs only.
- Advertise to Level-2 ISs only.
- Advertise to Level-1 and Level-2 ISs.

NOTE
The route map must be configured before you can use the route map as a parameter with the **default-information-originate** command.

To use a route map to configure the device to advertise a default route to Level 1, enter commands such as the following at the Global CONFIG level:

```
device(config)# route-map default_level1 permit 1
device(config-routemap default_level1)# set level level-1
device(config-routemap default_level1)# router isis
device(config-isis-router)# address-family ipv6 unicast
device(config-isis-router-ipv6u)# default-information-originate route-map
default_level1
```

These commands configure a route map to set the default advertisement level to Level 1 only.

Syntax: [no] route-map map-name permit | **deny sequence-number**

Syntax: [no] set level level-1 | **level-1-2** | **level-2**

For this use of a route map, use the **permit** option and do not specify a **match** statement. Specify a **set** statement to set the level to one of the following:

- **level-1** - Level 1 only.
- **level-1-2** - Level 1 and Level 2.
- **level-2** - Level 2 only (default).

Changing the administrative distance for IPv6 IS-IS

When the device has paths from multiple routing protocols to the same destination, it compares the administrative distances of the paths and selects the path with the lowest administrative distance to place in the IPv6 route table.

For example, if the device has a path from RIPng, from OSPFv3, and IPv6 IS-IS to the same destination, and all the paths are using their protocols' default administrative distances, the device selects the OSPFv3 path, because that path has a lower administrative distance than the RIPng and IPv6 IS-IS paths.

Here are the default IPv6 administrative distances on the device:

- Directly connected - 0 (this value is not configurable)
- Static - 1 (applies to all static routes, including default routes)
- EBGP - 20
- OSPFv3 - 110
- IPv6 IS-IS - 115
- RIPng - 120
- IBGP - 200
- Local BGP - 200
- Unknown - 255 (the device will not use this route)

Lower administrative distances are preferred over higher distances. For example, if the device receives routes for the same network from IPv6 IS-IS and from RIPng, it will prefer the IPv6 IS-IS route by default.

To change the administrative distance for IPv6 IS-IS routes, enter the following command at the IPv6 IS-IS unicast address family configuration level:

```
device(config-isis-router-ipv6u)# distance 100
```

Syntax: [no] distance number

This command changes the administrative distance for all IPv6 IS-IS routes to 100.

The *number* parameter specifies the administrative distance. You can specify a value from 1 - 255. (Routes with a distance value of 255 are not installed in the routing table.) The default for IPv6 IS-IS is 115.

Configuring summary prefixes

You can configure summary prefixes to aggregate IPv6 IS-IS route information. Summary prefixes can enhance performance by reducing the size of the Link State database, reducing the amount of data a router needs to send to its neighbors, and reducing the CPU cycles used for IPv6 IS-IS.

When you configure a summary prefix, the prefix applies only to Level-2 routes by default. You can specify Level-1 only, Level-2 only, or Level-1 and Level-2 when you configure the prefix.

For example, to configure a summary prefix of 2001:db8::/32 to be advertised to Level-1 routes only, enter the following command at the IPv6 IS-IS unicast address family configuration level:

```
device(config-isis-router-ipv6u)# summary-prefix 2001:db8::/32 level-1
```

Syntax: [no] summary-prefix ipv6-prefix/prefix-length [**level-1** | **level-1-2** | **level-2-only**]

The *ipv6-prefix*/*prefix-length* parameter specifies the aggregate address. You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The **level-1** | **level-1-2** | **level-2-only** parameter specifies the route types to which the aggregate route applies. The default is **level-2-only**.

Redistributing routes into IPv6 IS-IS

To redistribute routes into IPv6 IS-IS, you can perform the following configuration tasks:

- Change the default redistribution metric (optional).
- Configure the redistribution of a particular route type into IPv6 IS-IS (mandatory).

The device can redistribute routes from the following route sources into IPv6 IS-IS:

- BGP4+.
- RIPng.
- OSPFv3.
- Static IPv6 routes.
- IPv6 routes learned from directly connected networks.

The device can also can redistribute Level-1 IPv6 IS-IS routes into Level-2 IPv6 IS-IS routes, and Level-2 IPv6 IS-IS routes into Level-1 IPv6 IS-IS routes.

Route redistribution from other sources into IPv6 IS-IS is disabled by default. When you enable redistribution, the device redistributes routes only into Level 2 by default. You can specify Level 1 only, Level 2 only, or Level 1 and Level 2 when you enable redistribution.

The device automatically redistributes Level-1 routes into Level-2 routes. Thus, you do not need to enable this type of redistribution. You also can enable redistribution of Level-2 routes into Level-1 routes.

The device attempts to use the redistributed route's metric as the route's IPv6 IS-IS metric. For example, if an OSPFv3 route has an OSPF cost of 20, the device uses 20 as the route's IPv6 IS-IS metric. The device uses the redistributed route's metric as the IPv6 IS-IS metric unless the route does not a have a valid metric. In this case, the device assigns the default metric value to the route. For

information about the default metric, refer to the "Changing the Default Redistribution Metric" section, which follows this section.

Changing the default redistribution metric

When IPv6 IS-IS redistributes a route from another route source (such as OSPFv3, BGP4+, or a static IPv6 route) into IPv6 IS-IS, it uses the route's metric value as its metric when the metric is not modified by a route map or metric parameter and the default redistribution metric is set to its default value of 0. You can change the default metric to a value from 0 - 65535.

NOTE
The implementation of IS-IS does not support the optional metric types Delay, Expense, or Error.

For example, to change the default metric to 20, enter the following command at the IPv6 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv6u)# default-metric 20
```

Syntax: [no] default-metric number

The *number* parameter specifies the default metric. You can specify a value from 0 - 65535. The default is 0.

The restore the default value for the default metric, enter the **no** form of this command.

Globally change the default redistribution metric

The default-link-metric command allows you to change the metric value globally for all the active ISIS interfaces using one command. You can still configure the interface level metric. If ISIS metric is configured on the interface, it will take the precedence over the global configuration.

Configuration steps

1. Configure router ISIS using the router isis command.
2. Go to the appropriate address-family using address-family [ipv4/ipv6] unicast command.
3. Configure default metric using default-link-metric <value> command.

Configuration example

The following global configuration example is for the IPv4 address-family. It can be similarly configured for IPv6 address-family.

```
device(config)#router isis
device(config-isis-router)#address-family ipv4 unicast
device(config-isis-router-ipv4u)# default-link-metric  40
```

Syntax: [no] default-link-metric *value* [**level-1** | **level-2**]

The *value* parameter is the default-link-metric value to be set for the given address-family. This is a required parameter for this command. There is no default value for this parameter. For metric-style narrow: 1 to 63. For metric-style wide: 1 to 16777215.

The *level* parameter is an optional parameter used to set the default-metric for only one of the levels. If this parameter is not given, the default-link-metric will be applied to both level-1 and level-2.

The **no** version of command will revert the metric value to default, which is 10.

IPv6 metric behavior with multi-topology configuration

The default-link-metric for IPv6 will depend upon the multi-topology configuration.

No multi-topology: The IPv6 default-link-metric will be same as that configured for IPv4 address-family.

Multi-topology : The IPv6 default-link-metric will be equal to the value configured for IPv6 address-family.

Multi-topology transition : The IPv6 default-link-metric will be equal to the value configured for IPv6 address-family.

Metric behavior with change in metric-style

There are two types of metric styles in ISIS, narrow metric and wide metric. The range of the metric value is different in both of these styles. If there is a change in the metric-style configuration, the default-link-metric will also change with it. The new value of the default-link-metric will be equal to the minimum of a) configured value and b) the maximum value supported for the new metric-style.

If the metric style changes from narrow metric to wide metric, there will be no change in the value of default-link-metric. If the metric style changes from wide metric to narrow metric, and if the value of default-link-metric is greater than 63, the default-link-metric will now take the value 63, as it is the maximum supported in the narrow metric.

ISIS Show command

The show isis command and show ipv6 isis command output has been modified to reflect the default-link-metric configured.

```
device#sh ipv6 isis
.....
 Default redistribution metric: 0
 Default link metric for level-1: 15
 Default link metric for level-2: 9
 Protocol Routes redistributed into IS-IS:
```

Redistributing static IPv6 routes into IPv6 IS-IS

To redistribute static IPv6 routes from the IPv6 static route table into IPv6 IS-IS routes, enter the following command at the IPv6 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv6u)# redistribute static
```

This command configures the device to redistribute all static IPv6 routes into Level-2 IS-IS routes.

Syntax: [no] redistribute static [**level-1** | **level-1-2** | **level-2** | **metric number** | **metric-type external** | **internal** | **route-map name**]

The **level-1**, **level-1-2**, and **level-2** keywords restrict redistribution to the specified IPv6 IS-IS level.

The **metric***number* parameter changes the metric. You can specify a value from 0 - 4294967295.

The **metric-type external** | **internal** parameter restricts redistribution to one of the following:

- **external** - The metric value is not comparable to an IPv6 IS-IS internal metric and is always higher than the IPv6 IS-IS internal metric.
- **internal** - The metric value is comparable to metric values used by IPv6 IS-IS. This is the default.

The **route-map***name* parameter restricts redistribution to those routes that match the specified route map. The route map must already be configured before you use the route map name with the **redistribute** command. For example, to configure a route map that redistributes only the static IPv6 routes to the destination networks 2001:db8::/32, enter commands such as the following.

```
device(config)# ipv6 access-list static permit any 2001:db8::/32
device(config)# route-map static permit 1
device(config-routemap static)# match ip address static
device(config-routemap static)# router isis
device(config-isis-router)# address-family ipv6 unicast
device(config-isis-router-ipv6u)# redistribute static route-map static
```

Redistributing directly connected routes into IPv6 IS-IS

To redistribute directly connected IPv6 routes into IPv6 IS-IS routes, enter the following command at the IPv6 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv6u)# redistribute connected
```

This command configures the device to redistribute all directly connected routes in the IPv6 route table into Level-2 IPv6 IS-IS.

Syntax: [no] redistribute connected [**level-1** | **level-1-2** | **level-2** | **metric number** | **metric-type external** | **internal** | **route-map name**]

The parameters are the same as the parameters for the **redistribute static** command.

Redistributing RIPng routes into IPv6 IS-IS

To redistribute RIPng routes into IPv6 IS-IS, enter the following command at the IPv6 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv6u)# redistribute rip
```

This command configures the device to redistribute all RIPng routes into Level-2 IS-IS.

Syntax: [no] redistribute rip [**level-1** | **level-1-2** | **level-2** | **metric number** | **metric-type external** | **internal** | **route-map name**]

The parameters are the same as the parameters for the **redistribute static** command.

Redistributing OSPF version 3 routes into IPv6 IS-IS

To redistribute OSPFv3 routes into IPv6 IS-IS, enter the following command at the IPv6 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv6u)# redistribute ospf
```

This command configures the device to redistribute all OSPFv3 routes into Level-2 IPv6 IS-IS.

Syntax: [no] redistribute ospf [**level-1** | **level-1-2** | **level-2** | **match external1** | **external2** | **internal** | **metric number** | **metric-type external** | **internal** | **route-map name**]

Most of the parameters are the same as the parameters for the **redistribute static** command. However, the **redistribute ospf** command also has the **match external1** | **external2** | **internal** parameter. This

parameter specifies the OSPF route type you want to redistribute into IPv6 IS-IS. By default, the **redistribute ospf** command redistributes only internal routes:

- **external1** - An OSPF type 1 external route.
- **external2** - An OSPF type 2 external route.
- **internal** - An internal route calculated by OSPF.

Redistributing BGP4+ routes into IPv6 IS-IS

To redistribute BGP4+ routes into IPv6 IS-IS, enter the following command at the IPv6 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv6u)# redistribute bgp
```

This command configures the device to redistribute all its BGP4 routes into Level-2 IPv6 IS-IS.

Syntax: [no] redistribute bgp [**level-1** | **level-1-2** | **level-2** | **metric number** | **metric-type external** | **internal** | **route-map name**]

The parameters are the same as the parameters for the **redistribute static** command.

Redistributing IPv6 IS-IS routes within IPv6 IS-IS

In addition to redistributing routes from other route sources into IPv6 IS-IS, the device can redistribute Level 1 IPv6 IS-IS routes into Level 2 IPv6 IS-IS routes, and Level 2 IPv6 IS-IS routes into Level 1 IPv6 IS-IS routes. By default, the device redistributes routes from Level 1 into Level 2.

NOTE
The device automatically redistributes Level 1 routes into Level 2 routes, even if you do not enable redistribution.

For example, to redistribute all IPv6 IS-IS routes from Level 2 into Level 1, enter the following command at the IPv6 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv6u)# redistribute isis level-2 into level-1
```

The device automatically redistributes Level-1 routes into Level 2.

Syntax: [no] redistribute isis level-1 into level-2 | **level-2 into level-1** [**prefix-list name**]

The **level-1 into level-2** | **level-2 into level-1** parameter specifies the direction of the redistribution:

- **level-1 into level-2** - Redistributes Level 1 routes into Level 2. This is the default.
- **level-2 into level-1** - Redistributes Level 2 routes into Level 1.

The optional **prefix-list***name* parameter allows you to specify the IPv6 prefixes that you want redistributed from Level 1 into Level 2 and from Level 2 into Level 1. Specify the name of the IPv6 prefix list that contains the desired prefixes.

For example, to redistribute the IPv6 prefix 2001:db8::/32 from Level 2 into Level 1, enter commands such as the following.

```
device(config)# ipv6 prefix-list routesfor2001 permit 2001:db8::/32
device(config)# router isis
device(config-isis-router)# address-family ipv6 unicast
device(config-isis-router-ipv6u)# redistribute isis level-2 into level-1 prefix-list
routesfor2001
```

Disabling and re-enabling IPv6 protocol-support consistency checks

As discussed in IPv6 IS-IS single-topology mode on page 616, an IS-IS single topology must be configured to run the same set of network protocols (IPv4 IS-IS only, IPv6 IS-IS only, or both IPv4 IS-IS and IPv6 IS-IS).

By default, IS-IS performs consistency checks on hello packets. If a hello packet does not have the same configured network protocols, IS-IS rejects the packet. For example, a hello packet from a router running IPv4 and IPv6 IS-IS will be rejected by a router running either IPv4 IS-IS only or IPv6 IS-IS only, and the two routers will not become adjacent.

To allow two routers running different sets of network protocols to form an adjacency, enter the following command at the IPv6 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv6u)# no adjacency-check
```

This command disables the IPv6 IS-IS consistency check.

Syntax: [no] adjacency-check

To re-enable the consistency check, enter the following command at the IPv6 IS-IS unicast address family configuration level.

```
device(config-isis-router-ipv6u)# adjacency-check
```

Configuring IS-IS properties on an interface

The parameter settings for configuring IS-IS properties on a device apply to both IS-IS IPv4 and IS-IS IPv6 except for Changing the metric added to advertised routes on page 629 as described below. For details on how to perform all other IS-IS properties on an Interface, refer to Configuring IS-IS properties on an interface

Changing the metric added to advertised routes

When the device originates an IS-IS route or calculates a route, the device adds a metric (cost) to the route. Each IS-IS interface has a separate metric value. The default is 10.

The device applies the interface-level metric to routes originated on the interface and also when calculating routes. The device does not apply the metric to link-state information that the device receives from one IS and floods to other ISs.

The default interface metric is 10. You can change the metric on an individual interface to a value in one of the following ranges:

- 1 - 63 for the narrow metric style (the default metric style)
- 1 - 16777215 for the wide metric style

NOTE
If the metric value you want to use is higher than 63 but you have not changed the metric style to wide, change the metric style first, then set the metric. The IS-IS neighbors that will receive the advertisements also must be enabled to receive wide metrics.

To change the IS-IS metric on an interface, use the following CLI method.

```
device(config-isis-router)# interface ethernet 2/8
device(config-if-e1000-2/8)# isis metric 44
```

Syntax: [no] isis metric num

The *num* parameter specifies the metric. The range of values you can specify depends on the metric style. You can specify 1 - 63 for the narrow metric style or 1 - 16777215 for the wide metric style. The default in either case is 10.

When IPv6 IS-IS is enabled in a single topology, you must set the metric-style to be wide if you want to use an interface metric greater than 63.

IPv6 IS-IS Non-Stop Routing

Overview

NOTE
IPv6 IS-IS NSR is not supported on the Brocade NetIron CES Series and Brocade NetIron CER Series platforms.

IPv6 IS-IS Non-Stop Routing (NSR) enables the IPv6 IS-IS router to maintain topology and data flow to avoid re-convergence in the network during a processor switchover or hitless-reload event. The IS-IS Bidirectional Forwarding Detection (BFD) sessions survive the switchover and hitless-reload conditions. In general, a router restart causes its peer to remove the routes originated from the router and reinstalls them. This IS-IS NSR feature enables the router to maintain neighbors and LSA database with its peer on the event of a router restart. This feature is compatible with IPv4 IS-IS NSR.

NOTE
IPv6 IS-IS NSR is independent of Graceful Restart (GR) and GR help role mechanisms.

Limitations

- The IS-IS over GRE tunnel feature does not support IS-IS NSR. The GRE tunnel interface types are not supported.
- The IS-IS shortcuts are not supported because they depend on the MPLS tunnel.
- If the IS-IS hellos are forwarded at Layer 2 and the device executes a hitless-reload, hellos will not be forwarded for a brief time. The IS-IS adjacencies are lost for 12 seconds and there will be data traffic loss.
- The configuration events that occur close to switchover or hitless-reload may get lost due to CLI synchronization issues.
- The neighbor or interface state changes close to switchover or hitless-reload cannot be handled.
- The IS-IS neighbor hold timer is restarted upon IS-IS NSR switchover or hitless-reload.
- The traffic counters are not synchronized because the neighbor and LSP database counters are recalculated on the standby module during synchronization.

- With IS-IS NSR enabled, after switchover or hitless-reload to standby MP, IS-IS routes, LSP database and neighbor adjacencies are maintained so that there will be no loss of existing traffic to the IS-IS destinations.
- The IS-IS NSR hitless failover event may not be completely invisible to the network because, after switchover, additional flooding of CSNP packets will occur in the directly connected neighbors.

Configuring IS-IS NSR

To globally enable IS-IS NSR, enter the following commands.

```
device(config)# router isis
device(config-isis-router)# nonstop-routing
```

To globally disable IS-IS NSR, enter the following commands.

```
device(config)# router isis
device(config-isis-router)# no nonstop-routing
```

Syntax: [no] nonstop-routing

Displaying IPv6 IS-IS information

You can display the following information about IPv6 IS-IS:

- General IPv6 IS-IS information.
- IPv6 IS-IS configuration information.
- IPv6 IS-IS error statistics.
- LSP database entries.
- IS-IS system ID to hostname mappings.
- IPv6 IS-IS interface information.
- IPv6 IS-IS memory usage information.
- IPv6 IS-IS neighbor information.
- IPv6 IS-IS path information.
- IPv6 IS-IS redistribution information.
- IPv6 IS-IS route information.
- IPv6 IS-IS traffic statistics.

Displaying IPv6 IS-IS information

To display general IPv6 IS-IS information, enter the following command at any CLI level.

```
device# show ipv6 isis
 IS-IS  Routing Protocol Operation State: Enabled
 IS-Type: Level-1-2
 System ID: 0000.0055.0008
 Manual area address(es):
   49.8585
 Interfaces with Integrated IS-IS for IPv6 configured:
    Interface 4/1    Interface 4/2    Interface 4/11    Interface 4/12
    Interface 4/13   Interface 4/14   Interface 4/15    Interface 4/16
    Interface 4/17   Interface 4/35   Interface 4/36    Interface 4/37
    Interface 4/38   Interface v43    Interface v44     Interface lb1
 Following Routes are Redistributed into IS-IS for IPv6:
   CONNECTED
 Number of Routes redistributed into IS-IS: 1
```

```
Domain password: None
Area password: None
IS-IS for IPV6 Route Administrative Distance: 115
Hold Time Between Two SPF Calculations: 5
Global Hello Padding: Enabled
```

Syntax: show ipv6 isis

This display shows the following information.

TABLE 137 IPv6 IS-IS information fields

This field...	Displays...
IS-IS Routing Protocol Operation State	The operating state of IPv6 IS-IS. Possible states include the following: • Enabled - IPv6 IS-IS is enabled. • Disabled - IPv6 IS-IS is disabled.
IS Type	The intermediate system type. Possible types include the following: • Level 1 only - The device routes traffic only within the area in which it resides. • Level 2 only - The device routes traffic between areas of a routing domain. • Level 1-2 - The device routes traffic within the area in which it resides and between areas of a routing domain.
System ID	The unique IS-IS router ID. Typically, the device's base MAC address is used as the system ID.
Manual area address(es)	Area address(es) of the device.
Interfaces with Integrated IS-IS for IPv6 configured	Interfaces on which IPv6 IS-IS is configured.
Following Routes are Redistributed into IS-IS for IPv6	Routes that are redistributed into IPv6 IS-IS. Possible routes include the following: • BGP - BGP4+ routes are redistributed into IPv6 IS-IS. • RIP - RIPng routes are redistributed into IPv6 IS-IS. • OSPF - OSPFv3 routes are redistributed into IPv6 IS-IS. • STATIC - Static IPv6 routes are redistributed into IPv6 IS-IS. • CONNECTED - IPv6 routes learned from directly connected networks are redistributed into IPv6 IS-IS.
Number of Routes redistributed into IS-IS	The number of routes distributed into IS-IS
Domain password	The domain password, if one is configured.
Area password	The domain password, if one is configured.
IS-IS IPv6 Route Administrative Distance	The current setting of the IPv6 IS-IS administrative distance.
Hold Time Between Two SPF Calculations	The setting of the SPF timer, which causes the device to recalculate the SPF tree of its IPv6 IS-IS links following a change in topology or the link state database.

TABLE 137 IPv6 IS-IS information fields (Continued)

This field...	Displays...
Global Hello Padding	The setting of the global hello padding feature, which can be one of the following: • Disabled - Global padding for hello packets is disabled. • Enabled - Global padding for hello packets is enabled.

Displaying the IPv6 IS-IS configuration in the running configuration

You can display the IPv6 IS-IS commands that are in effect on the device.

NOTE
The running configuration does not list the default values. Only commands that change a setting or add configuration information are displayed.

To display the IPv6 IS-IS configuration, enter the following command at any CLI level.

```
device# show ipv6 isis config
Current ISIS configuration:
router isis
 net  49.6561.0000.0022.2222.00
 address-family ipv4 unicast
 distance  135
 redistribute static
 exit-address-family
 address-family ipv6 unicast
 redistribute static
 exit-address-family
end
```

Syntax: show ipv6 isis config

The running configuration shown in this example contains the following commands:

- Global IPv6 IS-IS commands that enable IS-IS.
- Address family commands that configure IPv4 IS-IS unicast routes.
- Address family commands that configure IPv6 IS-IS unicast routes.

Displaying IPv6 IS-IS error statistics

To display IPv6 IS-IS error statistics, enter the following command at any level of the CLI.

```
device# show ipv6 isis counts
 Area Mismatch: 0
 Max Area Mismatch: 0
 System ID Length Mismatch: 0
 Authentication Fail: 0
 Corrupted LSP: 0
 LSP Sequence Number Skipped: 0
 LSP Max Sequence Number Exceeded: 0
 Level-1 Database Overload: 0
 Level-2 Database Overload: 0
 Our LSP Purged: 0
```

Syntax: show ipv6 isis counts

This display shows the following information.

TABLE 138 IPv6 IS-IS error statistics

This field...	Displays...
Area Mismatch	The number of times the device interface was unable to create a Level-1 adjacency with a neighbor because the device interface and the neighbor did not have any areas in common.
Max Area Mismatch	The number of times the device received a PDU with a value for maximum number of area addresses that did not match the device's value for maximum number of area addresses.
System ID Length Mismatch	The number of times the device received a PDU with an ID field that was a different length than the ID field length configured on the device.
Authentication Fail	The number of times authentication failed because the device was configured to authenticate IPv6 IS-IS packets in the packet's domain or area, but the packet did not contain the correct password.
Corrupted LSP	The number of times the device detected a corrupted LSP in the device's memory.
LSP Sequence Number Skipped	The number of times the device received an LSP with a sequence number that was more than 1 higher than the sequence number of the previous LSP received from the same neighbor.
LSP Max Sequence Number Exceeded	The number of times the device attempted to set an LSP sequence number to a value higher than the highest number in the CSNP sent by the Designated IS.
Level-1 Database Overload	The number of times the Level-1 state on the device changed from Waiting to On or from On to Waiting: • Waiting to On - This change can occur when the device recovers from a previous Level-1 LSP database overload and is again ready to receive new LSPs. • On to Waiting - This change can occur when the device's Level-1 LSP database is full and the device receives an additional LSP, for which there is no room.
Level-2 Database Overload	The number of times the Level-2 state on the device changed from Waiting to On or from On to Waiting: • The change from Waiting to On can occur when the device recovers from a previous Level-2 LSP database overload and is again ready to receive new LSPs. • The change from On to Waiting can occur when the device's Level-2 LSP database is full and the device receives an additional LSP, for which there is no room.
Our LSP Purged	The number of times the device received an LSP that was originated by the device itself and had age zero (aged out).

Displaying LSP database entries

You can display summary or detailed information about the entries in the LSP database.

NOTE
The device maintains separate LSP databases for Level 1 LSPs and Level 2 LSPs.

To display summary information about the entries in the LSP database, enter the following command at any level of the CLI.

```
device# show ipv6 isis database
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
Router1.00-00         0x00000003   0x9a6b        574               0/0/0
Router2.00-00*        0x00000002   0x609d        540               0/0/0
Router2.01-00*        0x00000001   0x0fcf        539               0/0/0
IS-IS Level-2 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
Router1.00-00         0x00000003   0xe2da        574               0/0/0
Router2.00-00*        0x00000002   0x0585        540               0/0/0
Router2.01-00*        0x00000001   0x0fcf        539               0/0/0
```

The command in this example displays information for the LSPS in the device's Level-1 and Level-2 LSP databases. Notice that the display groups the Level-1 and Level-2 LSPs separately.

Syntax: show ipv6 isis database [**HHHH.HHHH.HHHH.HH-HH** | **detail** | **l1** | **l2** | **level1** | **level2**]

The *HHHH.HHHH.HHHH.HH-HH* parameter restricts the display to the entry for the specified LSPID. (The LSPID consists of the source ID (HHHH.HHHH.HHHH), the pseudonode (HH-), and LSPID (-HH). To determine the device's source ID, use the **show ipv6 isis** command. For more information, refer to Displaying IPv6 IS-IS information on page 631. To determine the pseudonode and LSPID, use the **show ipv6 isis database** command.

NOTE
Name mapping is enabled by default. When name mapping is enabled, the output of the **show ipv6 isis database** command uses the hostname instead of the system ID. To disable mapping so that these displays use the system ID instead, enter the **no hostname** command at the IS-IS router configuration level.

The **detail** parameter displays detailed information about the LSPs. The detailed information display is discussed later in this section.

The **l1** and **level1** parameters restrict the display to the Level-1 LSP entries. You can use these parameters interchangeably.

The **l2** and **level2** parameters restrict the display to the Level-2 LSP entries. You can use these parameters interchangeably.

This display shows the following information.

TABLE 139 IPv6 IS-IS summary LSP database information

This field...	Displays...
LSPID	The LSP ID, which consists of the source ID (HHHH.HHHH.HHHH), the pseudonode (HH-), and LSPID (-HH). **Note** : If the address has an asterisk (*) at the end, this indicates that the LSP is locally originated.
LSP Seq Num	The sequence number of the LSP.

TABLE 139 IPv6 IS-IS summary LSP database information (Continued)

This field...	Displays...
LSP Checksum	The checksum calculated by the device that sent the LSP and used by the device to verify that the LSP was not corrupted during transmission over the network.
LSP Holdtime	The maximum number of seconds during which the LSP will remain valid. **Note** : The IS that originates the LSP starts the timer for the LSP. As a result, LSPs do not all have the same amount of time remaining when they enter the device's LSP database.
ATT	A 4-bit value extracted from bits 4 - 7 in the Attach field of the LSP.
P	The value in the Partition option field of the LSP. The field can have one of the following values: • 0 - The IS that sent the LSP does not support partition repair. • 1 - The IS that sent the LSP supports partition repair.
OL	The value in the LSP database overload field of the LSP. The field can have one of the following values: • 0 - The overload bit is off. • 1 - The overload bit is on, indicating that the IS that sent the LSP is overloaded and should not be used as a Level-2 router.

You can display the detailed information of all the LSPs in the LSP databases with IS-IS MT transition support enabled or disabled, by entering the following command at any level of the CLI.

The following example shows the output for the **show ipv6 isis database detail** command without the transition support.

```
device# show ipv6 isis database detail
IS-IS Level-1 Link State Database
LSPID                                 Seq Num     Checksum  Holdtime  ATT/P/OL
Dist1.00-00                           0x000001b2  0x0ed4    1183      0/0/0
   Area Address:  00.0000
   NLPID:  IPv6 IP
   Topology:  IPv6(Ovld:0 Att:0) IPv4
   Hostname:  Dist1
   IP address:  101.1.1.1
   IPv6 address:  2000:56:1::1:1:1
   Metric:  10       IP-Extended 191.56.1.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.25.1.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.1.5.1/32        Up: 0 Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:56:1:0:0:1::/96   Up: 0  Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:25:1:0:0:1::/96   Up: 0  Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:1:1::5:1/128  Up: 0  Subtlv: 0
   Metric:  10       IS (MT-IPv6) Dist2.00
   Metric:  10       IS (MT-IPv6) Edge2.00
   Metric:  10       IS-Extended Dist2.00
   Metric:  10       IS-Extended Edge2.00
LSPID                                 Seq Num     Checksum  Holdtime  ATT/P/OL
Core2.00-00                           0x00000049  0xee12    1174      0/0/0
   Area Address:  00.0000
   NLPID:  IPv6 IP
   Topology:  IPv6(Ovld:0 Att:0) IPv4
   Hostname:  Core2
   IP address:  102.1.1.2
   IPv6 address:  2000:28:1::1:1:2
   Metric:  10       IP-Extended 191.28.1.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.68.1.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.1.8.1/32        Up: 0 Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:28:1:0:0:1::/96   Up: 0  Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:68:1:0:0:1::/96   Up: 0  Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:1:1::8:1/128  Up: 0  Subtlv: 0
```

```
   Metric:  10       IS (MT-IPv6) Dist2.12
   Metric:  10       IS (MT-IPv6) Core2.3c
   Metric:  10       IS (MT-IPv6) Core2.3d
   Metric:  10       IS (MT-IPv6) Edge2.00
   Metric:  10       IS-Extended Dist2.12
   Metric:  10       IS-Extended Core2.3c
   Metric:  10       IS-Extended Core2.3d
   Metric:  10       IS-Extended Edge2.00
LSPID                             Seq Num     Checksum  Holdtime  ATT/P/OL
Edge2.00-00*                      0x00000190  0x88a9    1080      0/0/0
   Area Address:  00.0000
   NLPID:  IPv6 IP
   Topology:  IPv6(Ovld:0 Att:0) IPv4
   Hostname:  Edge2
   IP address:  101.1.1.2
   IPv6 address:  2000:28:1::1:1:1
   Metric:  10       IP-Extended 101.1.0.0/16        Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.1.2.1/32        Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.28.1.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.25.1.0/24       Up: 0 Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:1:1::2:1/128  Up: 0  Subtlv: 0
   Metric:  10       IS-Extended Dist1.00
LSPID                             Seq Num     Checksum  Holdtime  ATT/P/OL
Dist2.00-00                       0x000001b5  0x4780    1191      0/0/0
   Area Address:  00.0000
   NLPID:  IPv6 IP
   Topology:  IPv6(Ovld:0 Att:0) IPv4
   Hostname:  Dist2
   IP address:  102.1.1.1
   IPv6 address:  2000:56:1::1:1:2
   Metric:  10       IP-Extended 191.56.1.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 192.68.1.0/31       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.68.1.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.69.1.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.1.6.1/32        Up: 0 Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:56:1:0:0:1::/96   Up: 0  Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:68:1:0:0:1::/96   Up: 0  Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:1:1::6:1/128  Up: 0  Subtlv: 0
   Metric:  10       IS (MT-IPv6) Dist2.12
   Metric:  10       IS (MT-IPv6) Dist2.45
   Metric:  10       IS (MT-IPv6) Dist2.46
   Metric:  10       IS (MT-IPv6) Dist1.00
   Metric:  10       IS-Extended Dist2.12
   Metric:  10       IS-Extended Dist2.3a
   Metric:  10       IS-Extended Dist2.45
   Metric:  10       IS-Extended Dist2.46
   Metric:  10       IS-Extended Dist1.00
LSPID                             Seq Num     Checksum  Holdtime  ATT/P/OL
Dist2.12-00                       0x00000008  0xea06    1190      0/0/0
   Metric:  0        IS-Extended Dist2.00
   Metric:  0        IS-Extended Core2.00
```

The following example shows the output for the **show ipv6 isis database detail** command with transition support enabled.

```
device# show ipv6 isis database detail
IS-IS Level-1 Link State Database
LSPID                             Seq Num     Checksum  Holdtime  ATT/P/OL
Dist1.00-00                       0x000001b3  0x8019    1066      0/0/0
   Area Address:  00.0000
   NLPID:  IPv6 IP
   Topology:  IPv6(Ovld:0 Att:0) IPv4
   Hostname:  Dist1
   IP address:  101.1.1.1
   Metric:  10       IP-Extended 191.56.1.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.25.1.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.1.5.1/32        Up: 0 Subtlv: 0
   Metric:  10       IPv6 Reachablity 2000:56:1:0:0:1::/96   Up: 0  Subtlv: 0
   Metric:  10       IPv6 Reachablity 2000:25:1:0:0:1::/96   Up: 0  Subtlv: 0
   Metric:  10       IPv6 Reachablity 2000:1:1::5:1/128  Up: 0  Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:56:1:0:0:1::/96   Up: 0  Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:25:1:0:0:1::/96   Up: 0  Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:1:1::5:1/128  Up: 0  Subtlv: 0
   Metric:  10       IS (MT-IPv6) Dist2.00
   Metric:  10       IS (MT-IPv6) Edge2.00
   Metric:  10       IS-Extended Dist2.00
   Metric:  10       IS-Extended Edge2.00
```

```
LSPID                                 Seq Num     Checksum  Holdtime  ATT/P/OL
Core2.00-00                           0x0000004a  0x3bd1    1086      0/0/0
   Area Address:  00.0000
   NLPID:  IPv6 IP
   Topology:  IPv6(Ovld:0 Att:0) IPv4
   Hostname:  Core2
   IP address:  102.1.1.2
   IPv6 address:  2000:28:1::1:1:2
   Metric:  10       IP-Extended 191.28.1.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.68.1.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.1.8.1/32        Up: 0 Subtlv: 0
   Metric:  10       IPv6 Reachablity 2000:28:1:0:0:1::/96   Up: 0  Subtlv: 0
   Metric:  10       IPv6 Reachablity 2000:68:1:0:0:1::/96   Up: 0  Subtlv: 0
   Metric:  10       IPv6 Reachablity 2000:1:1::8:1/128  Up: 0  Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:28:1:0:0:1::/96   Up: 0  Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:68:1:0:0:1::/96   Up: 0  Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:1:1::8:1/128  Up: 0  Subtlv: 0
   Metric:  10       IS (MT-IPv6) Dist2.12
   Metric:  10       IS (MT-IPv6) Edge2.00
   Metric:  10       IS-Extended Dist2.12
   Metric:  10       IS-Extended Edge2.00
LSPID                                 Seq Num     Checksum  Holdtime  ATT/P/OL
Edge2.00-00*                          0x00000191  0xdba2    1055      0/0/0
   Area Address:  00.0000
   NLPID:  IPv6 IP
   Topology:  IPv6(Ovld:0 Att:0) IPv4
   Hostname:  Edge2
   IP address:  101.1.1.2
   IPv6 address:  2000:28:1::1:1:1
   Metric:  10       IP-Extended 101.1.0.0/16        Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.28.1.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.25.1.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.1.2.1/32        Up: 0 Subtlv: 0
   Metric:  10       IPv6 Reachablity 2000:1:1::2:1/128  Up: 0  Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:1:1::2:1/128  Up: 0  Subtlv: 0
   Metric:  10       IS-Extended Dist1.00
LSPID                                 Seq Num     Checksum  Holdtime  ATT/P/OL
Dist2.00-00                           0x000001b6  0x24b8    1100      0/0/0
   Area Address:  00.0000
   NLPID:  IPv6 IP
   Topology:  IPv6(Ovld:0 Att:0) IPv4
   Hostname:  Dist2
   IP address:  102.1.1.1
   IPv6 address:  2000:56:1::1:1:2
   Metric:  10       IP-Extended 191.56.1.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.68.1.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.69.1.0/24       Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 191.1.6.1/32        Up: 0 Subtlv: 0
   Metric:  10       IP-Extended 192.68.1.0/31       Up: 0 Subtlv: 0
   Metric:  10       IPv6 Reachablity 2000:56:1:0:0:1::/96   Up: 0  Subtlv: 0
   Metric:  10       IPv6 Reachablity 2000:68:1:0:0:1::/96   Up: 0  Subtlv: 0
   Metric:  10       IPv6 Reachablity 2000:1:1::6:1/128  Up: 0  Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:56:1:0:0:1::/96   Up: 0  Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:68:1:0:0:1::/96   Up: 0  Subtlv: 0
   Metric:  10       IPv6 (MT-IPv6) 2000:1:1::6:1/128  Up: 0  Subtlv: 0
   Metric:  10       IS (MT-IPv6) Dist2.12
   Metric:  10       IS (MT-IPv6) Dist1.00
   Metric:  10       IS-Extended Dist2.12
   Metric:  10       IS-Extended Dist2.3a
   Metric:  10       IS-Extended Dist1.00
LSPID                                 Seq Num     Checksum  Holdtime  ATT/P/OL
Dist2.12-00                           0x00000009  0xe807    1100      0/0/0
   Metric:  0        IS-Extended Dist2.00
   Metric:  0        IS-Extended Core2.00
```

Syntax: show ipv6 isis database detail [**l1** | **l2** | **level1** | **level2**]

The **l1** and **level1** options restrict the display to the level 1 LSP entries. You can use these options interchangeably.

The **l2** and **level2** options restrict the display to the level 2 LSP entries. You can use these options interchangeably.

For example, to display details about level 1 LSPs only, enter the following command at any CLI level.

```
device# show ipv6 isis database detail l1
```

Table 140 describes the output parameters of the **show ipv6 isis database detail** command.

TABLE 140 Output parameters of the **show ipv6 isis database detail** command

Field	Description
LSPID	The LSP ID, which consists of the source ID (HHHH.HHHH.HHHH), the pseudonode (HH-), and the LSPID (-HH). **NOTE** An asterisk (*) at the end of the address indicates that the LSP is locally originated.
Seq Num	The sequence number of the LSP.
Checksum	The checksum calculated by the device that sent the LSP and used by the device to verify that the LSP was not corrupted during transmission over the network.
Holdtime	The maximum number of seconds during which the LSP remains valid. **NOTE** The IS that originates the LSP starts the timer for the LSP. As a result, all LSPs do not have the same amount of time remaining when they enter the device LSP database.
ATT/P/OL	A 4-bit value extracted from 4 through 7 bits in the **Attach** field of the LSP.
Area Address	The address of the area.
TLVs	The remaining output displays the type, length, and value (TLV) parameters included in the LSPs. These parameters advertise reachability to IPv6 devices or networks. For example: • A router identified as an IS and with its host name can be reached using the default metric. • An end system within the current area identified as an IP-Extended and with the IP address, can be reached using the default metric. • An IPv6 prefix is up and can be reached using the default metric.
NLPID	The Network Layer Protocol Identifier (NLPID), which specifies the protocol the IS that sent the LSP is using. Usually, this value is "cc" but can also be "iso".
Hostname	The host name of the router that contains the LSP database is displayed.
IPv6 address	The IPv6 address of the interface that sent the LSP. The device can use this address as the next hop in routes to the addresses listed in the following rows.
IP address	The IP address of the interface that sent the LSP. The device can use this address as the next hop in routes to the addresses listed in the following rows.

TABLE 140 Output parameters of the **show ipv6 isis database detail** command (Continued)

Field	Description
Destination addresses	The rows of information following the IP address row are the destinations advertised by the LSP. The device can reach these destinations by using the previously listed IP address as the next hop. Each destination entry contains the following information: • Metric - The value of the default metric, which is the IS-IS cost of using the previous IP address as the next hop to reach this destination. • Device type - The device type at the destination. The type can be one of the following: - End System - The device is an ES. - IP-Internal - The device is an ES within the current area. The IP address and subnet mask are listed. - IS - The device is another IS. The NET (NSAP address) is listed. - IP-Extended - Same as IP-Internal, except the device uses the extended TLV fields described in draft-ietf-isis-traffic-02.txt to carry the information. - IS-Extended - Same as IS, except the device uses the extended TLV fields described in draft-ietf-isis-traffic-02.txt to carry the information. - MT-IPv6 - The device uses the IPv6 Multi-Topology TLV fields to carry the information.
Topology	The topology of the interface.

Displaying the system ID to name mappings

IS-IS maps the IS-IS system IDs to the hostnames of the devices with those IS. To display these mappings, enter the following command at any level of the CLI.

```
device# show ipv6 isis hostname
Total number of entries in IS-IS Hostname Table: 2
   System ID       Hostname            * = local IS
 * 0000.0022.2222  Router2
   0000.0011.1111  Router1
```

Syntax: show ipv6 isis hostname

This example contains two mappings for this device. The device's IS-IS system ID is "2222.2222.2222"and its hostname is "Router2". The display contains an entry for another router. The display contains one entry for each IS that supports name mapping.

NOTE
Name mapping is enabled by default. When name mapping is enabled, the output of the **show ipv6 isis database** and **show ipv6 isis neighbor** commands uses the hostname instead of the system ID. To disable mapping so that these displays use the system ID instead, enter the **no hostname** command at the IS-IS router configuration level.

Displaying IPv6 IS-IS interface information

To display information about the interfaces on which IPv6 IS-IS is enabled, enter the following command at any level of the CLI.

```
device# show ipv6 isis interface
```

```
Total number of IS-IS Interfaces: 4
Interface : 2/1     Local Circuit Number: 00000001
   Circuit Type : BCAST Circuit Mode : LEVEL-1-2
   Circuit State: UP Passive State: FALSE
   MTU : 1497
   Level-1 Metric: 10, Level-1 Priority: 64
   Level-1 Hello Interval: 10 Level-1 Hello Multiplier: 3
   Level-1 Designated IS: Router2.01-22         Level-1 DIS Changes: 8
   Level-2 Metric: 10, Priority: 64
   Level-2 Hello Interval: 10 Level-2 Hello Multiplier: 3
   Level-2 Designated IS: Router2.01-00 Level-2 DIS Changes: 8
   Next IS-IS LAN Level-1 Hello in 1 seconds
   Next IS-IS LAN Level-2 Hello in 1 seconds
   Number of active Level-1 adjacencies: 1
   Number of active Level-2 adjacencies: 1

  Circuit State Changes: 0 Circuit Adjacencies State Changes: 2
   Rejected Adjacencies: 0
   Circuit Authentication Fails: 0 Bad LSP 0
   Control Messages Sent: 1696 Control Messages Received: 159
   IP Enabled: TRUE
   IP Address and Subnet Mask:

    10.0.0.2           255.0.0.0
     10.147.201.150    255.255.255.0

 IPv6 Enabled: TRUE
   IPv6 Address :
     2001:db8::2
. . .
```

NOTE
The latter part of this display is truncated for brevity. The purpose of this display is to show all possible fields that might display rather than to show complete output.

Syntax: show ipv6 isis interface

This display shows the following information.

TABLE 141 IPv6 IS-IS interface information

This field...	Displays...
Total number of IS-IS interfaces	The number of interfaces on which IPv6 IS-IS is enabled.
Interface	The port or virtual interface number to which the information listed below applies.
Local Circuit Number	The ID that the instance of IPv6 IS-IS running on the interface applied to the circuit between this interface and the interface at the other end of the link.
Circuit Type	The type of IS-IS circuit running on the interface. The circuit type can be one of the following: • BCAST- broadcast • PTP - point-to-point
Circuit Mode	The IS-IS type in use on the circuit. The mode can be one of the following: • LEVEL-1 • LEVEL-2 • LEVEL-1-2

TABLE 141 IPv6 IS-IS interface information (Continued)

This field...	Displays...
Circuit State	The state of the circuit, which can be one of the following: • DOWN • UP
Passive State	The state of the passive option, which determines whether the interface is allowed to form an IS-IS adjacency with the IS at the other end of the circuit. The state can be one of the following: • FALSE - The passive option is disabled. The interface can form an adjacency with the IS at the other end of the link. • TRUE - The passive option is enabled. The interface cannot form an adjacency, but can still advertise itself into the area.
MTU	The maximum length supported for IS-IS PDUs sent on this interface.
Level-1 Metric	The default-metric value that the device inserts in IS-IS Level-1 PDUs originated on this interface.
Level-1 Priority	The priority of this IS to be elected as the Designated IS for Level-1 in this broadcast network.
Level-1 Hello Interval	The number of seconds the software waits between sending Level-1 hello PDUs to the IS at the other end of the circuit.
Level-1 Hello Multiplier	The number by which the software multiplies the hello interval to calculate the hold time for Level-1 Hello messages received on the circuit.
Level-1 Designated IS	The NET of the Level-1 Designated IS.
Level-1 DIS Changes	The number of times the NET of the Level-1 Designated IS has changed.
Level-2 Metric	The default-metric value that the device inserts in IS-IS Level-2 PDUs originated on this interface.
Level-2 Priority	The priority of this IS to be elected as the Designated IS for Level-2 in this broadcast network.
Level-2 Hello Interval	The number of seconds the software waits between sending Level-2 Hello messages to the IS at the other end of the circuit.
Level-2 Hello Multiplier	The number by which the software multiplies the hello interval to calculate the hold time for Level-2 LSPs received on the circuit.
Level-2 Designated IS	The NET of the Level-2 Designated IS.
Level-2 DIS Changes	The number of times the NET of the Level-2 Designated IS has changed.
Next IS-IS LAN Level-1 Hello	Number of seconds before next Level-1 Hello message will be transmitted by the device.

TABLE 141 IPv6 IS-IS interface information (Continued)

This field...	Displays...
Next IS-IS LAN Level-2 Hello	Number of seconds before next Level-2 Hello message will be transmitted by the device.
Number of active Level-1 adjacencies	The number of ISs with which this interface has an active Level-1 adjacency.
Number of active Level-2 adjacencies	The number of ISs with which this interface has an active Level-2 adjacency.
Circuit State Changes	The number of times the state of the circuit has changed.
Circuit State Adjacencies Changes	The number of times an adjacency has started or ended on this circuit.
Rejected Adjacencies	The number of adjacency attempts by other ISs rejected by the device.
Circuit Authentication Fails	The number of times the device rejected a circuit because the authentication did not match the authentication configured on the device.
Bad LSP	The number of times the interface received a bad LSP from an IS at the other end of the circuit. The following conditions can cause an LSP to be bad: • Invalid checksum • Invalid length • Invalid lifetime value
Control Messages Sent	The number of IS-IS control PDUs sent on this interface.
Control Messages Received	The number of IS-IS control PDUs received on this interface.
IP Enabled	The state of IP on the interface, which can be one of the following: • TRUE - IP is enabled. • FALSE - IP is disabled.
IP Address and Subnet Mask	The IP address(es) and sub-net masks configured on this interface.
IPv6 Enabled	The state of IPv6 on the interface, which can be one of the following: • TRUE - IPv6 is enabled. • FALSE - IPv6 is disabled.
IPv6 Address	The IPv6 address(es) configured on this interface.

Displaying IPv6 IS-IS memory usage

To display information about IPv6 IS-IS memory usage, enter the following command at any level of the CLI.

```
device# show ipv6 isis memory
Total Static Memory Allocated : 1333 bytes
Total Dynamic Memory Allocated : 157952 bytes
Memory Type               Size       Allocated  Max-alloc  Alloc-Fails
MTYPE_ISIS_IP6_SUMMARY_PR 0          0          0          0
MTYPE_ISIS_OTHER          20         0          1          0
MTYPE_ISIS_IP6_ROUTE_NODE 21         22         1024       0
MTYPE_ISIS_IP6_ROUTE_INFO 12         17         1024       0
MTYPE_ISIS_IP6_NEXTHOP    24         2          256        0
MTYPE_ISIS_IP6_REDIS_ROUT 12         5          256        0
```

Syntax: show ipv6 isis memory

This display shows the following information.

TABLE 142 IPv6 IS-IS memory usage information

This field...	Displays...
Total Static Memory Allocated	A summary of the amount of static memory allocated, in bytes, to IPv6 IS-IS.
Total Dynamic Memory Allocated	A summary of the amount of dynamic memory allocated, in bytes, to IPv6 IS-IS.
Memory Type	The type of memory used by IPv6 IS-IS. (This information is for use by Brocade technical support in case of a problem.)
Size	The size of a memory type.
Allocated	The amount of memory currently allocated to a memory type.
Max-alloc	The maximum amount of memory that was allocated to a memory type.
Alloc-Fails	The number of times an attempt to allocate memory to a memory type failed.

Displaying IPv6 IS-IS neighbor information

You can display a summary or detailed information for all neighbors with which the device has formed an IS-IS adjacency.

To display a summary of all IPv6 IS-IS neighbors of a device, enter the following command at any level of the CLI.

```
device# show ipv6 isis neighbor
Total number of IS-IS Neighbors: 2
System Id      Interface   SNPA           State Holdtime Type Pri StateChgeTime
Router1        Ether 3/2   0000.0000.0020 UP    30       ISL2 64  0  :0 :14:1
Router1        Ether 3/2   0000.0000.0020 UP    30       ISL1 64  0  :0 :14:1
```

Syntax: show ipv6 isis neighbor [**detail**]

This display shows the following information.

TABLE 143 Summary of IPv6 IS-IS neighbor information

This field...	Displays...
Total number of IS-IS Neighbors	The number of ISs with which the device has formed an IS-IS adjacency.
System ID	The system ID of the neighbor. **Note:** Name mapping is enabled by default. When name mapping is enabled, the output of the **show ipv6 isis neighbor** command uses the hostname instead of the system ID. To disable mapping so that these displays use the system ID instead, enter the **no hostname** command at the IS-IS router configuration level. For more information about performing this task, refer to the "Displaying IPv6 IS-IS neighbor information".
Interface	The device port or virtual interface attached to the neighbor.
SNPA	The Subnetwork Point of Attachment (SNPA), which is the MAC address of the device physical or virtual interface attached to the neighbor.
State	The state of the adjacency with the neighbor. The state can be one of the following: • DOWN - The adjacency is down. • INIT - The adjacency is being established and is not up yet. • UP - The adjacency is up.
Holdtime	The time between transmissions of IS-IS hello messages.
Type	The IS-IS type of the adjacency. The type can be one of the following: • ISL1 - Level-1 IS • ISL2 - Level-2 IS • PTP - Point-to-Point IS • ES - ES **Note** : The device forms a separate adjacency for each IS-IS type. Thus, if the device has both types of IS-IS adjacencies with the neighbor, the display contains a separate row of information for each adjacency.
Pri	The priority of this IS to be elected as the Designated IS in this broadcast network.
StateChgeTime	The amount of time that has passed since the adjacency last changed state.

To display detailed information about all IPv6 IS-IS neighbors of a device, enter the following command at any level of the CLI.

```
device# show ipv6 isis neighbor detail
Total number of IS-IS Neighbors: 2
System ID      Interface   SNPA           State Holdtime Type Pri StateChgeTime
Router1        Ether 3/2   0000.0000.0020 UP    30       ISL2 64  0  :0 :14:5
Area Address(es): 49.6561
IP Address(es): 10.0.0.1
IPv6 Address: fe80::2e0:52ff:fe00:20
Circuit ID: 10.2222.2222.01
System ID      Interface   SNPA           State Holdtime Type Pri StateChgeTime
Router1        Ether 3/2   0000.0000.0020 UP    30       ISL1 64  0  :0 :14:5
Area Address(es): 49.6561
IP Address(es): 10.0.0.1
```

```
IPv6 Address: fe80::2e0:52ff:fe00:20
Circuit ID: 10.2222.2222.01
```

This display shows the following information.

TABLE 144 Detailed IPv6 IS-IS neighbor information

This field...	Displays...
Total number of IS-IS Neighbors	For information about this field, refer to Table 143 .
System ID	For information about this field, refer to Table 143 .
Interface	For information about this field, refer to Table 143 .
SNPA	For information about this field, refer to Table 143 .
State	For information about this field, refer to Table 143 .
Holdtime	For information about this field, refer to Table 143 .
Type	For information about this field, refer to Table 143 .
Pri	For information about this field, refer to Table 143 .
StateChgeTime	For information about this field, refer to Table 143 .
Area Address(es)	The address(es) of areas to which the neighbor interface belongs.
IP Address(es)	The IP address(es) assigned to the neighbor interface.
IPv6 Address	The IPv6 address(es) assigned to the neighbor interface.
Circuit ID	The ID of the IS-IS circuit running on the neighbor interface.

Displaying IPv6 IS-IS redistribution information

To display information about the IPv6 routes redistributed into IPv6 IS-IS, enter the following command at any level of the CLI.

```
device# show ipv6 isis redistributed-routes
Prefix                                  Protocol  Level        Metric
2001:db8:1002::/48                      Static    Level-2       1
2001:db8:2002::/48                      Static    Level-2       1
2001:db8:3002::/48                      Static    Level-2       1
2001:db8:4002::/48                      Static    Level-2       1
2001:db8:5002::/48                      Static    Level-2       1
```

Syntax: show ipv6 isis redistributed-routes

This display shows the following information.

TABLE 145 IPv6 IS-IS redistribution information

This field...	Displays...
Prefix	The IPv6 routes redistributed into IPv6 IS-IS.
Protocol	The protocol from which the route is redistributed into IPv6 IS-IS. Possible protocols include the following: • BGP - BGP4+. • RIP - RIPng. • OSPF - OSPFv3. • Static - IPv6 static route table. • Connected - A directly connected network.
Level	The IS-IS level into which a route is redistributed. Possible levels include the following: • Level-1 • Level-2 • Level-1-2
Metric	The value of the default redistribution metric, which is the IS-IS cost of redistributing the route into IPv6 IS-IS.

Displaying the IPv6 IS-IS route information

To display the routes in the device's IPv6 IS-IS route table, enter the following command at any level of the CLI.

```
device# show ipv6 isis routes
ISIS IPv6 Routing Table
Total Routes: 17  Level1: 17 Level2: 0 Equal-cost multi-path: 0
Type IPv6 Prefix                Next Hop Router           Interface  Cost
L1   2001:db8:1:1000::/48       fe80::2e0:52ff:fe00:20    ethe 3/2   20
L1   2001:db8:1:2000::/48       fe80::2e0:52ff:fe00:20    ethe 3/2   20
L1   2001:db8:1:3000::/48       fe80::2e0:52ff:fe00:20    ethe 3/2   20
L1   2001:db8:1:4000::/48       fe80::2e0:52ff:fe00:20    ethe 3/2   20
L1   2001:db8:1:5000::/48       fe80::2e0:52ff:fe00:20    ethe 3/2   20
L1   2001:db8:2:1000::/48       fe80::2e0:52ff:fe00:20    ethe 3/2   30
L1   2001:db8:2:2000::/48       fe80::2e0:52ff:fe00:20    ethe 3/2   30
L1   2001:db8:2:3000::/48       fe80::2e0:52ff:fe00:20    ethe 3/2   30
L1   2001:db8:2:4000::/48       fe80::2e0:52ff:fe00:20    ethe 3/2   30
```

Syntax: show ipv6 isis routes

This display shows the following information.

TABLE 146 IPv6 IS-IS route information

This field...	Displays...
Total Routes	The total number of routes in the device's IPv6 IS-IS route table. The total includes Level-1 and Level-2 routes.
Level1	The total number of Level-1 routes in the IPv6 IS-IS route table.
Level2	The total number of Level-1 routes in the IPv6 IS-IS route table.

TABLE 146 IPv6 IS-IS route information (Continued)

This field...	Displays...
Equal-cost multi-path	The total number of equal-cost routes to the same destination in the IPv6 IS-IS route table. If load sharing is enabled, the device equally distributes traffic among the routes.
Type	The route type, which can be one of the following: • L1 - Level-1 route • L2 - Level-2 route
IPv6 Prefix	The IPv6 prefix of the route.
Next Hop Router	The IPv6 address of the next-hop interface to the destination.
Interface	The device interface (physical or virtual interface) attached to the next hop.
Cost	The IPv6 IS-IS default metric for the route, which is the cost of using this route to reach the next-hop router to this destination.

Displaying IPv6 IS-IS traffic statistics

The device maintains statistics for common IS-IS PDU types. To display the IPv6 traffic statistics, enter the following command at any level of the CLI.

```
device# show ipv6 isis traffic
                          Message Received     Message Sent
 Level-1 Hellos             98                   1171
 Level-2 Hellos             96                   1170
 PTP Hellos                 0                    0
 Level-1 LSP                3                    6
 Level-2 LSP                3                    6
 Level-1 CSNP               1                    110
 Level-2 CSNP               1                    110
 Level-1 PSNP               0                    0
 Level-2 PSNP               0                    0
```

Syntax: show ipv6 isis traffic

This display shows the following information.

TABLE 147 IPv6 IS-IS traffic statistics

This field...	Displays...
Level-1 Hellos	The number of Level-1 hello PDUs sent and received by the device.
Level-2 Hellos	The number of Level-2 hello PDUs sent and received by the device.
PTP Hellos	The number of point-to-point hello PDUs sent and received by the device.
Level-1 LSP	The number of Level-1 link-state PDUs sent and received by the device.
Level-2 LSP	The number of Level-2 link-state PDUs sent and received by the device.

TABLE 147 IPv6 IS-IS traffic statistics (Continued)

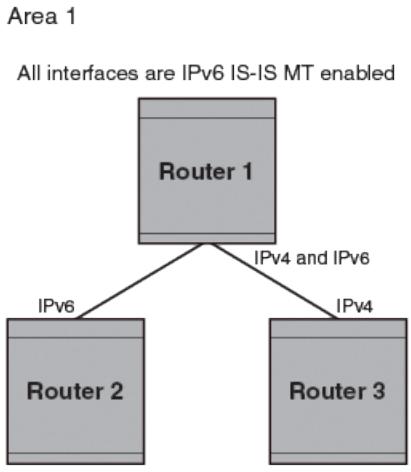
This field...	Displays...
Level-1 CSNP	The number of Level-1 Complete Sequence Number PDUs (CSNPs) sent and received by the device.
Level-2 CSNP	The number of Level-2 CSNPs sent and received by the device.
Level-1 PSNP	The number of Level-1 Partial Sequence Number PDUs (PSNPs) sent and received by the device.
Level-2 PSNP	The number of Level-2 PSNPs sent and received by the device.

IPv6 IS-IS Multi-Topology

IPv6 IS-IS supports Multi-Topology (MT) mode, which allows you to configure both IPv4 and IPv6 topologies on the router interfaces in an area or a domain. However, when implementing an MT, all routers in an area (Level 1 routing) or a domain (Level 2 routing) can be configured with a set of independent topologies on all their interfaces, even on loopback interfaces. All routers in an area or a domain use the same type of IPv6 support, either single-topology or MT. In a network, the Shortest Path First (SPF) is calculated for each configured topology.

Figure 36 depicts a non-congruent topology with IPv6 IS-IS MT enabled. Router 1 is an IPv4 and IPv6 dual stack router, Router 2 is an IPv6 router, and Router 3 is an IPv4 router. All the routers (Router 1, Router 2, and Router 3) in the Area 1 are configured with a set of independent topologies.

FIGURE 36 IS-IS non-congruent topology



Configuration considerations for IPv6 IS-IS MT

The following are the configuration considerations:

- The wide metric style must be configured before enabling IPv6 IS-IS MT.
- IPv4, IPv6, or IPv4 and IPv6 configured on the same interface must run on the same IS-IS level.

- Enabling or disabling IPv6 IS-IS MT clears all adjacencies, LSP databases, and IPv6 IS-IS routes.
- All routers on a point-to-point or a broadcast interface must support at least one common topology (IPv4 or IPv6), when MT is enabled.

Migrating to IPv6 IS-IS MT

The following steps must be performed to migrate from a non-MT environment to an MT environment.

1. Assume that the entire network is not an IPv6 IS-IS MT environment, and ensure that all the routes are correct.
2. Use the **multi-topology transition** command to enable transition mode on each router one by one, and ensure that all the routes are correct.
3. After all the routers are in transition mode, use the **no multi-topology transition** command to disable transition mode on each router one by one. Ensure that all the routes are correct.
4. Change the topology to make IPv4 and IPv6 different.

Maintaining MT adjacencies

With the extension of IPv6 IS-IS MT, the new type, length, and value (TLV) parameters are added into the IS to IS hello (IIH) packets that advertise the topologies of the interface. In IPv6 IS-IS MT, the router advertises its information using the new TLV parameters such as MT ID TLV, MT IS Reachability TLV, MT Reachable IPv4 TLV, and MT Reachable IPv6 TLV. The TLVs specify the types of data, the maximum length of the data, and the valid values for the data.

Forming adjacencies on the point-to-point interfaces

On a point-to-point interface, adjacencies are formed with IS-IS routers that do not implement MT extensions. If two peers share at least one common topology, then an adjacency is formed between the peers.

Forming adjacencies on the broadcast interfaces

On a broadcast interface, all the MT-enabled routers advertise their MT capability TLV in their IIH packets. The MT-enabled IS-IS routers form adjacency with any IS-IS routers whether or not MT is enabled. A peering MT-disabled IS-IS router does not form adjacency when NLPID TLVs do not match.

New TLV attributes

The new TLV parameters to support the IPv6 IS-IS MT extension are MT ID TLV, MT IS Reachability TLV, MT Reachable IPv4 TLV, and MT Reachable IPv6 TLV.

Enabling IPv6 IS-IS MT

When you enable IPv6 IS-IS MT in an area or a domain, the MT-enabled router runs IPv6 IS-IS in multi SPF mode. You can enable IPv6 IS-IS MT transition mode in an area or a domain using the **transition** option of the **multi-topology** command. The **transition** option allows the network operating in IPv6 IS-IS single-topology support mode to continue to work while upgrading routers to include IPv6 IS-IS MT support. When you enable transition mode, the router advertises both the single-topology TLVs and MT TLVs.

When transition mode is not enabled, the routers operating in single-topology mode do not establish IPv6 connectivity with the routers operating in MT mode.

To enable IPv6 IS-IS MT, enter the following command at the IPv6 unicast address family configuration level.

```
device(config-isis-router-ipv6u)# multi-topology
```

Syntax: [no] multi-topology

The**no** form of the command disables IPv6 IS-IS MT.

To enable IPv6 IS-IS MT with transition support, enter the following command at the IPv6 unicast address family configuration level.

```
device(config-isis-router-ipv6u)# multi-topology transition
```

Syntax: [no] multi-topology [**transition**]

The **transition** option allows the network to undergo transition from IPv6 IS-IS single-topology mode to IPv6 IS-IS MT mode. By default, the transition mode is off.

The**no** form of the command disables the transition support.

Configuring the IS-IS IPv6 PSPF exponential back-off feature

The exponential back-off mechanism allows you to schedule PSPF processing for IPv6 IS-IS MT. An initial-hold-time interval is the wait time after an LSP change until the first PSPF calculation. Optionally, this value is followed by another configurable variable called the exponential-hold-time interval that is used as a wait time between the first and second PSPF calculations.

The exponential-hold-time interval is increased in multiples of two until it reaches the maximum hold time as configured by the max-hold-time variable. Once reached, the maximum hold time remains the hold interval between PSPF calculations until there are no further changes in the network. When there are no network changes in a hold down period, the gap between PSPF calculations returns to the initial-hold-time interval and the process begins again.

If the initial-hold-time interval is configured without an exponential-hold-time, the max-hold-time variable is used for the second and all subsequent intervals.

If the initial-hold-time and exponential-hold-time intervals are not configured, the max-hold-time variable is used for the first and all subsequent intervals.

To configure the minimum time between the two consecutive partial route calculations for IPv6 IS-IS MT, enter the following command under the IPv6 unicast address family configuration level.

```
device(config-isis-router-ipv6u)# partial-spf-interval 60 1000 5000
```

Syntax: [no] partial-spf-interval *max-hold-time initial-hold-time exponential-hold-time*

The *max-hold-time* variable specifies the maximum hold time between two Partial Shortest Path First (PSPF) calculations. The range is from 0 through 120000 milliseconds. The default value is 5000 milliseconds.

The *initial-hold-time* variable is an optional value that specifies the hold time after an LSP change until the first PSPF calculation. The range is from 0 through 120000 milliseconds. The default value is 2000 milliseconds.

The *exponential-hold-time* variable is an optional value that specifies the hold time between the first and second PSPF calculations. The range is from 0 through 120000 milliseconds. The default value is 5000 milliseconds.

The**no** form of the command resets all parameters to their default values.

Changing the SPF timer

You can configure the minimum time between two consecutive SPF computations for IPv6 IS-IS MT, by entering the following command under the IPv6 unicast address family configuration level.

```
device(config-isis-router-ipv6u)# spf-interval 5 2000 2000
```

Syntax: [no] spf-interval *max-hold-time initial-hold-time exponential-hold-time*

The *max-hold-time* variable specifies the maximum time gap between consecutive SPF calculations. The range is from 0 through 120 seconds. The default value is five seconds.

The *initial-hold-time* variable specifies the initial time gap between an SPF event and the first running of SPF. The range is from 0 through 120000 milliseconds. The default value is 5000 milliseconds.

The *exponential-hold-time* variable specifies the interval between two SPF calculations. The range is from 0 through 120000 milliseconds. The default value is 5000 milliseconds.

The**no** form of the command resets all parameters to their default values.

Changing the metric added value

When the device calculates a route, the device adds a metric (cost) to the route. Each IS-IS interface has a separate metric value. In IPv6 IS-IS MT, different metrics are configured on an interface for IPv4 and IPv6. When the metric value is configured for an interface, it rebuilds the route LSP and triggers IPv6 IS-IS MT SPF calculation.

To configure the metric value for an interface under IPv6 IS-IS MT, enter the following command.

```
device(config-if-e10000-1/4)# isis ipv6 metric 15
```

Syntax: [no] isis ipv6 metric *value* [**level-1** | **level-2**]

The *value* variable specifies the metric. The metric range depends on the metric style. You can specify the range from 1 through 63 for the narrow metric style and from 1 through 16777215 for the wide metric style. The default value for both styles is 10.

The **level-1** option specifies that the level 1 router routes traffic only within an area. To forward traffic to another area, the level 1 router sends the traffic to its nearest level-2 router.

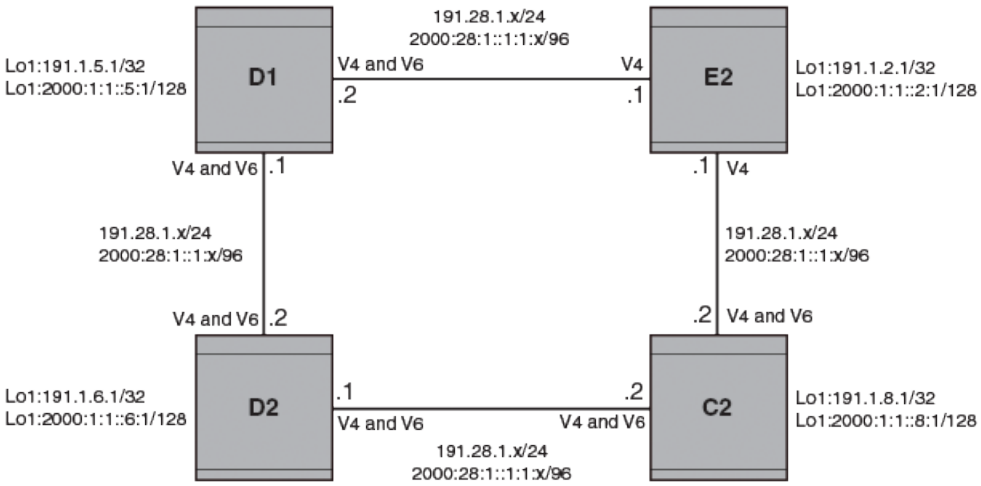
The **level-2** option specifies that the level 2 router routes traffic between the areas within a domain. The **level-1** | **level-2** options apply the change to only the level you specify. If you do not use one of the options, the change applies to both the levels.

The **no** form of the command resets all parameters to their default values.

Configuration example to deploy IPv6 IS-IS MT

Figure 37 shows an example of a non-congruent topology enabled with IPv6 IS-IS MT. Router D1 supports both the IPv4 and IPv6 topologies, router D2 supports both the IPv4 and IPv6 topologies, router E2 supports an IPv4 topology, and router C2 supports both the IPv4 and IPv6 topologies.

FIGURE 37 IPv6 IS-IS MT configuration



Configuration commands to enable IPv6 IS-IS MT on router D1

The following commands enable IPv6 IS-IS MT on router D1.

```
device(config)# router isis
device(config-isis-router)# net 00.0000.001b.ed03.1400.00
device(config-isis-router)# address-family ipv4 unicast
device(config-isis-router-ipv4u)# metric-style wide
device(config-isis-router-ipv4u)# exit-address-family
device(config-isis-router)# address-family ipv6 unicast
device(config-isis-router-ipv6u)# multi-topology
device(config-isis-router-ipv6u)# exit-address-family
```

Configuration commands to enable IPv6 IS-IS MT on router D2

The following commands enable IPv6 IS-IS MT on router D2.

```
device(config)# router isis
device(config-isis-router)# net 00.0000.001b.ed04.4400.00
device(config-isis-router)# address-family ipv4 unicast
device(config-isis-router-ipv4u)# metric-style wide
device(config-isis-router-ipv4u)# exit-address-family
device(config-isis-router)# address-family ipv6 unicast
device(config-isis-router-ipv6u)# multi-topology
device(config-isis-router-ipv6u)# exit-address-family
```

Configuration commands to enable IPv6 IS-IS MT on router E2

The following commands enable IPv6 IS-IS MT on router E2.

```
device(config)# router isis
device(config-isis-router)# net 00.0000.001b.ed04.4000.00
device(config-isis-router)# address-family ipv4 unicast
device(config-isis-router-ipv4u)# metric-style wide
device(config-isis-router-ipv4u)# exit-address-family
device(config-isis-router)# address-family ipv6 unicast
device(config-isis-router-ipv6u)# multi-topology
device(config-isis-router- ipv6u)# exit-address-family
```

Configuration commands to enable IPv6 IS-IS MT on router C2

The following commands enable IPv6 IS-IS MT on router C2.

```
device(config)# router isis
device(config-isis-router)# net 00.0000.001b.ed04.0000.00
device(config-isis-router)# address-family ipv4 unicast
device(config-isis-router-ipv4u)# metric-style wide
device(config-isis-router-ipv4u)# exit-address-family
device(config-isis-router)# address-family ipv6 unicast
device(config-isis-router-ipv6u)# multi-topology
device(config-isis-router-ipv6u)# exit-address-family
```

To display current running configuration for the router D2, enter the following command.

```
device# show running-config
router isis
 net 00.0000.001b.ed04.4400.00
  address-family ipv4 unicast
  metric-style wide
 exit-address-family
  address-family ipv6 unicast
  multi-topology
 exit-address-family
End
```

default-link-metric

Configures the metric value globally on all active IPv6 IS-IS interfaces.

Syntax **default-link-metric** *value* [**level-1** | **level-2**]

no default-link-metric *value* [**level-1** | **level-2**]

Command Default The **default-link-metric** command is disabled by default.

Parameters

default-link-metric

Specifies the global default-link-metric parameter for an IPv6 IS-IS unicast address family configuration.

value

Specifies the default-link-metric value in metric style and configurable range. The metric style consists of narrow or wide style. The narrow metric range is from 1 - 63. The wide metric range is from 1 - 16777215. If you change the metric style configuration, the default-link-metric value will also change. The new default-link-metric value is equal to the minimum of the configured value, and the maximum value supported by the new metric style. For example, if the metric style changes from a wide metric to a narrow metric, and the default-link-metric value is greater than 63, the default-link-metric value changes to 63 because it is the maximum value supported in the narrow metric style. When the metric style changes from a narrow metric to a wide metric, there is no change to the default-link-metric value.

level-1 | level-2

Specifies the IS-IS routing parameter as level-1 or level-2. You can choose to configure the default-link-metric parameter as either level-1 or level-2. If the IS-IS routing parameter is not configured, the default-link-metric value is applied to both level-1 and level-2.

Modes IPv6 IS-IS unicast address family configuration level.

Usage Guidelines Use the **default-link metric** *value* command to change the metric value globally on all active IPv6 IS-IS interfaces. The **default-link metric** *value* command is useful when you have a common IS-IS metric value on all IS-IS interfaces, other than the default metric value of 10. The command enables the metric value for IPv6 routes per address family configuration. Use the **no** form of the command to reset the metric value to the default value 10. The **default-link metric** *value* command is not applicable to MPLS IS-IS shortcuts and tunnel interfaces.

You can change the metric value for a specific interface using the **isis metric** command or the **isis ipv6 metric** command. The **isis metric** command configuration takes precedence over the **default-link metric** *value* command configuration.

The IPv6 default link metric value is dependent upon the configuration of the **multi-topology** command under the IPv6 IS-IS unicast address family. The IPv6 default link metric value is displayed in the output of the **show isis interface** command. Consider the following when the IS-IS multi-topology feature is enabled.

IS-IS Multi-topology feature	IPv6 default link metric configuration
The **multi-topology** command is enabled.	The IPv6 default link metric is equal to the value configured under the IPv6 IS-IS unicast address family. For an example of this configuration, refer to the **multi-topology** command configuration in the Example section.

IS-IS Multi-topology feature	IPv6 default link metric configuration
The **no multi-topology** command is enabled.	The IPv6 default link metric is equal to the value configured under the IPv4 IS-IS unicast address family.
The **multi-topology transition** command is enabled.	The IPv6 default link metric is equal to the value configured under the IPv6 IS-IS unicast address family.

During switchover or hitless upgrade, the IS-IS default link metric configuration is not affected. Backward compatibility is not supported.

NOTE
The **default-link metric** *value* command is supported on the Brocade NetIron XMR Series, the Brocade MLX Series, and the Brocade NetIron CER Series and Brocade NetIron CES Series platforms.

Examples

The following example configures the IS-IS default link metric value to 20 for an IPv6 address family. The default link metric value of 20 is applied to both level-1 and level-2.

```
device(config)#router isis
device(config-isis-router)#address-family-ipv6 unicast
device(config-isis-router-ipv6u)#default-link-metric 20
device(config-isis-router-ipv6u)#
```

The following example configures the **multi-topology** command under the IPv6 IS-IS unicast address family.

```
device(config)#router isis
device(config-isis-router)#address-family-ipv6 unicast
device(config-isis-router-ipv6u)#multi-topology
device(config-isis-router-ipv6u)#default-link-metric 20
```

Use the **show isis interface** command to display the configuration for the IPv6 IS-IS default link metric value. In the output below, the IPv4 default link metric displays a different value than the IPv6 default link metric because the **multi-topology** is configured. Refer to the **multi-topology** command configuration above for more information.

```
device(config)#show isis interface
…..
Circuit State: UP Circuit Mode: LEVEL-1-2
   Circuit Type: PTP Passive State: FALSE
   Circuit Number: 1, MTU: 1500
   Auth-mode: None
   Level-1 Metric: 30
   Hello Interval: 10 Hello Multiplier: 3
   Level-2 Metric: 30
   Next IS-IS PTPT Hello in 0 seconds
   Circuit State Changes: 1 Circuit Adjacencies State Changes: 0
   Rejected Adjacencies: 0
   Circuit Authentication L1 failures: 0
   Bad LSPs: 0
   Control Messages Sent: 7 Control Messages Received: 0
   Hello Padding: Enabled
   IP Enabled: TRUE
   IPv6 Enabled: TRUE
   IPv6 Level-1 Metric: 20
   IPv6 Level-2 Metric: 20
…..
```

History

Release version	Command history
Multi-Service Ironware R05.7.00	This command was introduced.

reverse-metric

Configures the reverse metric value at the IS-IS router level.

Syntax **reverse-metric** [*value*] [**whole-lan**] [**te-def-metric**]

no reverse-metric [*value*] [**whole-lan**] [**te-def-metric**]

reverse-metric tlv-type[*value*]

no reverse-metric tlv-type[*value*]

Command Default The **reverse-metric** command is disabled by default.

Parameters

reverse-metric

Specifies the reverse metric parameter at the IS-IS router level.

value

Specifies the reverse metric value in metric style. The metric style consists of narrow or wide style. The narrow metric range is from 1 - 63. The wide metric range is from 1 - 16777215. The default value is 16777214 irrespective of the metric style configured. If the reverse-metric value is configured, the local LSP is updated with the sum of the default metric and the reverse metric value. When the IS-IS neighbor router receives the reverse metric value through the IS hello, the neighbor router updates the cost to reach the original IS-IS router with the sum of default metric and the reverse metric value. This helps in shifting traffic to the other alternate paths.

whole-lan

Specifies changing the reverse metric parameter for the entire LAN. The **whole-lan** option indicates the whole LAN bit in the flag. If the **whole-lan** option is enabled, the configured reverse metric value affects the entire LAN. If the **whole-lan** option is not enabled, the reverse metric value affects only the neighbor router. This option takes effect only on the multi-access LAN. IS-IS point-to-point interfaces are not affected when the **whole-lan** option is enabled.

te-def-metric

Specifies setting the TE default metric sub-TLV. If the **te-def-metric** option is enabled, the router sends a TE default metric sub-TLV within the reverse-metric TLV.

tlv-type *value*

Specifies the TLV type for the reverse metric parameter. The TLV type can only be configured at the IS-IS router level. The **tlv-type** *value* parameter must be configured in the range of unassigned IS-IS TLV values. The **tlv-type** *value* parameter should not be configured with existing IS-IS TLV types. The default value is 254.

Modes IS-IS router level.

Usage Guidelines Use the **reverse-metric** command when you are performing network maintenance operations, such as software upgrades, on an IS-IS router node. When maintenance operations are performed, the router undergoing maintenance should not be used by the neighbor routers to forward transit traffic. In order to shift traffic away from the router undergoing maintenance, configure the **reverse-metric** command on the maintenance router. The router undergoing maintenance first advertises a reverse metric TLV in a IS-IS hello PDU to its neighbor router on a point-to-point or multi-access link. When the neighbor router receives a high reverse metric value, the router selects alternate paths to forward traffic while

maintenance is going on. The neighbor router adds the reverse metric TLV to its own TE default metric sub-TLV and recalculates its SPF tree and route topology. The neighbor router floods the new LSP containing the extended IS reachability TLV throughout the domain. Traffic gradually shifts onto alternate paths away from the link between the maintenance router and the neighbor router as nodes in the IS-IS domain receive the new LSP. Once the maintenance is complete, you can remove the **reverse-metric** command configuration from the router, and the reverse metric TLV in the IS-IS hello PDU is no longer advertised to the neighbor router. The IS-IS neighbor router reverts back to its original IS-IS metric, and the traffic switches to the original IS-IS router to reach its destination.

In a multi-access link, the IS-IS DIS router adds the reverse metric TLV value to each node's default metric value in the pseudonode LSP when the whole-lan flag is set. All non-DIS nodes ignore the reverse metric TLV. If multiple neighbor routers advertise the reverse metric TLV with the whole LAN flag set, the neighbor router with the highest MAC address takes precedence, and the value advertised by that neighbor is updated in the pseudonode LSP for all neighbors. If some neighbor routers do not set the whole LAN flag, then the reverse metric TLV value advertised by the neighbor router is updated in the pseudonode LSP for that neighbor only.

The S flag is set when the sender of the reverse metric TLV signals to the neighbor router to use the TE sub-tlv for the default metric (sub-tlv type 18) in the reverse metric TLV. When the receiving router finds the S flag set in the reverse metric TLV, the router searches for the TE sub-tlv. The router adds the default metric value in the TE sub-tlv to the configured TE default metric value and recalculates the CSPF.

The **no** form of the command, specified with the configured value, resets the metric value to the default value of 16777214. The **no reverse-metric** command removes the entire reverse metric configuration.

NOTE
The **reverse-metric** *value* command is supported on the Brocade NetIron XMR Series, the Brocade MLX Series, and the Brocade NetIron CER Series and Brocade NetIron CES Series platforms.

Examples The following example configures the reverse metric value to 50 at the router level. The **whole-lan** option is enabled to include the entire LAN.

```
device(config)#router isis
device(config-isis-router)#reverse-metric ?
DECIMAL           Narrow metric range 1-63, Wide metric range 1-16777214,
                  Default is 16777214
te-def-metric   Update TE default metric sub-tlv
tlv-type        Configure reverse metric TLV type
whole-lan       Change metric for whole LAN
device(config-isis-router)#reverse-metric 50 ?
te-def-metric   Update TE default metric sub-tlv
whole-lan       Change metric for whole LAN
<cr>
device(config-isis-router)#reverse-metric 50 whole-lan
device(config-isis-router)#
```

The following example configures the reverse metric TLV type in the range of unassigned IS-IS TLV values.

```
device(config-isis-router)#reverse-metric tlv-type ?
DECIMAL   Configure in the range of unassigned ISIS TLV values
device(config-isis-router)#reverse-metric tlv-type 230
device(config-isis-router)#
```

Use the **show isis config** command to display the configuration of the reverse metric value at the router level. The reverse metric value and the parameters, **whole-lan** and **te-def-metric** are highlighted in the output.

```
device(config)#show isis config
    router isis
 net 49.2211.aaaa.bbbb.cccc.00
 reverse-metric 50 whole-lan te-def-metric
 address-family ipv4 unicast
 exit-address-family

 address-family ipv6 unicast
 exit-address-family
```

History

Release version	Command history
Multi-Service Ironware R05.7.00	This command was introduced.

isis reverse-metric

Configures the reserve metric value on a single IPv6 IS-IS interface.

Syntax **isis reverse-metric** [*value*] [**whole-lan**] [**te-def-metric**]

no isis reverse-metric [*value*] [**whole-lan**] [**te-def-metric**]

Command Default The **isis reverse-metric** command is disabled by default.

Parameters

isis reverse-metric

Specifies the reverse metric parameter at the interface level.

value

Specifies the reverse metric value in metric style. The metric style consists of narrow or wide style. The narrow metric range is from 1 - 63. The wide metric range is from 1 - 16777215. The default value is 16777214 irrespective of the metric style configured. If the reverse-metric value is configured, the local LSP is updated with the sum of the default metric and the reverse metric value. When the IS-IS neighbor router receives the reverse metric value through the IS hello, the neighbor router updates the cost to reach the original IS-IS router with the sum of default metric and the reverse metric value. This helps in shifting traffic to the other alternate paths.

whole-lan

Specifies changing the reverse metric parameter for the entire LAN. The **whole-lan** option indicates the whole LAN bit in the flag. If the **whole-lan** option is enabled, the configured reverse metric value affects the entire LAN. If the **whole-lan** option is not enabled, the reverse metric value affects only the neighbor router. This option takes effect only on the multi-access LAN. IS-IS point-to-point interfaces are not affected when the **whole-lan** option is enabled.

te-def-metric

Specifies setting the TE default metric sub-TLV. If the **te-def-metric** option is enabled, the router sends a TE default metric sub-TLV within the reverse-metric TLV.

Modes IPv6 IS-IS interface level.

Usage Guidelines Use the **isis reverse-metric** command when you are performing network maintenance operations, such as software upgrades, at the link level. When maintenance operations are performed, the link undergoing maintenance should not be used by the neighbor routers to forward transit traffic. In order to shift traffic away from the link undergoing maintenance, configure the **isis reverse-metric** command on the maintenance link. The router undergoing maintenance first advertises a reverse metric TLV in a IS-IS hello PDU to its neighbor router on a point-to-point or multi-access link. When the neighbor router receives a high reverse metric value, the router selects alternate paths to forward traffic while maintenance is going on. The neighbor router adds the reverse metric TLV to its own TE default metric sub-TLV and recalculates its SPF tree and route topology. The neighbor router floods the new LSP containing the extended IS reachability TLV throughout the domain. Traffic gradually shifts onto alternate paths away from the link between the maintenance router and the neighbor router as nodes in the IS-IS domain receive the new LSP. Once the maintenance is complete, you can remove the **isis reverse-metric** command configuration on the link, and the reverse metric TLV in the IS-IS hello PDU is no longer advertised to the neighbor router. The IS-IS neighbor router reverts back to its original IS-IS metric, and the traffic switches to the original IS-IS link to reach its destination.

In a multi-access link, the IS-IS DIS router adds the reverse metric TLV value to each node's default metric value in the pseudonode LSP when the whole-lan flag is set. All non-DIS nodes ignore the

reverse metric TLV. If multiple neighbor routers advertise the reverse metric TLV with the whole LAN flag set, the neighbor router with the highest MAC address takes precedence, and the value advertised by that neighbor is updated in the pseudonode LSP for all neighbors. If some neighbor routers do not set the whole LAN flag, then the reverse metric TLV value advertised by the neighbor router is updated in the pseudonode LSP for that neighbor only.

The S flag is set when the sender of the reverse metric TLV signals to the neighbor router to use the TE sub-tlv for the default metric (sub-tlv type 18) in the reverse metric TLV. When the receiving router finds the S flag set in the reverse metric TLV, the router searches for the TE sub-tlv. The router adds the default metric value in the TE sub-tlv to the configured TE default metric value and recalculates the CSPF.

The **no** form of the command, specified with the configured value, resets the metric value to the default value of 16777214. The **no isis reverse-metric** command removes the entire reverse metric configuration.

NOTE
The **isis reverse-metric** *value* command is supported on the Brocade NetIron XMR Series, the Brocade MLX Series, and the Brocade NetIron CER Series and Brocade NetIron CES Series platforms.

Examples The following example configures the reverse metric value to 70 on a single IPv6 IS-IS interface level. The **whole-lan** option is enabled to include the entire LAN.

```
device(config)#interface ethernet 1/3
device(config-if-e1000-1/3)#ipv6 address 100::1/64
device(config-if-e1000-1/3)#ipv6 router isis
device(config-if-e1000-1/3)# isis reverse-metric ?
DECIMAL           Narrow metric range 1-63, Wide metric range 1-16777214,
                  Default is 16777214
te-def-metric   Update TE default metric sub-tlv
whole-lan       Change metric for whole LAN
device(config-if-e1000-1/3)#isis reverse-metric 70 ?
te-def-metric   Update TE default metric sub-tlv
whole-lan       Change metric for whole LAN
<cr>
device(config-if-e1000-1/3)#isis reverse-metric 70 whole-lan
device(config-if-e1000-1/3)#
```

Use the **show ipv6 isis** command to display the configuration of the reverse metric value at the global level. The reverse metric value and flags are highlighted in the output.

```
device(config)#show ipv6 isis
IS-IS Routing Protocol Operation State: Enabled
 IS-Type: Level-1-2
 System ID: 1234.1234.1234
 Manual area address(es):
   49

 BFD: Disabled, BFD HoldoverInterval: 0
 NSR: Disabled
 ISIS Global Reverse Metric 63
 ISIS Global Reverse Metric Flags: W S
 LDP-SYNC: Not globally enabled
 Interfaces with IPv6 IS-IS configured:
    None
```

Use the **show ipv6 isis interface** command to display the configuration of the reverse metric value at the interface level. The output below displays the isis configuration for a specific VE interface. The reverse metric value and flags are highlighted in the output.

```
device(config)#show ipv6 isis interface ve 30
Interface: ve 30
   Circuit State: UP Circuit Mode: LEVEL-1-2
   Circuit Type: BCAST Passive State: FALSE
   Circuit Number: 1, MTU: 1500
   Level-1 Auth-mode: None
   Level-2 Auth-mode: None
   Level-1 Metric: 63, Level-1 Priority: 64
   Level-1 Hello Interval: 10 Level-1 Hello Multiplier: 3
   Level-1 Designated IS: NI-MLX-6-01 Level-1 DIS Changes: 4
   Level-2 Metric: 63, Level-2 Priority: 64
   Level-2 Hello Interval: 10 Level-2 Hello Multiplier: 3
   Level-2 Designated IS: NI-MLX-6-01 Level-2 DIS Changes: 4
   …….
            …….
            IPv6 Enabled: TRUE
   IPv6 Level-1 Metric: 63
   IPv6 Level-2 Metric: 63
   IPv6 Addresses:
     30::1/64
   IPv6 Link-Local Addresses:
     fe80::768e:f8ff:fe2a:1200
   MPLS TE Enabled: FALSE
   ISIS Reverse Metric 16777214
   ISIS Reverse Metric Flags: W S
   LDP-SYNC: Disabled, State: -
```

History

Release version	Command history
Multi-Service Ironware R05.7.00	This command was introduced.

RIPng

- RIPng feature support....663
- RIPng Overview....663
- Configuring RIPng....664
- Clearing RIPng routes from IPv6 route table....669
- Displaying RIPng information....669

RIPng feature support

The following features are documented in the *Multi-Service IronWare Routing Configuration Guide*.

Features supported	Brocade NetIron XMR Series	Brocade NetIron MLX Series	Brocade NetIron CES 2000 Series BASE package	Brocade NetIron CES 2000 Series ME_PREM package	Brocade NetIron CES 2000 Series L3_PREM package	Brocade NetIron CER 2000 Series Base package	Brocade NetIron CER 2000 Series Advanced Services package
RIPng	Yes	Yes	Yes	Yes	Yes	Yes	Yes
RIPng timers	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Route redistribution into RIPng	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Route loop prevention - poison reverse and split horizon	Yes	Yes	Yes	Yes	Yes	Yes	Yes

RIPng Overview

Routing Information Protocol (RIP) is an IP route exchange protocol that uses a distance vector (a number representing a distance) to measure the cost of a given route. RIP uses a hop count as its cost or metric.

IPv6 RIP, known as Routing Information Protocol Next Generation or RIPng , functions similarly to IPv4 RIP version 2. RIPng supports IPv6 addresses and prefixes.

In addition, some new commands that are specific to RIPng have been implemented. This chapter describes the commands that are specific to RIPng. This section does not describe commands that apply to both IPv4 RIP and RIPng.

RIPng maintains a Routing Information Database (RIB) , which is a local route table. The local RIB contains the lowest-cost IPv6 routes learned from other RIP routers. In turn, RIPng attempts to add routes from its local RIB into the main IPv6 route table.

NOTE
Brocade IPv6 devices support up to 10,000 RIPng routes. ICX 6650 IPv6 devices support up to 2000 RIPng routes.

Configuring RIPng

To configure RIPng, you must enable RIPng globally on the Brocade device and on individual device interfaces. The following configuration tasks are optional:

- Change the default settings of RIPng timers
- Configure how the Brocade device learns and advertises routes
- Configure which routes are redistributed into RIPng from other sources
- Configure how the Brocade device distributes routes through RIPng
- Configure poison reverse parameters

Enabling RIPng

Before configuring the device to run RIPng, you must do the following:

- Enable the forwarding of IPv6 traffic on the device using the **ipv6 unicast-routing** command.
- Enable IPv6 on each interface over which you plan to enable RIPng. You enable IPv6 on an interface by configuring an IPv6 address or explicitly enabling IPv6 on that interface.

By default, RIPng is disabled. To enable RIPng, you must enable it globally on the Brocade device and also on individual device interfaces.

NOTE
Enabling RIPng globally on the Brocade device does not enable it on individual device interfaces.

To enable RIPng globally, enter the following command.

```
device(config-rip-router)#ipv6 router rip
device(config-ripng-router)#
```

After you enter this command, the device enters the RIPng configuration level, where you can access several commands that allow you to configure RIPng.

Syntax: [no] ipv6 router rip

To disable RIPng globally, use the **no** form of this command.

After enabling RIPng globally, you must enable it on individual Brocade device interfaces. You can enable it on physical as well as virtual routing interfaces. For example, to enable RIPng on Ethernet interface 3/1, enter the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 rip enable
```

Syntax: [no] ipv6 rip enable

To disable RIPng on an individual device interface, use the **no** form of this command.

Enabling RIPng for a VRF instance

To enable RIPng for a specific VRF instance, enter the following commands:

```
device(config-rip-router)#ipv6 router rip vrf red
device(config-ripng-router-vrf-red)#
```

Syntax: [no] ipv6 router rip vrf*vrf-name*

vrf-name is the specified VRF name for the RIPng. If the VRF name is not specified, RIPng is configured using the default VRF.

To disable RIPng for a specific VRF instance, use the **no** form of this command.

Configuring RIPng timers

TABLE 148 RIPng timers

Timer	Description	Default
Update	Amount of time (in seconds) between RIPng routing updates.	30 seconds.
Timeout	Amount of time (in seconds) after which a route is considered unreachable.	180 seconds.
Hold-down	Amount of time (in seconds) during which information about other paths is ignored.	180 seconds.
Garbage-collection	Amount of time (in seconds) after which a route is removed from the routing table.	120 seconds.

You can adjust these timers for RIPng. Before doing so, keep the following caveats in mind:

- If you adjust these RIPng timers, Brocade strongly recommends setting the same timer values for all routers and access servers in the network.
- Setting the update timer to a shorter interval can cause the devices to spend excessive time updating the IPv6 route table.
- Brocade recommends setting the timeout timer value to at least three times the value of the update timer.
- Brocade recommends a shorter hold-down timer interval, because a longer interval can cause delays in RIPng convergence.

The following example sets updates to be advertised every 45 seconds. If a route is not heard from in 135 seconds, the route is declared unusable. Further information is suppressed for an additional 10 seconds. Assuming no updates, the route is flushed from the routing table 20 seconds after the end of the hold-down period.

```
device(config)# ipv6 router rip
device(config-ripng-router)# timers 45 135 10 20
```

Syntax: [no] timers*update-timer timeout-timer hold-down-timer garbage-collection-timer*

Possible values for the timers are as follows:

- Update timer: 3 through 65535 seconds.
- Timeout timer: 9 through 65535 seconds.

- Hold-down timer: 9 through 65535 seconds.
- Garbage-collection timer: 9 through 65535 seconds.

NOTE
You must enter a value for each timer, even if you want to retain the current setting of a particular timer.

To return to the default values of the RIPng timers, use the **no** form of this command.

Configuring route learning and advertising parameters

You can configure the following learning and advertising parameters:

- Learning and advertising of RIPng default routes.
- Advertising of IPv6 address summaries.
- Metric of routes learned and advertised on a Brocade device interface.

Configuring default route learning and advertising

By default, the device does not learn IPv6 default routes (::/0). You can originate default routes into RIPng, which causes individual Brocade device interfaces to include the default routes in their updates. When configuring the origination of the default routes, you can also do the following:

- Suppress all other routes from the updates.
- Include all other routes in the updates.

For example, to originate default routes in RIPng and suppress all other routes in updates sent from Ethernet interface 3/1, enter the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 rip default-information only
```

To originate IPv6 default routes and include all other routes in updates sent from Ethernet interface 3/1, enter the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 rip default-information originate
```

Syntax: [no] ipv6 rip default-information { **only** | **originate** }

The **only** keyword originates the default routes and suppresses all other routes from the updates.

The **originate** keyword originates the default routes and includes all other routes in the updates.

To remove the explicit default routes from RIPng and suppress advertisement of these routes, use the **no** form of this command.

Advertising IPv6 address summaries

You can configure RIPng to advertise a summary of IPv6 addresses from a Brocade device interface and to specify an IPv6 prefix that summarizes the routes.

If a route's prefix length matches the value specified in the **ipv6 rip summary-address** command, RIPng advertises the prefix specified in the **ipv6 rip summary-address** command instead of the original route.

For example, to advertise the summarized prefix 2001:db8::/36 instead of the IPv6 address 2001:db8:0:adff:8935:e838:78:e0ff with a prefix length of 64 bits from Ethernet interface 3/1, enter the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 address 2001:db8:0:adff:8935:e838:78:
e0ff /64
device(config-if-e100-3/1)# ipv6 rip summary-address 2001:db8::/36
```

Syntax: [no] ipv6 rip summary-address *ipv6-prefix*|*prefix-length*

You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the ipv6-prefix parameter and precede the prefix-length parameter.

To stop the advertising of the summarized IPv6 prefix, use the **no** form of this command.

Changing the metric of routes learned and advertised on an interface

A device interface increases the metric of an incoming RIPng route it learns by an offset (the default is one). The device then places the route in the route table. When the device sends an update, it advertises the route with the metric plus the default offset of zero in an outgoing update message.

You can change the metric offset an individual interface adds to a route learned by the interface or advertised by the interface. For example, to change the metric offset for incoming routes learned by Ethernet interface 3/1 to one and the metric offset for outgoing routes advertised by the interface to three, enter the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e100-3/1)# ipv6 rip metric-offset 1
device(config-if-e100-3/1)# ipv6 rip metric-offset out 3
```

In this example, if Ethernet interface 3/1 learns about an incoming route, it will increase the incoming metric by two (the default offset of 1 and the additional offset of 1 as specified in this example). If Ethernet interface 3/1 advertises an outgoing route, it will increase the metric by 3 as specified in this example.

Syntax: [no] ipv6 rip metric-offset [**out**] *1-16*

To return the metric offset to its default value, use the **no** form of this command.

Redistributing routes into RIPng

You can configure the Brocade device to redistribute routes from the following sources into RIPng:

- IPv6 static routes
- Directly connected IPv6 networks
- BGP4+
- IPv6 IS-IS
- OSPFv3

When you redistribute a route from BGP4+, IPv6 IS-IS, or OSPFv3 into RIPng, the device can use RIPng to advertise the route to its RIPng neighbors.

When configuring the Brocade device to redistribute routes, such as BGP4+ routes, you can optionally specify a metric for the redistributed routes. If you do not explicitly configure a metric, the default metric value of one is used.

For example, to redistribute OSPFv3 routes into RIPng, enter the following command.

```
device(config)# ipv6 router rip
device(config-ripng-router)# redistribute ospf
```

Syntax: [no] redistribute{ **bgp** | **connected** | **isis** | **ospf** | **static** [**metric** *number*] }

For the metric, specify a numerical value that is consistent with RIPng.

Controlling distribution of routes through RIPng

You can create a prefix list and then apply it to RIPng routing updates that are received or sent on a device interface. Performing this task allows you to control the distribution of routes through RIPng.

For example, to permit the inclusion of routes with the prefix 2001:db8::/32 in RIPng routing updates sent from Ethernet interface 3/1, enter the following commands.

```
device(config)# ipv6 prefix-list routesfor2001 permit 2001:db8::/32
device(config)# ipv6 router rip
device(config-ripng-router)# distribute-list prefix-list routesfor2001 out
```

To deny prefix lengths greater than 64 bits in routes that have the prefix 2001:db8::/64 and allow all other routes received on tunnel interface 3/1, enter the following commands.

```
device(config)# ipv6 prefix-list 2001routes deny 2001:db8::/64 le 128
device(config)# ipv6 prefix-list 2001routes permit ::/0 ge 0 le 128
device(config)# ipv6 router rip
device(config-ripng-router)# distribute-list prefix-list 2001routes in
```

Syntax: [no] distribute-list prefix-list *name* { **in** | **out** }

The name parameter indicates the name of the prefix list generated using the **ipv6 prefix-list** command.

The **in** keyword indicates that the prefix list is applied to incoming routing updates on the specified interface.

The **out** keyword indicates that the prefix list is applied to outgoing routing updates on the specified interface.

For the *interface* parameter, you can specify the ethernet, loopback, ve, or tunnel keywords. If you specify an Ethernet interface, also specify the port number associated with the interface. If you specify a VE or tunnel interface, also specify the VE or tunnel number.

To remove the distribution list, use the **no** form of this command.

Configuring poison reverse parameters

By default, poison reverse is disabled on a RIPng Brocade device. If poison reverse is enabled, RIPng advertises routes it learns from a particular interface over that same interface with a metric of 16, which means that the route is unreachable.

If poison reverse is enabled on the RIPng Brocadedevice, it takes precedence over split horizon (if it is also enabled).

To enable poison reverse on the RIPng Brocade device, enter the following commands.

```
device(config)# ipv6 router rip
device(config-ripng-router)# poison-reverse
```

Syntax:[no] poison-reverse

To disable poison-reverse, use the **no** form of this command.

By default, if a RIPng interface goes down, the Brocade device does not send a triggered update for the interface's IPv6 networks.

To better handle this situation, you can configure a RIPng Brocade device to send a triggered update containing the local routes of the disabled interface with an unreachable metric of 16 to the other RIPng routers in the routing domain. You can enable the sending of a triggered update by entering the following commands.

```
device(config)# ipv6 router rip
device(config-ripng-router)# poison-local-routes
```

Syntax: [no] poison-local-routes

To disable the sending of a triggered update, use the **no** form of this command.

Clearing RIPng routes from IPv6 route table

To clear all RIPng routes from the RIPng route table and the IPv6 main route table and reset the routes, enter the following command at the Privileged EXEC level or any of the Config levels of the CLI.

```
device# clear ipv6 rip route
```

Syntax: clear ipv6 rip route

Clearing RIPng for a VRF instance

To clear all RIPng routes for a specific VRF, enter the following command:

```
device# clear ipv6 rip vrf red route
```

Syntax: clear ipv6 rip vrf *vrf-name* **route**

Displaying RIPng information

You can display the following RIPng information:

- RIPng configuration
- RIPng routing table

Displaying RIPng configuration

To display RIPng configuration information, enter the **show ipv6 rip** command at any CLI level.

```
device# show ipv6 rip
IPv6 rip enabled, port 521
      Administrative distance is 120
      Updates every 30 seconds, expire after 180
      Holddown lasts 180 seconds, garbage collect after 120
      Split horizon is on; poison reverse is off
      Default routes are not generated
      Periodic updates 5022, trigger updates 10
```

```
      Distribute List, Inbound : Not set
      Distribute List, Outbound : Not set
      Redistribute: CONNECTED
```

Syntax: show ipv6 rip

TABLE 149 show ipv6 rip output descriptions

Field	Description
IPv6 RIP status/port	The status of RIPng on the device. Possible status is "enabled" or "disabled." The UDP port number over which RIPng is enabled.
Administrative distance	The setting of the administrative distance for RIPng.
Updates/expiration	The settings of the RIPng update and timeout timers.
Holddown/garbage collection	The settings of the RIPng hold-down and garbage-collection timers.
Split horizon/poison reverse	The status of the RIPng split horizon and poison reverse features. Possible status is "on" or "off."
Default routes	The status of RIPng default routes.
Periodic updates/trigger updates	The number of periodic updates and triggered updates sent by the RIPng Brocade device.
Distribution lists	The inbound and outbound distribution lists applied to RIPng.
Redistribution	The types of IPv6 routes redistributed into RIPng. The types can include the following: • STATIC - IPv6 static routes are redistributed into RIPng. • CONNECTED - Directly connected IPv6 networks are redistributed into RIPng. • BGP - BGP4+ routes are redistributed into RIPng. • IS-IS – IPv6IS-IS routes are redistributed into RIPng. • OSPF - OSPFv3 routes are redistributed into RIPng.

Displaying RIPng configuration for a VRF instance

To display the RIPng configuration information for a VRF instance, enter the following command:

```
device# show ipv6 rip vrf red
IPv6 rip enabled, port 521
      Administrative distance is 120
      Updates every 30 seconds, expire after 180
      Holddown lasts 180 seconds, garbage collect after 120
      Split horizon is on; poison reverse is off
      Default originate routes are not generated
      Periodic updates 1137, trigger updates 6
      Distribute List, Inbound : Not set
      Distribute List, Outbound : Not set
      Redistribute: BGP
```

Syntax: ipv6 rip vrf *vrf-name*

Displaying RIPng routing table

To display the RIPng routing table, enter the following command at any CLI level.

```
device# show ipv6 rip route
IPv6 RIP Routing Table - 4 entries:
2001:db8::/64, from ::, null (0)
         CONNECTED, metric 1, tag 0, timers: none
2001:db8:46a::/64, from ::, null (1)
         CONNECTED, metric 1, tag 0, timers: none
2001:db8::1/128, from ::, null (2)
          CONNECTED, metric 1, tag 0, timers: none
2001:db8:2::/64, from ::, null (3)
CONNECTED, metric 1, tag 0, timers: none
```

Syntax: show ipv6 rip route [*ipv6-prefix*/*prefix-length* | *ipv6-address*]

The *ipv6-prefix/prefix-length* parameters restrict the display to the entries for the specified IPv6 prefix. You must specify the *ipv6-prefix* parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373. You must specify the *prefix-length* parameter as a decimal value. A slash mark (/) must follow the *ipv6-prefix* parameter and precede the *prefix-length* parameter.

The *ipv6-address* parameter restricts the display to the entries for the specified IPv6 address. You must specify this parameter in hexadecimal using 16-bit values between colons as documented in RFC 2373.

TABLE 150 show ipv6 rip route output descriptions

Field	Description
IPv6 RIP Routing Table entries	The total number of entries in the RIPng routing table.
ipv6-prefix /*prefix-length*	The IPv6 prefix and prefix length.
ipv6-address	The IPv6 address.
Next-hop router	The next-hop router for this Brocade device. If :: appears, the route is originated locally.
Interface	The interface name. If "null" appears, the interface is originated locally.
Source of route	The source of the route information. The source can be one of the following: • RIP - routes learned by RIPng. • CONNECTED - IPv6 routes redistributed from directly connected networks. • STATIC - IPv6 static routes are redistributed into RIPng. • BGP - BGP4+ routes are redistributed into RIPng. • OSPF - OSPFv3 routes are redistributed into RIPng.
Metric *number*	The cost of the route. The *number* parameter indicates the number of hops to the destination.
Tag *number*	The tag value of the route.
Timers	Indicates if the hold-down timer or the garbage-collection timer is set.

Displaying RIPng routing table for a VRF instance

To display the RIPng route information for a specified VRF, enter the following command at any CLI level.

```
device# show ipv6 rip vrf red route
IPv6 RIP Routing Table - 4 entries:
2001:db8::/64, from ::, null (0)
         CONNECTED, metric 1, tag 0, timers: none
2001:db8:46a::/64, from ::, null (1)
         CONNECTED, metric 1, tag 0, timers: none
2001:db8::1/128, from ::, null (2)
          CONNECTED, metric 1, tag 0, timers: none
2001:db8:2::/64, from ::, null (3)
CONNECTED, metric 1, tag 0, timers: none
```

Syntax: ipv6 rip [**vrf** *vrf-name*] **route** [*ipv6-prefix/prefix-length* | *ipv6-address*]

Policy-Based Routing (IPv6)

- Configuration considerations.....674
- Configuring an IPv6 PBR policy.....675
- Configuration examples.....679
- Displaying IPv6 PBR information.....680
- Policy based routing with the preserve VLAN option.....682
- Configuration examples.....683
- Policy-based routing support for preserve VLAN.....685

Table 151 displays the individual Brocade devices and the IPv6 Policy-Based Routing features they support.

TABLE 151 Supported Brocade IPv6 Policy-Based Routing features

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
IPv6 Policy-Based Routing (PBR)	Yes	Yes	No	Yes	Yes	Yes	Yes
Policy-based routing support for preserve VLAN	Yes	Yes	No	No	No	No	No

Features supported							
Policy-Based Routing (PBR) with Preserve VLAN option	Yes	Yes	No	No	No	No	

IPv6 Policy-Based Routing (IPv6 PBR) allows you to manually configure how IPv6 packets that match certain criteria can be forwarded instead of following the IPv6 Routing Table Manager (RTM) routes. ACLs and route maps are used to selectively modify and route IP packets in hardware. The ACLs classify the traffic. Route maps that match on the ACLs set routing attributes for the traffic.

An IPv6 PBR policy specifies the next hop for traffic that matches the policy. With IPv6 ACLs, you can route IPv6 packets based on all of the match criteria in the IPv6 ACL.

You can configure the Brocade device to perform the following types of PBR:

- Select the next hop gateway.
- Send the packet to the null interface (Null0).

When an IPv6 PBR policy has multiple next hops to a destination, PBR selects the first live next hop specified in the policy that is up. If none of the policy's next hops is available, the packets are forwarded as per the routing table.

Configuration considerations

The configuration considerations are as follows:

- IPv6 PBR cannot be applied globally. IPv6 PBR can only be applied at the interface level.
- IPv6 PBR only supports default VRF. Multi-VRF is not supported.
- IPv6 PBR currently does not support the changing IPv4/IPv6 MTU.
- IPv6 PBR policies are not supported on Layer 3 VPNs.
- IPv6 PBR is applied to routed traffic only by default, except when the flood VLAN option is enabled.
- IPv6 PBR can only be configured on physical ports, Link Aggregation Groups (LAG) ports, and Virtual Ethernet (VEs).
- If IPv6 PBR is applied on a VE, it works only when the VE is enabled. When the VE is disabled, IPv6 PBR will not work and normal routing or switching takes place for the traffic received on the VE.
- The following combinations of IPv6 PBR and IPv4 PBR and IPv6 ACL and IPv4 ACL are allowed:
  - IPv6 PBR and IPv4 PBR can be applied to the same interface at the same time.
  - IPv6 PBR and IPv4 ACL can be applied to the same interface at the same time.
  - IPv4 PBR and IPv6 ACL can be applied to the same interface at the same time.
- IPv6 ACL-based rate limiting is not supported on Brocade NetIron MLX Series and Brocade NetIron XMR Series devices.
- You cannot apply IPv6 PBR on a port if that port already has inbound IPv6 ACLs.
- IPv6 PBR only supports IPv6 as the next hop and IPv6 PBR to VLAN flooding.
- IPv6 PBR does not support IPv6 PBR to GRE and IPv6 PBR to MPLS.
- The number of route maps that can be defined is limited by the system memory. When a route map is used in an IPv6 PBR policy, the IPv6 PBR policy uses up to 64 instances of a route map, up to 5 ACLs in a matching policy of each route map instance.

The following two conditions can cause more than 64 route map instances to be used.

1. If one or more of the first 64 instances has a deny clause.
2. If the access list used in the first 64 instances is not configured.

- ACLs with the **log** option configured should not be used for IPv6 PBR purposes.
- IPv6 PBR ignores implicit **deny ip any any** ACL entries to ensure that traffic is compared to all the ACLs for route maps that use multiple ACLs. However, if an explicit **deny ip any any** entry is configured, traffic matching this clause will be routed normally using Layer 3 paths and will not be compared to any ACL clauses that follow this clause.
- IPv6 PBR always selects the first next hop from the next hop list that is up. If an IPv6 PBR policy's next hop goes down, the policy uses another next hop if available. If no next hops are available, the device routes the traffic in the normal way. No IPv6 ECMP is supported.
- Any changes to route map definitions will be effective immediately for the interfaces where the IPv6 PBR route map is applied, without rebinding. However, rebinding is required for an ACL definition change to take effect.
- PBR policies are not supported on Layer-3 VPNs.
- In a PBR route-map definition, if even one route-map instance contains a "set next-hop-flood-vlan" statement, all instances of that route-map will apply to both routed and switched traffic.

- Flooding traffic to a POS interface is not allowed. It can only be flooded to Ethernet ports on the VLAN, including the default VLAN.
- When an incoming port is POS then the SA of the outgoing flooded packets will be 0.

Considerations specific to Brocade NetIron CES Series and Brocade NetIron CER Series

- IPv6 PBR only supports routed traffic. Switched traffic is not supported.
- IPv6 PBR on transit MPLS uplinks is not supported. IPv6 PBR on an Egress MPLS interface is supported. For example, an IPv6 PBR policy is applied on an MPLS interface.
  - In a transit router, if a 6PE packet is received on an MPLS interface, after the MPLS Label is swapped/pushed, the underlying IPv6 packet will not be subjected to PBR and will be switched based on the MPLS Label.
  - In an egress PE router, if a 6PE packet is received on an MPLS interface, after the MPLS Labels are removed, the underlying IPv6 packet will be subjected to PBR and will be forwarded based on PBR next hop.
- IPv6 PBR to flood VLAN is not supported for switched traffic for the Brocade NetIron CES Series and Brocade NetIron CER Series.

Configuring an IPv6 PBR policy

To configure IPv6 PBR, first define the policies using IP ACLs and route maps, and then enable IPv6 on individual interfaces. The device programs the ACLs into the Layer 4 CAM on the interfaces and forwards traffic that matches the ACLs according to the instructions in the route maps.

To configure an IPv6 PBR policy:

- Configure IPv6 ACLs that specify all the conditions required to match the desired packets.
- Configure a route map that matches on the IPv6 ACLs and sets the route information.
- Apply the route map to a specific IPv6 interface to enable IPv6 PBR.

Configuring the route map

After configuring the ACLs, you can configure an IPv6 PBR route map that matches based on the ACLs and sets routing information in the IP traffic.

NOTE
The **match** and **set** statements described in this section are the only route map statements supported for IPv6 PBR. Other route map statements described in the documentation apply only to the protocols with which they are described.

NOTE
If none of the clauses of an IPv6 PBR route map definition contain both **match** and **set** statements together, PBR will not work and normal routing takes place.

To configure an IPv6 PBR route map, enter commands such as the following.

```
device(config-ipv6-access-list v6acl)#route-map v6pbr permit 10
```

```
device(config-routemap v6pbr)#match ipv6 address v6acl
device(config-routemap v6pbr)#set ipv6 next-hop 2001:db8::1
```

The commands in this example configure an entry in a route map named "ipv6_pbr_map". The **match** statement matches on IP information in ACL v6acl. The **set** statement changes the next hop IPv6 address for packets that match ACL v6acl to 2001:db8::1.

Syntax: [no] route-map map-name permit | **deny num**

The *map-name* variable is a string of characters that names the map. Map names can be up to 80 characters in length. You can define an unlimited number of route maps on the Brocade device, as long as system memory is available.

The **permit | deny** parameter specifies the action the Brocade device will take if a route matches a match statement:

- If you specify a **deny** route map instance, it is ignored and not programmed in Layer 4 CAM.
- If you specify **permit**, the Brocade device applies the match and set statements associated with this route map instance.

The *num* parameter specifies the instance of the route map you are defining. Routes are compared to the instances in ascending numerical order. For example, a route is compared to instance 1, then instance 2, and so on.

IPv6 PBR uses up to 64 route map instances for comparison and ignores the rest.

Syntax: [no] match ipv6 address [**ipv6_access_list_name**]

The *ipv6_access_list_name* parameter specifies a IPv6 ACL name. Up to five ACL names can be configured in one match statement.

To add IPv6 ACLs to the existing **match** command, enter the entire command line, including the old IPv6 ACL names. For example, if you want to add v6_acl2, to the previous example, you need to enter the complete command line, as follows:

```
device(config-routemap v6pbr)# match ipv6 address v6_acl1 v6_acl2
```

Setting the next hop

Traffic that matches a match statement in the route map is forwarded as defined by **set** commands. Multiple **set** commands can be configured and when a match condition is met, the device works sequentially through the list of **set** commands until it finds the first next hop that is operational and uses it. If that next hop goes down, the next hop as defined in a **set** command is chosen and if all next hop interfaces in the list are down, the packet is routed as determined in the IP Route Table. If a next hop interface that was down comes back up, the next hop selection process begins again and restarts its selection process from the top of the list.

Options for setting the next hop are described in the following sections:

- Setting the next hop to an IPv6 address on page 676
- Setting the next hop to a Null0 interface on page 677
- Setting next hop VLAN flooding on page 677

Setting the next hop to an IPv6 address

You can set the next hop to an IPv6 address as shown in the following example.

```
device(config)#route-map v6pbr permit 10
device(config-routemap v6pbr)#match ipv6 address v6acl
device(config-routemap v6pbr)#set ipv6 next-hop 2001:db8::1
```

Syntax: [no] set ipv6 next-hop ipv6-address

The *ipv6-address* variable specifies the IPv6 address to which the packets will be sent.

NOTE
Do not use the IPv6 link-local address, unique local address, or the IPv6 address of the router as the IPv6 next hop address.

Setting the next hop to a Null0 interface

Sending traffic to a NullO interface drops the traffic. You can set the next hop to a Null0 interface as shown in the following example.

```
device(config)#route-map v6pbr permit 10
device(config-routemap v6pbr)#match ipv6 address v6acl
device(config-routemap v6pbr)#set interface null0
```

Syntax: [no] set interface null0

Setting next hop VLAN flooding

Using the **set next-hop-flood-vlan** command, matched traffic can be flooded on all ports of the VLAN except the incoming physical port. Any IPv6 PBR policy that contains the **set next-hop-flood-vlan** command applies to both routed and switched traffic. If any instance in an IPv6 PBR route map contains the **set next-hop-flood-vlan** command, all instances of that route map will be applied to both routed and switched traffic.

The following example floods all traffic matched from ACL 101 on all ports of VLAN 10 except the incoming physical port.

```
device(config)# access-list for_pbr_match_src permit ipv6 any any
device(config)# route-map calea permit 10
device(config-routemap
calea)# match ipv6 address for_pbr_match_src
device(config-routemap
calea)# set next-hop-flood-vlan 10
device(config-routemap
calea)# exit
```

Syntax: [no] set next-hop-flood-vlan vlan-id [**outgoing-da mac-address**]

If the VLAN specified by the *vlan-id* variable is not configured, the IPv6 PBR route map set statement will fall through to the next configured set statement. If no valid next hop is available, the packet is forwarded as per the Layer 2 or Layer 3 forwarding decision. If the VLAN specified by the *vlan-id* variable has no valid outgoing ports (such as when all ports in the VLAN are down or when the VLAN is empty), matching packets will be dropped.

The **outgoing-da** option directs the device to send packets flooded to the ports on the VLAN to carry the destination MAC address specified in the *mac-address* variable.

If the destination MAC address is not set using the **outgoing-da** option, the destination address is set as described in Table 152 .

TABLE 152 Destination address on VLAN flooded packets

Incoming port	Outgoing port	Routed traffic	Switched traffic
Ethernet	Ethernet	Replaced Destination Address	Original Destination Address from Incoming Packet

TABLE 152 Destination address on VLAN flooded packets (Continued)

Incoming port	Outgoing port	Routed traffic	Switched traffic
POS	Ethernet	Replaced Destination Address	N/A

The Brocade NetIron CES Series and Brocade NetIron CER Series exhibit different behavior which is described in Table 153 .

The **no set next-hop-flood-vlan** *vlan-id***outgoing-da***mac-address* command deletes only the **outgoing-da** option from the set statement. It does not delete the set statement itself. To delete the set statement, specify the **no set next-hop-flood-vlan** *vlan-id* command.

In the case of traffic incoming on the MPLS uplink, IPv6 PBR to VLAN flooding is only supported for IPv6 traffic, and not for MPLS traffic.

There is no Layer 3 header processing. For example, in the case of an IPv6 packet header, the hop-limit will not be decremented.

Table 153 describes the difference in behavior between different brocade products.

TABLE 153 VLAN flooding behavior differences

Scenario	Brocade NetIron CER Series and Brocade NetIron CES Series	Brocade NetIron MLX Series and Brocade NetIron XMR Series
Support for Switched Traffic	Not Supported (due to hardware limitation)	Supported
If the **outgoing-da** option is not set.	CES/CER: Packet floods with original SA/DA	XMR/MLX: Switched traffic: Packet floods with original SA/DA Routed traffic (L3 CAM miss): Packet floods with original SA/DA Routed traffic (L3 CAM hit): Packet floods with original SA and DA as next-hop MAC
If the **outgoing-da** option is set.	DA: Configured outgoing-da SA: outgoing port MAC	DA: Configured outgoing-da SA: Original SA
In Set next-hop-flood-vlan, if the VLAN-ID is configured but no ports are added or all ports are down.	Go to next set statement.	DROP matching packets.
Ingress VLAN and PBR Flood VLAN are the same.	Packets are flooded back to source port.	Packets are not flooded back to source port.

Enabling IPv6 PBR

After you configure the ACLs and route map entries, you can enable IPv6 PBR on individual interfaces. To enable IPv6 PBR, you apply a route map you have configured for IPv6 PBR locally.

Enabling IPv6 PBR locally

To enable IPv6 PBR locally, enter the following commands:

```
device(config)# interface ve 1
device(config-vif-1)# ipv6 policy route-map test-route
```

The commands in this example change the CLI to the Interface level for virtual interface 1, and then applies the "test-route" route map to the interface. You can apply an IPv6 PBR route map to Ethernet ports or virtual interfaces.

Syntax: [no] ipv6 policy route-map map-name

Enter the name of the route map you want to use for the route map *map-name* parameter.

LAG formation

When a LAG is formed, all ports must have the same PBR configuration before deployment. During deployment, the configuration on the primary port is replicated to all ports. On undeployment, each port inherits the same PBR configuration.

Configuration examples

This section presents configuration examples for:

- Basic example on page 679
- Combined example on page 679
- Selectively applying normal routing to packets on page 680

Basic example

The following commands configure and apply an IPv6 PBR policy that routes HTTP traffic received on a virtual routing interface.

```
device(config)# route-map v6pbr permit 10
device(config-routemap v6pbr)#match ipv6 address v6acl
device(config-routemap v6pbr)#set ipv6 next-hop 2001:db8::1
```

Combined example

If both IPv4 and IPv6 configurations exist in a route map, the IPv4 portions of the route map will be ignored when the route map is used for IPv6 PBR, and the IPv6 portions of the route map will be ignored when the route map is used for IPv4 PBR.

In the following example the IPv4 traffic that matches the route map will be sent to 10.1.1.1 and IPv6 traffic that matches the route map will be sent to 2001:db8::1.

```
device(config)# interface ethernet 1/2
device(config-route-map)# ip address 10.1.1.1/24
device(config-route-map)# ip policy route-map ipv6_pbr_map
device(config-route-map)# ipv6 address 2001:db8::1/64
device(config-route-map)# ipv6 policy route-map ipv6_pbr_map
```

Selectively applying normal routing to packets

Certain situations demand selected traffic to undergo normal routing based on IPv6 RTM while other traffic is to be forwarded based on PBR. One such situation is provided in the following scenario. The example provides a recommended solution.

To communicate with a direct connected host, the IPv6 address has to be resolved. To resolve the IPv6 address, an ICMPv6 Neighbor Solicitation (NS) will be sent and an ICMPv6 Neighbor Advertisement (NA) is expected as a reply. If the match ACL rules matches the IPv6 address of ICMPv6 NA, as per the PBR route map, it will be redirected to the first reachable next hop and the IPv6 address is not resolved. This results in packets being dropped. To avoid this situation, use the **deny** ACL clause to selectively allow ICMP traffic and define it as a permit route map entry, as shown in the following example.

```
device(config)#
device(config)#ipv6 access-list v6acl
device(config-ipv6-access-list v6acl)#deny icmp 2001:db8::/64 any
device(config-ipv6-access-list v6acl)#permit ipv6 2001:db8::/64 any
device(config-ipv6-access-list v6acl)#
device(config-ipv6-access-list v6acl)#route-map v6pbr permit 10
device(config-routemap v6pbr)#match ipv6 address v6acl
device(config-routemap v6pbr)#set next-hop-flood-vlan 200 outgoing-da
0000.0022.3333
device(config-routemap v6pbr)#
device(config-routemap v6pbr)#interface ethernet 2/2
device(config-if-e10000-2/2)#ipv6 address 2001:db8::1/64
device(config-if-e10000-2/2)#ipv6 enable
device(config-if-e10000-2/2)#ipv6 policy route-map v6pbr
```

To allow normal routing for certain traffic, a corresponding deny ACL filter can be added before any permit ACL filter.

Displaying IPv6 PBR information

The following sections describe the commands used to display the IPv6 ACL accounting information and the route map information used in the IPv6 PBR configuration.

Displaying IPv6 accounting information

IPv6 PBR accounting is supported through the use of IPv6 ACL accounting in Brocade NetIron XMR Series and Brocade NetIron MLX Series devices. ACL counters need to be enabled to view accounting data.

```
device(config)#enable-acl-counter
device# sh ipv6 access-list accounting brief policy-based-routing
IPv6 Policy Based Routing Accounting Summary: (ac = accumulated since accounting
started)
    Int    In ACL              Total In Hit   Out ACL             Total Out Hit
    2/6v6ACL1-udp                     0(1s)
```

```
                                        0(1m)
                                        0(5m)
                                        0(ac)
    2/6v6ACL1-tcp                       0(1s)
                                        0(1m)
                                        0(5m)
                                        0(ac)
    2/6ipv6-src-d                       0(1s)
                                        0(1m)
                                        0(5m)
                                        0(ac)
  VE 40v6ACL1-udp                   11659(1s)
                                   700150(1m)
                                  3500549(5m)
                                160277093(ac)
  VE 40v6ACL1-tcp                     127319(1s)
                                     7645985(1m)
                                    38227679(5m)
                                  1750303054(ac)
```

Syntax: show ipv6 access-list accounting [**brief** | **eth** *s/p* | **ve** *id*] **policy-based-routing**

The **brief** option provides a brief display of the statistics. The **eth***s*/*p* option provides Ethernet statistics for the specified port. The **ve** *id* option provides VE statistics for the specified ID.

Displaying IPv6 PBR route map information

Use the commands listed in the following sections to display the route map information used in the IPv6 PBR configuration.

Displaying IPv6 route map information

To view the route map information, use the **show route-map** command.

```
device# show route-map ipv6_pbr_map
route-map ipv6_pbr_map permit 10
    match ipv6 address for_pbr_match_src
    set ipv6 next-hop 2001:db8::1
```

Syntax: show route-map *map-name*

The *map-name* variable is the name of the PBR route map you want to view.

To view the route map binding information, use the **show route-map binding** command.

```
device# show route-map binding ipv6_pbr_map
    IPv6 Bindings of ipv6_pbr_map :
        1/1
```

Syntax: show route-map binding [*map-name*]

The *map-name* variable is the name of the route map binding you want to view.

Displaying IPv6 ACL and route map informationon the Brocade NetIron CES Series and Brocade NetIron CER Series

Displaying IPv6 ACL output

To view the IPv6 ACL output on the Brocade NetIron CES Series and Brocade NetIron CER Series use the **show ipv6 access-list** command.

```
device# show ipv6 access-list ipv6acl
ipv6 access-list ipv6acl: 3 entries
10: permit enable-accounting ipv6 2001:db8:3::/64 2001:db8:4::/64
20: permit ipv6 2001:db8:3::/64 any
30: permit ipv6 any 2001:db8:4::/64
```

Syntax: show ipv6 access-list *ACL listname*

The *ACL list name* variable is the name of the ACL you want to view.

Displaying IPv6 route map output

To view the route map output on the Brocade NetIron CES Series and Brocade NetIron CER Series use the **show route-map** command.

```
device# show route-map pbr-1
route-map pbr-1 permit 5
match ipv6 address ipv6acl
set ipv6 next-hop 2001:db8:5::2
set ipv6 next-hop 2001:db8:5::3
set ipv6 next-hop 2001:db8:4::2
```

Syntax: show route-map [*map-name*]

The *map-name* variable is the name of the route map you want to view.

Displaying IPv6 PBR selected next hop information

To view the PBR selected next hop information on the Brocade NetIron CES Series and Brocade NetIron CER Series use the **show pbr inte ve** command.

```
device# show pbr int ve 20
Interface VE 20
No PBR routemap
IPv6 PBR Routemap pbr-1
PPCR 1:1: Routemap instance 5
Valid 1 ACL ipv6acl First table index 10
Metro Specific information in PPCR: 1:1
Instance No: 5
SET information
Next-Hop To NH value PumaNextStage SW NextStage id HW NextStage id Resolved
IPv6 address 2001:db8:5::2 IPv6 NEXT HOP 0x02000020 32 Y
IPv6 address 2001:db8:5::3 IPv6 NEXT HOP 0xFFFFFFFF -1 N
IPv6 address 2001:db8:4::2 IPv6 NEXT HOP 0x02000021 33 Y
```

The SET information table provides the hop status list. The ACL uses the first resolved next hop.

Syntax: show pbr int ve *id*

Policy based routing with the preserve VLAN option

When an IP packet matches the PBR policy with the **preserve-vlan** option, the Layer 2 and Layer 3 information is retained (for example, the VLAN information and the MAC address are retained). The hop limit is not decremented. A packet is sent to the configured next hop. IP packets not matching the

PBR policy with **preserve-vlan** will be dropped. If none of the policy's direct routes or next hops are available, the packets are forwarded as per the routing table.

Configuring a physical interface to accept all VLAN packets for PBR

The **allow-all-vlan pbr** command configures a physical interface to accept all VLAN packets for the purpose of PBR. This command reduces configuration complexity since the physical interface does not have to be configured individually in multiple VLAN interfaces.

Syntax: allow-all-vlan pbr

NOTE
The **allow-all-vlan pbr** command cannot be applied to a VE.

Configuration considerations

- The command **allow-all-vlan pbr** cannot be configured when the physical port is configured with an IPv6 address, MPLS, VPLS, VLL, ICL, Layer 3 VPN; or when the port is part of other VLAN.
- The route map with **preserve-vlan** set policies cannot be configured globally.
- A route map used for PBR with a preserve VLAN policy must have the **preserve-vlan** keyword configured for each set policy.

Configuring policy based routing with the preserve VLAN option

The interface, on which PBR with **preserve-vlan** is configured, should be part of the VLANs through which packets are expected. A route map policy with set policies to preserve VLAN can be applied on a physical port or on a VE port.

Preserve VLAN option as part of a set policy

In a route map set policy configuration, the **preserve-vlan** keyword is used to preserve the packet.

Syntax: set ipv6 next-hop IP *v6-address* **preserve-vlan**

Syntax: set next-hop-flood-vlan *vlan-id* **preserve-vlan**

Configuration examples

This section presents the following configuration examples:

- Preserve VLAN IDs and forwarding to specific egress port on page 684
- Preserve VLAN IDs and forwarding to multiple ports within a VLAN on page 684
- Applying IPv6 PBR next hop VLAN flooding on page 684

Preserve VLAN IDs and forwarding to specific egress port

1. Configure the access list for IPv6.

```
device(config)# ipv6 access-list v6-acl
device(config-ipv6-access-list v6-acl)# permit ipv6 2001:db8::/64 any
device(config-ipv6-access-list v6-acl)# exit
```

2. Configure the route map with a set policy to preserve VLAN for IPv6 traffic.

```
device(config)# route-map map6 permit 1
device(config-routemap map4ve)# match ipv6 address v6-acl
device(config-routemap map4ve)# set ipv6 next-hop 2001:db8::1 preserve-vlan
device(config-routemap map4ve)# exit
```

3. Apply route map to physical/VE interface.

```
device(config)# interface e1/2
device(config-if-e1000-1/2)# allow-all-vlan pbr
device(config-if-e1000-1/2)# ipv6 policy route-map map6
device(config-if-e1000-1/2)# exit
device(config)# int ve 40
device(config-vif-40)# ipv6 policy route-map map6
```

Preserve VLAN IDs and forwarding to multiple ports within a VLAN

1. Configure the route map with set policies to preserve VLAN for IPv4/v6 traffic.

```
device(config)# route-map test permit 100
device(config-routemap test)# match ipv6 address v6-acl
device(config-routemap test)# set next-hop-flood-vlan 200 preserve-vlan
device(config-routemap test)# exit
```

2. Apply the route map to physical or VE interface.

```
device(config)# interface ethernet 3/1
device(config-if-e1000-3/1)# allow-all-vlan pbr
device(config-if-e1000-3/1)# ipv6 policy route-map test
device(config-if-e1000-3/1)# exit
device(config-)# int ve 10
device(config-vif-10)# ipv6 policy route-map test
```

Applying IPv6 PBR next hop VLAN flooding

This example demonstrates how to configure matched traffic to be flooded on all ports of the VLAN except the incoming physical port.

ACL configuration

```
device(config)#ipv6 access-list ipv6acl-3
device(config-ipv6-access-list ipv6acl-3)#permit ipv6 2001:db8:3::/64 2001:db8:6::/64
device(config-ipv6-access-list ipv6acl-3)#deny ipv6 any 2001:db8:6::/64
device(config-ipv6-access-list ipv6acl-3)#exit
```

Route-map configuration

```
device(config)#route-map pbr-2 permit 1
device(config-routemap pbr-2)#match ipv6 address ipv6acl-3
device(config-routemap pbr-2)#set next-hop-flood-vlan 30
device(config-routemap pbr-2)#set next-hop-flood-vlan 40 outgoing-da 0000.0034.5678
device(config-routemap pbr-2)#set next-hop-flood-vlan 20
device(config-routemap pbr-2)#set ipv6 next-hop 2001:db8:5::2
```

Apply route-map

```
device(config)#interface ethernet 1/3
device(config-if-e1000-1/3)#ipv6 policy route-map pbr-2
device(config)#interface ve 10
device(config-vif-10)#ipv6 policy route-map pbr-2
```

Policy-based routing support for preserve VLAN

NOTE
Policy-based routing support for preserve VLAN is supported only on Brocade NetIron XMR Series and Brocade NetIron MLX Series routers. This feature is now supported on the BR-MLX-10GX24 module.

Previously, PBR transparent VLAN flooding (TVF) replaced the ingress traffic's VLAN ID with the egress TVF VLAN ID, while flooding the egress TVF VLAN. Policy-based routing support for preserve VLAN allows the ingress packet VLAN header (VLAN ID and priority) to be preserved, while simultaneously flooding the PBR TVF VLAN.

The PBR TVF VLAN egress ports can be in strict tagged VLAN mode or dual VLAN mode. When PBR TVF VLAN egress ports are in strict tagged VLAN mode, the ingress tagged packets flood as "tagged" with the original VLAN ID and priority preserved. The ingress untagged packets flood as "tagged" with the default VLAN ID. When the PBR TVF VLAN egress ports are in dual VLAN mode, the ingress tagged packets flood as "tagged" with the original VLAN ID and priority preserved. The ingress untagged packets flood as "untagged".

Configuration considerations

Consider the following when policy-based routing is supported for preserve VLAN:

- To preserve the ingress VLAN priority value, the ingress VLAN and the port QoS feature should not be configured at the same time.
- IPv4 and IPv6 ACL VLAN ID matches are supported for both ingress and egress ACLs.
- An egress ACL is supported to filter traffic.
- Policy-based routing support for the preserve VLAN option does not affect the feature implementation of policy-based routing support for the preserve VLAN.

VRRP and VRRP-E

- Overview of VRRP 688
- Overview of VRRP-E 694
- Comparison of VRRP and VRRP-E 695
- VRRP and VRRP-E parameters 696
- Configuring parameters specific to VRRP 700
- Configuring parameters specific to VRRP-E 703
- Configuring additional VRRP and VRRP-E parameters 705
- Multiple virtual IP address support for VRRP-E 712
- VRRP-E scaling using logical groups 715
- Alternate VRRPv2 checksum for VRRPv3 IPv4 sessions 718
- Displaying VRRP and VRRP-E information for IPv4 719
- Displaying VRRP and VRRP-E information for IPv6 725
- Displaying configuration information for VRRP and VRRP-E 727
- Clearing VRRP or VRRP-E statistics 728
- Configuration examples 728
- VRRP-E Extension for Server Virtualization 734

Table 154 displays the individual Brocade devices and the VRRP and VRRP-E features they support.

TABLE 154 Supported Brocade VRRP and VRRP-E features

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
Standard VRRP	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VRRP Extended (VRRP-E)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VRRP v2 Authentication	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VRRP v3 for IPv4 and IPv6	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VRRP-E v6	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VRRP-E MD5 Authentication	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VRRP-E and VRRP password display	Yes	Yes	Yes	Yes	Yes	Yes	Yes

TABLE 154 Supported Brocade VRRP and VRRP-E features (Continued)

Features supported	Brocade NetIron XMR Series Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
VRRP alongside RIP	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VRRP alongside OSPF	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VRRP alongside BGP4	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VRRP Track Port	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VRRP Track Priority	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VRRP Backup Preempt	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VRRP Master Router Abdication and Reinstatement	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VRRP-Extended Slow Start	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VRRP-Extended Scale Timer	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VRRP-E Extension for Server Virtualization	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Virtual MAC address per VRID	Yes	Yes	Yes	Yes	Yes	Yes	Yes

This chapter describes how to configure the following router redundancy protocols:

- Virtual Router Redundancy Protocol (VRRP) - The standard router redundancy protocol. There are two versions of the protocol. VRRP v2 supports the IPv4 environment and VRRP v3 supports the IPv4 and IPv6 environments.
- VRRP Extended (VRRP-E) - A proprietary version of VRRP that overcomes limitations in the standard protocol. This protocol works only with Brocade IP devices. The devices support VRRP-E version 2 and VRRP-E version 3. VRRP-E v2 supports the IPv4 environment and VRRP-E v3 supports the IPv6 environment.

NOTE
The maximum number of configured VRRP and VRRP-E instances on a Brocade NetIron XMR Series or Brocade NetIron MLX Series device is 2000. However, only 255 instances of VRRP v3 are supported with a sub-second hello interval, and VRRP-E is supported with the scale timer. The maximum number of configured VRRP and VRRP-E instances on Brocade NetIron CES Series and Brocade NetIron CER Series devices is 255.

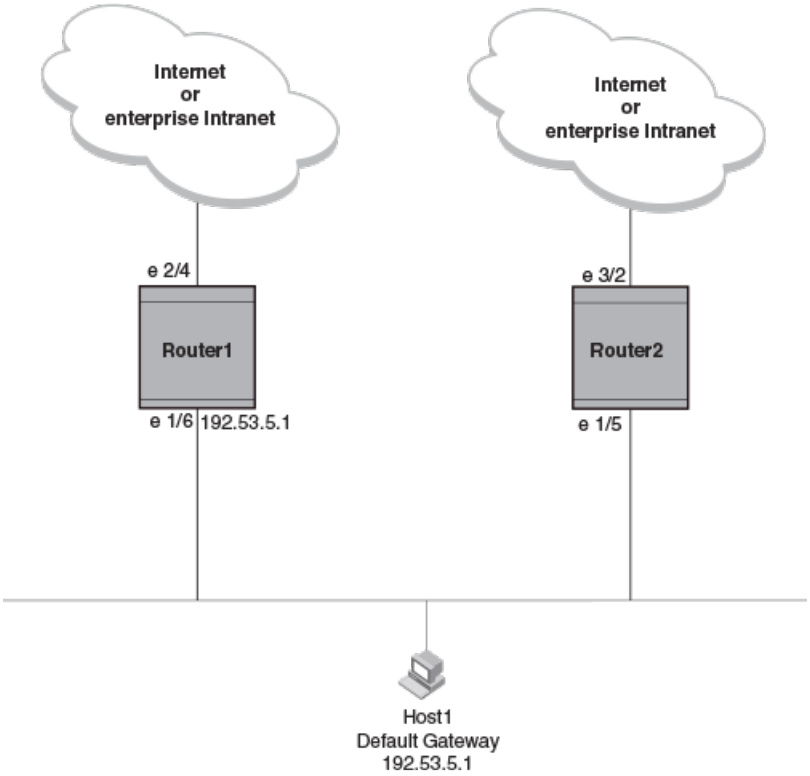
Overview of VRRP

This section presents the standard VRRP options and the options that were added in its implementation of VRRP.

Standard VRRP

VRRP is an election protocol that provides redundancy to routers within a LAN. VRRP allows you to provide alternate router paths for a host without changing the IP address or MAC address by which the host knows its gateway. Consider the situation shown in Figure 38 .

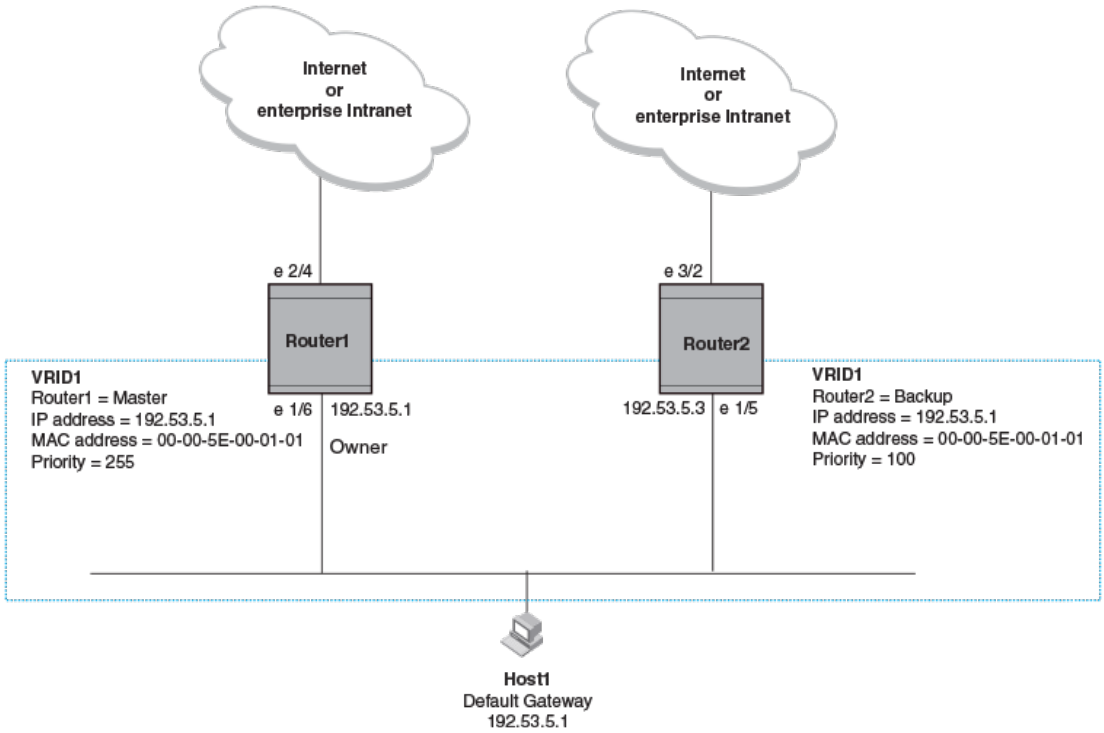
FIGURE 38 Router1 is Host1's default gateway but is a single point of failure



As shown in this example, Host1 uses 192.53.5.1 on Router1 as the host's default gateway out of the subnet. If this interface goes down, Host1 is cut off from the rest of the network. Router1 is thus a single point of failure for Host1's access to other networks.

If Router1 fails, you could configure Host1 to use Router2. Configuring one host with a different default gateway might not require too much extra administration. However, consider a more realistic network with dozens or even hundreds of hosts per subnet; reconfiguring the default gateways for all the hosts is impractical. It is much simpler to configure a VRRP virtual router on Router1 and Router2 to provide a redundant path for the hosts. If VRRP is enabled as in Standard VRRP , Router2 provides the default gateway out of the subnet if Router1 fails.

FIGURE 39 Router1 and Router2 configured as VRRP virtual routers for redundant network access for Host1



With VRRP, you configure virtual routers that span across the physical routers. A virtual router acts as a default router for hosts on a shared LAN. For example, Standard VRRP has one virtual router configured (identified as VRID1).This virtual router ID (VRID) is associated with Router1 and Router2.

Because there is more than one IP address configured on Router1 and Router2, one of the physical addresses is assigned to the virtual router. For example, in Standard VRRP , IP address 192.53.5.1, the IP address assigned to Router1's interface 1/6, is assigned as the IP address of virtual router VRID1. Router1 becomes the "Owner" of the virtual router VRID1 and is the router that responds to packets addressed to any of the IP addresses in virtual router VRID1.

One router in the virtual router is elected as the Master router. Other routers act as backups. The Master router is the one that forwards packets sent to the IP addresses in the virtual router and answers ARP requests for these IP addresses. The Backup router takes over for the Master router if the Master router fails.

NOTE
You can provide more redundancy by also configuring a second VRID with Router2 as the Owner and Router1 as the Backup. This type of configuration is sometimes called Multigroup VRRP.

Master router election

Virtual routers use the VRRP priority values associated with each VRRP router to determine which router becomes the Master. When you configure an Owner router, the VRRP priority is automatically set to 255, the highest VRRP priority. The router in the virtual router with the highest priority becomes the Master. Other routers become the Backup routers and can be assigned priorities from 3 through 254. The default priority value is 100.

Virtual routers use VRID Hello messages to determine if a Master router is available. They send Hello messages to IP Multicast address 224.0.0.18 at a specified frequency. The Backup routers wait for a duration of time for a Hello message from the Master. This duration is called the dead interval. If a Backup router does not receive a Hello message by the time the dead interval expires, the Backup router assumes that the Master router is dead. The Backup router with the highest priority becomes the Master router. Once the Owner router becomes available again, it becomes the Master router and the current Master router returns to being a Backup router.

Pre-emption

If the pre-emption feature is enabled, a Backup router that is acting as the Master can be pre-empted by another Backup router that has a higher priority. This can occur if you add a new Backup while the Owner is still available and the new Backup router has a higher priority than the Backup router that is acting as the Master router.

Virtual router MAC address

When you configure a VRID, the software automatically uses the MAC address as the MAC address of the virtual router. The first five octets of the address are the standard MAC prefix for VRRP packets. The last octet is the VRID. For example, the MAC address for VRID is 000.5e00.0101.

When the virtual router becomes the Master router, it broadcasts a gratuitous ARP request containing the virtual router's MAC address for each IP address associated with the virtual router. In Standard VRRP on page 688, Router1 sends a gratuitous ARP request with MAC address 00-00-00-00-01-01 and IP address 192.53.5.1. Hosts use the virtual router's MAC address in routed traffic they send to their default IP gateway (in this example, 192.53.5.1).

Enhancements to VRRP

Brocade has enhanced VRRP by adding the following options:

- Configuring unique virtual MAC addresses per VRID on page 690
- Track ports and track priority on page 692
- Suppression of RIP advertisements for backed-up interfaces on page 693
- Authentication on page 693
- VRRP alongside RIP, OSPF, and BGP4 on page 694

Configuring unique virtual MAC addresses per VRID

In addition to system-configured standards-based virtual MAC addresses, you can manually configure a unique virtual MAC address for each IPv4 and IPv6 VRRP instance per VRID. For Brocade NetIron XMR Series and Brocade NetIron MLX Series platforms, you can configure a maximum of 2000 virtual MAC addresses.

If there is no manually configured virtual MAC address for a VRRP instance, the system automatically assigns one.

For Brocade NetIron CES Series and Brocade NetIron CER Series platforms, you can configure a maximum of 255 virtual MAC addresses.

This feature is subject to the following limitations:

- This feature does not support configurable VRRP virtual MAC addresses over MCT.
- This feature has no impact on short-path forwarding for VRRP-E.

NOTE
System-assigned virtual MAC addresses and manually configured virtual MAC addresses can exist at the same time on the device under the same VRID, however the configured value takes precedence. When the configured value is deleted, the assigned value again applies.

To configure a unique VRRP or VRRP-E virtual MAC address for a VRID, complete the following steps.

1. To configure an IPv4 virtual MAC address for VRID 1 (for example), enter the following command at the configure VRID level of the CLI:

   ```
   device(config-if-e1000-1/110-vrid-1)virtual-mac aaaa.bbbb.cccc
   ```

   Syntax: [no] virtual - mac *mac-address*

   Use the **no** version of this command to remove the configured address.
2. To configure an IPv6 virtual MAC address for VRID 1 (for example), enter the following command at the configure VRID level of the CLI:

   ```
   device(config-if-e1000-1/110-ipv6-vrid-1)virtual-mac aaaa.bbbb.cccc
   ```

   Syntax: [no] virtual-mac *ipv6mac-address*

   Use the **no** version of this command to remove the configured address.
3. To display IPv4 VRRP virtual MAC address configuration information about VRID 1 (for example), enter the following command:

   ```
   device#show ip vrrp vrid 1
   Interface 1/1
   ----------------
   auth-type no authentication
   VRID 1 (index 1)
    interface 1/1
    state master
    administrative-status enabled
    version v2
    mode owner
    virtual mac aaaa.bbbb.cccc (configured)
    priority 255
    current priority 255
    track-priority 2
    hello-interval 1 sec
    backup hello-interval 60 sec
    ip-address 10.20.1.100
   ```
4. To display IPv4 VRRP-E virtual MAC address configuration information about VRID 1 (for example), enter the following command:

   ```
   device#show ip vrrp-extended vrid 1
   Interface 1/1
   ----------------
   auth-type md5-authentication
   VRID 1 (index 1)
   interface 1/1
   state master
   administrative-status disabled
   mode non-owner(backup)
   virtual mac aaaa.bbbb.cccc (configured)
   priority 100
   current priority 100
   track-priority 5
   hello-interval 1 sec
   backup hello-interval 60 sec
   ```

```
slow-start timer (configured) 30 sec
advertise backup disabled
dead-interval 0 ms
preempt-mode true
virtual ip address 10.20.1.100
short-path-forwarding disabled
```

5. To display IPv6 VRRP virtual MAC address configuration information for VRID 1 (for example), enter the following command:

```
device#show ipv6 vrrp vrid 1
Interface 1/1
---------------
auth-type no authentication
VRID 1 (index 1)
 interface 1/1
 state master
 administrative-status enabled
 version v3
 mode non-owner(backup)
 virtual mac dddd.eeee.ffff (configured)
 priority 100
 current priority 100
 track-priority 1
 hello-interval 1000 ms
 backup hello-interval 60000 ms
 advertise backup disabled
 dead-interval 3600 ms
 preempt-mode true
 ipv6 address 10:20:1::100
 next hello sent in 400 ms
```

6. To display IPv6 VRRP-E virtual MAC address configuration information for VRID 1 (for example), enter the following command:

```
device#show ipv6 vrrp-extended vrid 1
Interface 1/1
---------------
auth-type md5-authentication
VRID 1 (index 1)
 interface 1/1
 state master
 administrative-status enabled
 mode non-owner(backup)
 virtual mac dddd.eeee.ffff (configured)
 priority 100
 current priority 100
 track-priority 5
 hello-interval 1 sec
 backup hello-interval 60 sec
 advertise backup disabled
 dead-interval 0 ms
 preempt-mode true
 virtual ipv6 address 10:20:1::100
```

You can also identify configured virtual MAC addresses by entering the **show running-config** command, as shown in the following example.

```
device# show running-config interface ethernet 1/11
interface ethernet 1/11
enable
ip ospf area 0
ip address 1.1.1.15/24
```

Syntax: show running-config interface *slot/port*

Track ports and track priority

Brocade enhanced VRRP by giving a VRRP router the capability to monitor the state of the interfaces on the other end of the route path through the router. For example, in Standard VRRP on page 688, interface e1/6 on Router1 owns the IP address to which Host1 directs route traffic on its default gateway. The exit path for this traffic is through Router1's e2/4 interface.

Suppose interface e2/4 goes down. Even if interface e1/6 is still up, Host1 is cut off from other networks. In conventional VRRP, Router1 would continue to be the Master router despite the unavailability of the exit interface for the path the router is supporting. However, if you configure interface e1/6 to track the state of interface e2/4, if e2/4 goes down, interface e1/6 responds by changing Router1's VRRP priority to the value of the track priority. In the configuration shown in Standard VRRP on page 688, Router1's priority changes from 255 to 20. One of the parameters contained in the Hello messages the Master router sends to its Backup routers is the Master router's priority. If the track port feature results in a change in the Master router's priority, the Backup routers quickly become aware of the change and initiate a negotiation for Master router.

In Standard VRRP on page 688, the track priority results in Router1's VRRP priority becoming lower than Router2's VRRP priority. As a result, when Router2 learns that it now has a higher priority than Router1, Router2 initiates negotiation for Master router and becomes the new Master router, thus providing an open path for Host1's traffic. To take advantage of the track port feature, make sure the track priorities are always lower than the VRRP priorities. The default track priority for the router that owns the VRID IP addresses is 2. The default track priority for Backup routers is 1. If you change the track port priorities, make sure you assign a higher track priority to the Owner of the IP addresses than the track priority you assign on the Backup routers.

Suppression of RIP advertisements for backed-up interfaces

The Brocade implementation also enhances VRRP by allowing you to configure the protocol to suppress RIP advertisements for the backed-up paths from Backup routers. Normally, a VRRP Backup router includes route information for the interface it is backing up in RIP advertisements. As a result, other routers receive multiple paths for the interface and might sometimes unsuccessfully use the path to the Backup router rather than the path to the Master router. If you enable the Brocade implementation of VRRP to suppress the VRRP Backup routers from advertising the backed-up interface in RIP, other routers learn only the path to the Master router for the backed-up interface.

Authentication

For backward compatibility, VRRP can use simple passwords to authenticate VRRP packets. The VRRP authentication type is not a parameter specific to the VRID. Instead, VRRP uses the authentication type associated with the interfaces on which you define the VRID. For example, if you configure your router interfaces to use a simple password to authenticate traffic, VRRP uses the same simple password, and VRRP packets that do not contain the password are dropped. If your interfaces do not use authentication, neither does VRRP.

NOTE
The MD5 authentication type is not supported by VRRP.

NOTE
Authentication is not supported by VRRP v3.

Forcing a Master router to abdicate to a standby router

You can force a VRRP Master router to abdicate (give away control) of a virtual router to a Backup router by temporarily changing the Master router's priority to a value of the Backup router. When you change the priority of a VRRP Owner, the change takes effect only for the current power cycle. The change is not saved to the startup configuration file when you save the configuration and is not retained across a reload or reboot. Following a reload or reboot, the VRRP Owner again has priority 255.

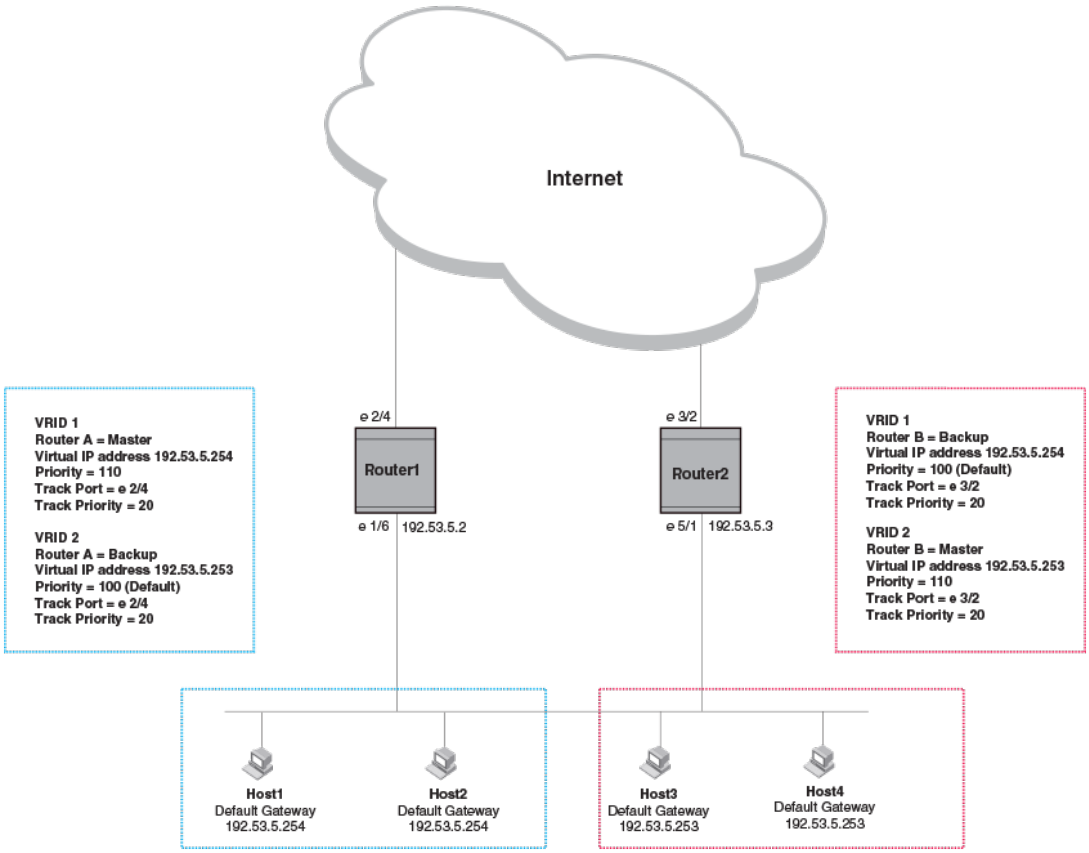
VRRP alongside RIP, OSPF, and BGP4

VRRP operation is independent of the RIP, OSPF, and BGP4 protocols. Their operation is unaffected when VRRP is enabled on a RIP, OSPF, or BGP4 interface.

Overview of VRRP-E

VRRP-E is proprietary version of VRRP that overcomes limitations in the standard protocol. Overview of VRRP-E shows an example of a VRRP-E configuration.

FIGURE 40 Router1 and Router2 are configured to provide dual redundant network access for the host



In this example, Router1 and Router2 use VRRP-E to load share as well as provide redundancy to the hosts. The load sharing is accomplished by creating two VRRP-E groups. Each group has its own virtual IP addresses. Half of the clients point to VRID 1's virtual IP address as their default gateway and the other half point to VRID 2's virtual IP address as their default gateway. This will enable some of the outbound Internet traffic to go through Router1 and the rest to go through Router2.

Router1 is the master for VRID 1 (backup priority = 110) and Router2 is the backup for VRID 1 (backup priority = 100). Router1 and Router2 both track the uplinks to the Internet. If an uplink failure occurs on Router1, its backup priority is decremented by 20 (track priority = 20), so that all traffic destined to the Internet is sent through Router2 instead.

Similarly, Router2 is the master for VRID 2 (backup priority = 110) and Router1 is the backup for VRID 2 (backup priority = 100). Router1 and Router2 are both tracking the uplinks to the Internet. If an uplink failure occurs on Router2, its backup priority is decremented by 20 (track priority = 20), so that all traffic destined to the internet is sent through Router1 instead.

The Brocade device configured for VRRP-E can interoperate only with other Brocade devices.

ARP behavior with VRRP-E

In the VRRP-E implementation, the source MAC address of the gratuitous ARP sent by the VRRP-E master router will be the VRRP-E virtual MAC address. When the router (either master or backup router) sends an ARP request or reply packet, the sender's MAC address will be the MAC address of the interface on the router. When an ARP request packet for the virtual router IP address is received by the backup router, it will be forwarded to the master router to resolve the ARP. Only master router will answer the ARP request for the virtual router IP address.

Comparison of VRRP and VRRP-E

VRRP-E is similar to VRRP, but differs in the following respects:

- Owners and Backups:
  - VRRP has an Owner and one or more Backups for each virtual router. The Owner is the router that has the IP address used for the virtual router. All the other routers supporting the virtual router are Backups.
  - VRRP-E does not use Owners. All routers are Backups for a given virtual router. The router with the highest priority becomes the Master. If there is a tie for highest priority, the router with the highest IP address becomes the Master. The elected Master owns the virtual IP address and answers ping and ARP requests and so on.
- Master and Backups:
  - VRRP - The "Owner" of the IP address of the VRID is the default Master and has the highest priority (255). The precedence of the Backups is determined by their priorities. The default Master is always the Owner of the IP address of the VRID.
  - VRRP-E - The Master and Backups are selected based on their priority. You can configure any of the Brocade devices to be the Master by giving it the highest priority. There is no Owner.
- Virtual Router’s IP address:
  - VRRP requires that the virtual router has an IP address that is configured on the Owner router.
  - VRRP-E requires only that the virtual router’s IP address be in the same subnet as an interface configured on the VRID's interface. In fact, VRRP-E does not allow you to specify an IP address configured on the interface as the VRID IP address.
- VRID's MAC Address:
  - VRRP uses the interfaces’s actual MAC address as the source MAC address. The virtual MAC address for IPv4 VRRP is 00-00-00-00-01-<*vrid*> and for IPv6 VRRP is 00-00-00-00-02-<*vrid*>.The <*vrid*> is the ID of the virtual router. The Master owns the Virtual MAC address.
  - VRRP-E uses the interface’s actual MAC address as the source MAC address. The virtual MAC address for IPv4 VRRP-E and IPv6 VRRP-E is 02-E0-52-<*hash-value*>-<*vrid*>, where <*hash-value*> is a two-octet hashed value for the IP address and <*vrid*> is the virtual router ID.

NOTE
You cannot reuse the same VRID across IPv4 VRRP-E and IPv6 VRRP-E, if they are in the same broadcast domain.

- Hello packets:
  - VRRP sends Hello messages to IP Multicast address 224.0.0.18.
  - VRRP-E uses UDP to send Hello messages in IP multicast messages. The Hello packets use the interface's actual MAC address and IP address as the source addresses. The destination MAC address is 00-00-00-00-00-02, and the destination IP address is 224.0.0.2 (the well-known IP multicast address for "all routers"). Both the source and destination UDP port number is 8888. VRRP messages are encapsulated in the data portion of the packet.
- Track ports and track priority:
  - VRRP changes the priority of the VRID to the track priority, which typically is lower than the VRID priority and lower than the VRID's priorities configured on the Backups. For example, if the VRRP interface's priority is 100 and a tracked interface with track priority 20 goes down, the software changes the VRRP interface's priority to 20.
  - VRRP-E reduces the priority of a VRRP-E interface by the amount of a tracked interface's priority if the tracked interface's link goes down. For example, if the VRRP-E interface's priority is 200 and a tracked interface with track priority 20 goes down, the software changes the VRRP-E interface's priority to 180. If another tracked interface goes down, the software reduces the VRID's priority again, by the amount of the tracked interface's track priority.

The most important difference is that all VRRP-E routers are Backups. There is no Owner router. VRRP-E overcomes the limitations in standard VRRP by removing the Owner.

VRRP and VRRP-E parameters

The table below lists the VRRP and VRRP-E parameters. Most of the parameters and default values are the same for both protocols. Any exceptions are noted in the table.

TABLE 155 VRRP and VRRP-E parameters

Parameter	Description	Default	Refer page...
Protocol	Virtual Router Redundancy Protocol (VRRP) or VRRP-Extended, Brocade's enhanced implementation of VRRP	Disabled	Configuring parameters specific to VRRP on page 700
		NOTE Only one of the protocols can be enabled at a time.	Configuring parameters specific to VRRP-E on page 703

TABLE 155 VRRP and VRRP-E parameters (Continued)

Parameter	Description	Default	Refer page...
VRRP or VRRP-E router	The Brocade device's active participation as a VRRP or VRRP-E router. Enabling the protocol does not activate the Brocade device for VRRP or VRRP-E. You must activate the Brocade device as a VRRP or VRRP-E router after you configure the VRRP or VRRP-E parameters.	Inactive	Configuring parameters specific to VRRP on page 700 Configuring parameters specific to VRRP-E on page 703
Virtual Router ID (VRID)	The ID of the virtual router you are creating by configuring multiple routers to back up an IP interface. You must configure the same VRID on each router that you want to use to back up the address. No default.	None	Configuring parameters specific to VRRP on page 700 Configuring parameters specific to VRRP-E on page 703
Virtual Router IP address	This is the address you are backing up. No default. • VRRP - The virtual router IP address must be a real IP address configured on the VRID interface on one of the VRRP routers. This router is the IP address Owner and is the default Master. • VRRP-E - The virtual router IP address must be in the same subnet as a real IP address configured on the VRRP-E interface, but cannot be the same as a real IP address configured on the interface.	None	Configuring parameters specific to VRRP on page 700 Configuring parameters specific to VRRP-E on page 703
VRID MAC address	The source MAC address in VRRP or VRRP-E packets sent from the VRID interface, and the destination for packets sent to the VRID. • VRRP - A virtual MAC address defined as 00-00-00-00-01-<*vrid*> for IPv4 VRRP and 00-00-00-00-02-<*vrid*> for IPv6 VRRP. The Master owns the Virtual MAC address. • VRRP-E - A virtual MAC address defined as 02-E0-52-<*hash-value*>-<*vrid*> for IPv4 VRRP-E and IPv6 VRRP-E, where <*hash-value*> is a two-octet hashed value for the IP address and <*vrid*> is the ID of the virtual router.	Not configurable	Virtual router MAC address on page 690

TABLE 155 VRRP and VRRP-E parameters (Continued)

Parameter	Description	Default	Refer page...
Authentication type	The type of authentication the VRRP or VRRP-E routers use to validate VRRP or VRRP-E packets. The authentication type must match the authentication type the VRID's port uses with other routing protocols such as OSPF: • No authentication - The interfaces do not use authentication. This is the VRRP default. • Simple - The interface uses a simple text-string as a password in packets sent on the interface. If the interface uses simple password authentication, the VRID configured on the interface must use the same authentication type and the same password. • MD5 - This method of authentication ensures the packet is authentic and cannot be modified in transit. **NOTE** MD5 is not supported by VRRP.Authentication is not supported by VRRP v3.	No authentication	Authentication on page 693 Authentication type on page 705
Router type	Whether the router is an Owner or a Backup: • Owner (VRRP only) - The router on which the real IP address used by the VRID is configured. • Backup - Routers that can provide routing services for the VRID but do not have a real IP address matching the VRID.	VRRP - The Owner is always the router that has the real IP address used by the VRID. All other routers for the VRID are Backups. VRRP-E - All routers for the VRID are Backups.	Configuring parameters specific to VRRP on page 700 Configuring parameters specific to VRRP-E on page 703
Backup priority	A numeric value that determines a Backup's preferability for becoming the Master for the VRID. During negotiation, the router with the highest priority becomes the Master: • VRRP - The Owner has the highest priority (255); other routers can have a priority from 8 through 255. • VRRP-E - All routers are Backups and have the same priority by default. If two or more Backups are tied with the highest priority, the Backup interface with the highest IP address becomes the Master for the VRID.	VRRP - 255 for the Owner; 100 for each Backup VRRP-E - 100 for all Backups	Configuring parameters specific to VRRP on page 700 Configuring parameters specific to VRRP-E on page 703
Suppression of RIP advertisements	A router that is running RIP normally advertises routes to a backed up VRID even when the router is not currently the active router for the VRID. Suppression of these advertisements helps ensure that other routers do not receive invalid route paths for the VRID.	Disabled	Suppressing RIP advertisements on backup routers for the backup up interface on page 707

TABLE 155 VRRP and VRRP-E parameters (Continued)

Parameter	Description	Default	Refer page...
Hello interval	The number of seconds or milliseconds between Hello messages from the Master to the Backups for a given VRID. The interval can be from 1 through 84 seconds for VRRP v2 and VRRP-E v2. The interval can be from 100 milliseconds through 84000 milliseconds for VRRP v3 and VRRP-E v3.	One second (VRRP v2 and VRRP-E v2) 1000 milliseconds (VRRP v3 and VRRP-E v3)	Hello interval on page 707
Dead interval	The number of seconds or milliseconds a Backup waits for a Hello message from the Master for the VRID before determining that the Master is no longer active. If the Master does not send a Hello message before the dead interval expires, the Backups negotiate (compare priorities) to select a new Master for the VRID.	Internally derived from Hello Interval. It is equal to 3 times the Hello interval + Skew time, where Skew time is equal to (256 - priority) divided by 256.	Dead interval on page 708
Backup Hello interval	The number of seconds between Hello messages from a Backup to the Master. The message interval can be from 60 through 3600 seconds. You must enable the Backup to send the messages. The messages are disabled by default on Backups. The current Master (whether the VRRP Owner or a Backup) sends Hello messages by default.	Disabled 60 seconds when enabled	Backup hello message state and interval on page 708
Track port	Another Brocade port or virtual interface whose link status is tracked by the VRID's interface. If the link for a tracked interface goes down, the VRRP or VRRP-E priority of the VRID interface is changed, causing the devices to renegotiate for Master.	None	Track ports and track priority on page 692 Track port on page 709
Track priority	A VRRP or VRRP-E priority value assigned to the tracked ports. If a tracked port's link goes down, the VRID port's VRRP or VRRP-E priority changes: • VRRP - The priority changes to the value of the tracked port's priority. • VRRP-E - The VRID port's priority is reduced by the amount of the tracked port's priority.	VRRP - 2 VRRP-E - 5	Track ports and track priority on page 692 Track priority on page 709
Backup preempt mode	Prevents a Backup with a higher VRRP priority from taking control of the VRID from another Backup that has a lower priority but has already assumed control of the VRID.	Enabled	Backup preempt on page 709
Slow Start	Causes a specified amount of time to elapse between the time the original Master router is restored and when it takes over from the Backup router. For VRRP-E only.	Disabled	VRRP-extended slow start on page 711

TABLE 155 VRRP and VRRP-E parameters (Continued)

Parameter	Description	Default	Refer page...
Scale Timer	Allows you to increase timing sensitivity across all configured or default VRRP-Extended timers. For VRRP-E only.	Disabled	VRRP-extended scale timer on page 711
Short-Path-Forwarding	Enables VRRP-E Extension for Server Virtualization. If enabled, the traffic that is destined to the clients will travel through the short-path forwarding path (dashed line) to reach the client (as shown in Figure 24.4 on page 24-30). Any packets coming from the local subnet of the virtual IP address will be routed to the VRRP-E master router. For example, with VRRP-E Extension for Server Virtualization enabled, the traceroute output will show one extra hop that display the Backup router's interface IP address	Disabled	VRRP-E Extension for server virtualization configuration example on page 735

Configuring parameters specific to VRRP

VRRP is configured at the interface level. This section describes how to implement a simple VRRP configuration using all the default values.

NOTE
When you use the command **router vrrp** to enter the VRRP configuration mode, the command prompt does not change and results in the general configuration command prompt as shown in the following: device `(config)#` This differs from entering the VRRP extended mode where entering the **router vrrp-extended** command results in a command prompt such as the following: device `(config-vrrpe-router)#`

Configuring the VRRP version

You can specify the version for the VRRP instance. For example, use the following command to configure the instance of VRRP to use VRRP v3.

```
device(config-if-e100-1/3-vrid-13)# version v3
```

Syntax: [no] version *v2 |v3*

- VRRP v2 supports IPv4 environment
- VRRP v3 supports IPv4 and IPv6 environment

The default configuration is VRRP v2.

NOTE
Mixed mode VRRP v2 and VRRP v3 is not supported in the same VRRP group.

Configuring the Owner for IPv4

To configure the VRRP Owner router for IPv4, enter the following commands on the router.

```
device(config)# router vrrp
device(config)# interface ethernet 1/6
device(config-if-e10000-1/6)# ip address 10.53.5.1/24
device(config-if-e10000-1/6)# ip vrrp vrid 1
device(config-if-e10000-1/6-vrid-1)# owner
device(config-if-e10000-1/6-vrid-1)# ip-address 10.53.5.1
device(config-if-e10000-1/6-vrid-1)# activate
```

Syntax: [no] router vrrp

Syntax: [no] ip vrrp vrid num

Syntax: [no] owner [**track-priority value**]

Syntax: [no] activate

The *num* parameter specifies the virtual router ID.

The **track-priority***value* parameter changes the track-port priority for this interface and VRID from the default (2) to a value from 1 through maximum VRID supported by the device.

Syntax: [no] ip-address ip-addr

The *ip-addr* parameter specifies the IPv4 address of the Owner router.

The IP address you assign to the Owner must be an IP address configured on an interface that belongs to the virtual router.

Refer to Configuration rules and feature limitations for VRRP on page 703 for additional requirements.

Configuring the Owner for IPv6

To configure the VRRP Owner router for IPv6, enter the following commands on the router.

```
device(config)# ipv6 router vrrp
device(config)# interface ethernet 1/6
device(config-if-e10000-1/6)# ipv6 address 3013::1/64
device(config-if-e10000-1/6)# ipv6 vrrp vrid 1
device(config-if-e10000-1/6-ipv6-vrid-1)# owner
device(config-if-e10000-1/6-ipv6-vrid-1)# ipv6-address 3013::1
device(config-if-e10000-1/6-ipv6-vrid-1)# activate
```

Syntax: [no] ipv6 router vrrp

Syntax: [no] ipv6 vrrp vrid *num*

Syntax: [no] ipv6-address *ipv6-addr*

The *num* parameter specifies the virtual router ID.

The *ipv6-addr* parameter specifies the IPv6 address of the Owner router.

The IP address you assign to the Owner must be an IP address configured on an interface that belongs to the virtual router.

Refer to Configuration rules and feature limitations for VRRP on page 703 for additional requirements.

Configuring a Backup for IPv4

To configure the VRRP Backup router for IPv4, enter the following commands.

```
device(config)# router vrrp
device(config)# interface ethernet 1/5
device(config-if-e10000-1/5)# ip address 10.53.5.3/24
device(config-if-e10000-1/5)# ip vrrp vrid 1
device(config-if-e10000-1/5-vrid-1)# backup
device(config-if-1/5-vrid-1)# advertise backup
device(config-if-e10000-1/5-vrid-1)# ip-address 10.53.5.1
device(config-if-e10000-1/5-vrid-1)# activate
```

When you configure a Backup router, the router interface on which you are configuring the VRID must have a real IP address that is in the same subnet as the address associated with the VRID by the Owner. However, the address cannot be the same.

Syntax: [no] router vrrp

Syntax: [no] ip vrrp vrid num

Syntax: [no] backup [**priority value**] [**track-priority value**]

The *num* parameter specifies the virtual router ID.

The **priority***value* parameter specifies the VRRP priority for this virtual router. You can specify a value from 3 through 254. The default is 100.

Enter a value from 3 through 254 for the **track-priority***value* parameter if you want VRRP to monitor the state of the interface. The default is 100.

Syntax: [no] ip-address ip-addr

Refer to Configuration rules and feature limitations for VRRP on page 703 for additional requirements.

Configuring a Backup for IPv6

To configure the VRRP Backup router for IPv6, enter the following commands.

```
device(config)# ipv6 router vrrp
device(config)# interface ethernet 1/5
device(config-if-e10000-1/5)# ipv6 address 3013::3/64
device(config-if-e10000-1/5)# ipv6 vrrp vrid 1
device(config-if-e10000-1/5-vrid-1)# backup
device(config-if-1/5-vrid-1)# advertise backup
device(config-if-e10000-1/5-vrid-1)# ipv6-address 3013::1
device(config-if-e10000-1/5-vrid-1)# activate
```

When you configure a Backup router, the router interface on which you are configuring the VRID must have a real IP address that is in the same subnet as the address associated with the VRID by the Owner. However, the address cannot be the same.

Syntax: [no] ipv6 router vrrp

Syntax: [no] ipv6-address *ipv6-addr*

The *num* parameter specifies the virtual router ID.

The *ipv6-addr* **parameter specifies the IPv6 address of the Backup router.**

Refer to Configuration rules and feature limitations for VRRP on page 703 for additional requirements.

Configuration rules and feature limitations for VRRP

Consider the following rules when configuring VRRP:

- The interfaces of all routers in a virtual router must be in the same IP subnet:
- The IP addresses associated with the virtual router must already be configured on the router that will be the Owner router.
- The IP address for the virtual router must be on only one router.
- The Hello interval must be set to the same value on both the Owner and Backups for the virtual router.
- The Dead interval must be set to the same value on both the Owner and Backups for the virtual router.
- The track priority on a router must be lower than the router's VRRP priority. Also, the track priority on the Owner must be higher than the track priority on the Backups.
- The tracking-port configuration for IPv6 VRRP v3 is not allowed if the router is configured as the VRRP Owner.
- The priority configuration for IPv6 VRRP v3 is not allowed for Owner router. The Owner router's priority is always 255.
- Hitless switchover is not supported.
- The ping command is not supported for VRRP virtual IPv4 and IPv6 addresses.
- Mixed mode VRRP v2 and VRRP v3 is not supported in the same VRRP group.
- With VRRP or VRRP-E the maximum numbers of VRIDs per logical interface is 12.

Configuring parameters specific to VRRP-E

The following sections describe the configuration of the parameters specific to IPv4 and IPv6 VRRP-E.

Configuring IPv4 VRRP-E

VRRP-E is configured at the interface level. To implement a simple IPv4 VRRP-E configuration using all the default values, enter the following commands.

```
device(config)# router vrrp-extended
device(config)# interface ethernet 1/5
device(config-if-e10000-1/5)# ip address 10.53.5.3/24
device(config-if-e10000-1/5)# ip vrrp-extended vrid 1
device(config-if-e10000-1/5-vrid-1)# backup priority 50 track-priority 10
device(config-if-e10000-1/5-vrid-1)# ip-address 10.53.5.254
device(config-if-e10000-1/5-vrid-1)# activate
```

Syntax: [no] ip vrrp-extended vrid vrid

NOTE
If VRRP is not configured globally, then you will see the response "Invalid input..." when you try to create a VRRP instance.

Syntax: [no] backup [**priority value**] [**track-priority value**]

Syntax: [no] ip-address ip-addr

The *vrid* parameter specifies the virtual router ID.

The *ip-addr* parameter specifies the IPv4 address of the router.

Refer to the section Authentication type on page 705 for information on the **auth-type no-auth** | **simple-text-auth***auth-data* parameters.

Also, refer to Configuration rules and feature limitations for VRRP-E on page 704 additional information on how to configure VRRP-E.

You must identify a VRRP-E router as a Backup before you can activate the virtual router on a Brocade device. However, after you configure the virtual router, you can use the **backup** command to change its priority or track priority.

You also can use the **enable** command to activate the configuration.The **enable** command does the same thing as the **activate** command.

Configuring IPv6 VRRP-E

To implement a IPv6 VRRP-E configuration using all the default values, enter the following commands.

```
device(config)# ipv6 router vrrp-extended
device(config-ipv6-VRRP-E-router)# interface ethernet 1/5
device(config-if-e10000-1/5)# ipv6 address 3013::2/64
device(config-if-e10000-1/5)# ipv6 vrrp-extended vrid 1
device(config-if-e10000-1/5-vrid-1)# backup priority 50 track-priority 10
device(config-if-e10000-1/5-vrid-1)# ipv6-address fe80::768e:f8ff:fe2a:0099
device(config-if-e10000-1/5-vrid-1)# ipv6-address 3013::99
device(config-if-e10000-1/5-vrid-1)# activate
```

Syntax: ipv6 router vrrp-extended

Syntax: ipv6 vrrp-extended vrid vrid

Syntax: [no] ipv6-address ipv6-addr

The *vrid* parameter specifies the virtual router ID.

NOTE
If VRRP is not configured globally, then you will see the response "Invalid input..." when you try to create a VRRP instance.

The *ipv6-addr* parameter specifies the IPv6 address of the router.

Configuration rules and feature limitations for VRRP-E

Consider the following rules when configuring VRRP-E:

- The interfaces of all routers in a virtual router must be in the same IP subnet.
- The IP address assigned to the virtual router cannot be configured on any of the Brocade devices.
- The Hello interval must be set to the same value on all the Brocade devices.
- The Dead interval must be set to the same value on all the Brocade devices.
- The track priority for a virtual router must be lower than the VRRP-E priority.
- The same VRID must not be used across IPv6 VRRP-E and IPv4 VRRP-E if they are in the same broadcast domain.
- Hitless switchover is not supported.

NOTE
If you disable VRRP-E, the Brocade device removes all the configuration information for the disabled protocol from the running configuration. Moreover, when you save the configuration to the startup

configuration after disabling the protocol, all configuration information for the disabled protocol is removed from the startup configuration.

Configuring additional VRRP and VRRP-E parameters

You can modify the following VRRP and VRRP-E parameters on each individual virtual router. These parameters apply to both protocols:

- Authentication type (if the interfaces on which you configure the virtual router use authentication)
- Backup priority
- Suppression of RIP advertisements on Backup routes for the backed up interface
- Hello interval
- Dead interval
- Backup Hello messages and message timer (Backup advertisement)
- Track port
- Track priority
- Backup preempt mode
- Master Router Abdication and Reinstatement
- VRRP-Extended Slow Start
- VRRP-Extended Scale Timer
- Enable password display

Refer to VRRP and VRRP-E parameters on page 696 for a summary of the parameters and their defaults.

Authentication type

If the interfaces on which you configure the virtual router use authentication, the VRRP or VRRP-E packets on those interfaces also must use the same authentication. Brocade's implementation of VRRP and VRRP-E supports the following authentication types:

- **No authentication** - The interfaces do not use authentication. This is the default for VRRP and VRRP-E.
- **Simple** - The interface use a simple text-string as a password in packets sent on the interface. If the interfaces use simple password authentication, the virtual router configured on the interfaces must use the same authentication type and the same password.
- **MD5** - This method of authentication ensures the packet is authentic and cannot be modified in transit. MD5 authentication configuration for a VRRP-E router is unique on a per-interface basis. The MD5 authentication configuration on an interface takes effect for all the VRRP-E routers configured on a particular interface.

Simple Authentication

To configure the interface on Router1 for simple-password authentication using the password "ourpword", enter the following commands:

Configuring router 1

```
device(config)# interface ethernet 1/6
device(config-if-e10000-1/6)# ip vrrp auth-type simple-text-auth ourpword
```

Configuring router 2

```
device(config)# interface ethernet 1/5
device(config-if-e10000-1/5)# ip vrrp auth-type simple-text-auth ourpword
```

Syntax: [no] ip vrrp auth-type no-auth | **simple-text-auth auth-data** | **md5-auth** *auth-data*

The **auth-type no-auth** parameter indicates that the virtual router and the interface it is configured on do not use authentication.

The **auth-type simple-text-auth***auth-data* parameter indicates that the virtual router and the interface it is configured on use a simple text password for authentication. The *auth-data* parameter is the password. If you use this parameter, make sure all interfaces on all the routers supporting this virtual router are configured for simple password authentication and use the same password.

NOTE
Authentication is not supported by VRRP v3.

MD5 Authentication

To configure MD5 authentication on an interface, the CLI commands should be entered at the interface level.

To configure MD5 Authentication on VRRP-E IPv4, enter the following commands at the interface level:

```
device(config)#ip vrrp-extended auth-type md5-auth ourpword
```

To configure MD5 Authentication on VRRP-E IPv6, enter the following commands at the interface level:

```
device(config)#ipv6 vrrp-extended auth-type md5-auth ourpword
```

Syntax: ip | ipv6 vrrp-extended auth-type [md5-auth *string* **]**

The *string* variable specifies a text string that is used as an authentication password key. The maximum length of the key string is limited to 64 characters.

SYSLOG and SNMP traps are generated in the event of a packet being dropped due to MD5 authentication failure.

When MD5 authentication is configured on an interface, the following syslog message is displayed:

```
Aug 10 18:17:39 VRRP6: Configuration VRRP_CONFIG_MD5_AUTHENTICATION request received
Aug 10 18:17:39 VRRP6: Port 2/6, VRID 2 - send advertisement
 Ver:3 Type:1 Vrid:2 Pri:240 #IP:1 AuthType:2 Adv:1 Chksum:0x0000
         HMAC-MD5 CODE:[00000000000000000000400010]
 IpAddr: 2001:DB8::40:10
```

When MD5 authentication is valid on with it is VRRP-E peer, the following syslog message is displayed:

```
Aug 10 18:48:51 VRRP6: Port 2/6, VRID 2 - rcvd advertisement from 200:160::40:1
```

```
Ver:3 Type:1 Vrid:2 Pri:255 #IP:1 AuthType:2 Adv:1 Chksum:0x0000
        HMAC-MD5 CODE:[00000000000000000000400010]
IpAddr: 200:160::40:10
```

NOTE
Using md5-authentication implies that the software need not run checksum verification on the receiving router, and can rely on the authentication code (message digest 5 algorithm) to verify the integrity of the VRRP-E message header.

Suppressing RIP advertisements on backup routers for the backup up interface

Normally, a VRRP or VRRP-E backup includes route information for the virtual IP address in RIP advertisements. As a result, other routers receive multiple paths for the backup router and might sometimes unsuccessfully use the path to the backup router rather than the path to the Master.

You can prevent the backup routers from advertising route information for the interface on which they are defined by enabling suppression of the advertisements.

To suppress RIP advertisements for interface on which a backup router is defined in Router2, enter the following commands.

```
device(config)# router rip
device(config-rip-router)# use-vrrp-path
```

Syntax: [no] use-vrrp-path

The syntax is the same for VRRP and VRRP-E.

Hello interval

The Master periodically sends Hello messages to the Backups. The Backups use the Hello messages as verification that the Master is still on-line. If the Backup routers stop receiving the Hello messages for the period of time specified by the Dead interval, the Backup routers determine that the Master router is dead. At this point, the Backup router with the highest priority becomes the new Master router.

The Dead interval is internally derived from the Hello Interval, by default. It is equal to 3 times the Hello interval + Skew time, where Skew time is equal to (256 - priority) divided by 256. Generally, if you change the Hello interval, you also should change the Dead interval on the Backup routers.To change the Hello interval on the Master to 10 seconds for VRRPv2 and VRRP-Ev2, enter the following commands.

```
device(config)# interface ethernet 1/6
device(config-if-e10000-1/6)# ip vrrp vrid 1
device(config-if-e10000-1/6-vrid-1)# hello-interval 10
```

Syntax: [no] hello-interval value

The Hello interval can be from 1 through 84 seconds. The default is 1 second.

To change the Hello interval on the Master to 200 milliseconds for VRRPv3, enter the following commands.

```
device(config)# interface ethernet 1/6
device(config-if-e10000-1/6)# ip vrrp vrid 1
device(config-if-e10000-1/6-vrid-1)# hello-interval msec 200
```

Syntax: [no] hello-interval [**msec**] *value*

The **Hello-interval** can be from 100 through 84000 milliseconds. The default is 1000 milliseconds.

The **Hello-interval** syntax is the same for VRRP and VRRP-E.

NOTE
VRRP-E does not support the Hello interval timer if it is set to milliseconds.

Dead interval

The **Dead interval** is the number of seconds a Backup waits for a Hello message from the Master before determining that the Master is dead. When Backups determine that the Master is dead, the Backup with the highest priority becomes the new Master. The Dead interval can be from 1 - 84 seconds. The default is internally derived by software. It is equal to 3 times the Hello interval + Skew time, where Skew time is equal to (256 - priority) divided by 256. To change the Dead interval on a Backup to 30 seconds for VRRPv2 and VRRP-Ev2, enter the following commands.

```
device(config)# interface ethernet 1/5
device(config-if-e10000-1/5)# ip vrrp vrid 1
device(config-if-e10000-1/5-vrid-1)# dead-interval 30
```

Syntax: [no] dead-interval*value*

To change the **Dead-interval** on a Backup to 600 milliseconds for VRRPv3, enter the following commands.

```
device(config)# interface ethernet 1/5
device(config-if-e10000-1/5)# ip vrrp vrid 1
device(config-if-e10000-1/5-vrid-1)# dead-interval msec 600
```

Syntax: [no] dead-interval [**msec**] *value*

The **Dead-interval** can be from 100 through 84000 milliseconds. The default is 3500 milliseconds.

The **dead-interval** syntax is the same for VRRP and VRRP-E.

NOTE
VRRP-E does not support the Dead interval timer if it is set to milliseconds.

Backup hello message state and interval

By default, **Backup** do not send Hello messages to advertise themselves to the Master. You can enable these messages if desired and also change the message interval.

To enable a Backup to send Hello messages to the Master, enter the following commands.

```
device(config)# router vrrp
device(config)# interface ethernet 1/6
device(config-if-e10000-1/6)# ip vrrp vrid 1
device(config-if-e10000-1/6-vrid-1)# advertise backup
```

Syntax: [no] advertise backup

When you enable a Backup to send Hello messages, the Backup sends a Hello messages to the Master every 60 seconds by default. You can change the interval to be up to 3600 seconds. To do so, enter the following commands.

```
device(config)# router vrrp
device(config)# interface ethernet 1/6
```

```
device(config-if-e10000-1/6)# ip vrrp vrid 1
device(config-if-e10000-1/6-vrid-1)# backup-hello-interval 180
```

Syntax: [no] backup-hello-interval num

The *num* parameter specifies the message interval and can be from 60 through 3600 seconds. The default is 60 seconds.

The syntax is the same for VRRP and VRRP-E.

Track port

You can configure the virtual router to track the link state of interfaces on the Brocade device. This capability is quite useful for tracking the state of the exit interface for the path for which the virtual router is providing redundancy. Refer to Track ports and track priority on page 692.

To configure 1/6 on Router1 to track interface 2/4, enter the following commands.

```
device(config)# interface ethernet 1/6
device(config-if-e10000-1/6)# ip vrrp vrid 1
device(config-if-e10000-1/6-vrid-1)# track-port ethernet 2/4
```

Syntax: [no] track-port ethernet slot/portnum | **ve num** [**priority num**]

The syntax is the same for VRRP and VRRP-E.

Track priority

If you configure a virtual router to track the link state of interfaces and one of the tracked interface goes down, the software changes the VRRP or VRRP-E priority of the virtual router:

- For VRRP, the software changes the priority of the virtual router to a track priority that is lower than that of the virtual router priority and lower than the priorities configured on the Backups. For example, if the virtual router priority is 100 and a tracked interface with track priority 60 goes down, the software changes the virtual router priority to 60.
- For VRRP-E, the software reduces the virtual router priority by the amount of the priority of the tracked interface that went down. For example, if the VRRP-E interface's priority is 100 and a tracked interface with track priority 60 goes down, the software changes the VRRP-E interface's priority to 40. If another tracked interface goes down, the software reduces the virtual router's priority again, by the amount of the tracked interface's track priority.

The default track priority for a VRRP Owner is 2. The default track priority for Backups is 1.

You enter the track priority as a parameter with the **owner** or **backup** command. Refer to Track port on page 709.

Syntax: [no] owner [**track-priority value**]

Syntax: [no] backup [**priority value**] [**track-priority value**]

The syntax is the same for VRRP and VRRP-E.

Backup preempt

By default, a Backup that has a higher priority than another Backup that has become the Master can preempt the Master, and take over the role of Master. If you want to prevent this behavior, disable preemption.

Preemption applies only to Backups and takes effect only when the Master has failed and a Backup has assumed ownership of the virtual router. The feature prevents a Backup with a higher priority from

taking over as Master from another Backup that has a lower priority but has already become the Master of the virtual router.

Preemption is especially useful for preventing flapping in situations where there are multiple Backups and a Backup with a lower priority than another Backup has assumed ownership, since the Backup with the higher priority was unavailable when ownership changed.

If you enable the non-preempt mode (thus disabling the preemption feature) on all the Backups, the Backup that becomes the Master following the disappearance of the Master continues to be the Master. The new Master is not preempted.

NOTE
In VRRP, regardless of the setting for the preempt parameter, the Owner always returns to be the Master when it comes back online.

To disable preemption on a Backup, enter the following commands.

```
device(config)# interface ethernet 1/6
device(config-if-e10000-1/6)# ip vrrp vrid 1
device(config-if-e10000-1/6-vrid-1)# non-preempt-mode
```

Syntax: [no] non-preempt-mode

The syntax is the same for VRRP and VRRP-E.

Master router abdication and reinstatement

To change the Master's priority, enter the following commands.

```
device(config)# interface ethernet 1/6
device(config-if-e10000-1/6)# ip vrrp vrid 1
device(config-if-e10000-1/6-vrid-1)# owner priority 99
```

Syntax: [no] owner priority | **track-priority num**

The *num* parameter specifies the new priority and can be a number from 1 through 254.

When you press Enter, the software changes the priority of the Master to the specified priority. If the new priority is lower than at least one Backup's priority for the same virtual router, the Backup takes over and becomes the new Master until the next software reload or system reset.

To verify the change, enter the following command from any level of the CLI.

```
device(config-if-e10000-1/6-vrid-1)# show ip vrrp
Total number of VRRP routers defined: 1
Interface ethernet 1/6
auth-type no authentication
VRID 1
state backup
administrative-status enabled
mode owner
priority 99
current priority 99
hello-interval 1 sec
ip-address 10.53.5.1
backup routers 10.53.5.2
```

This example shows that even though this Brocade device is the Owner of the virtual router ("mode owner"), the Brocade device's priority for the virtual router is only 99 and the state is now "backup" instead of "active". In addition, the administrative status is "enabled".

To change the Master's priority back to the default Owner priority 255, enter "**no**" followed by the command you entered to change the priority. For example, to change the priority of a VRRP Owner back to 255 from 99, enter the following command.

```
device(config-if-e10000-1/6-vrid-1)# no owner priority 99
```

You cannot set the priority to 255 using the **owner priority** command.

VRRP-extended slow start

In a VRRP-E configuration, if a Master router goes down, the Backup router with the highest priority takes over after expiration of the dead interval. When the original Master router comes back up again, it takes over from the Backup router (which became the Master router when the original Master router went down). By default, this transition from Backup router back to Master router takes place after expiration of the dead interval.

You can configure the VRRP-E slow start timer feature, which causes a specified amount of time to elapse between the time the original Master router is restored and when it takes over from the Backup router (This range is currently set to between 1-57600 seconds). This interval allows time for OSPF convergence when the Master is restored.

You can use the use-track-port and restart options to implement the slow start timer upon track port state changes. The use-track-port option implements a slow start timer for the first track port "up" state change, in addition to the VRRP-E initialization state. The restart option restarts the slow-start timer for subsequent track port "up" state changes.

To set the VRRP-E slow start timer to 30 seconds, enter the following command.

```
device(config)# router vrrp-extended
device(config-vrrpe-router)# slow-start 30
```

Syntax: [no] slow-start seconds [**use-track-port** [**restart**]]

When the VRRP-E slow start timer is enabled, if the Master router goes down, the Backup router takes over after expiration of the dead interval. If the original Master router subsequently comes back up again, the amount of time specified by the VRRP-E slow start timer elapses (in this example, 30 seconds) before the original Master router takes over from the Backup router (which became the Master router when the original Master router went down).

In the event that no other routers are currently Master, the router will immediately (after the dead-interval) become Master.

Without the use-track-port option, the slow start timer will be started only for the VRRP-E router initialization, not for the track port state change.

NOTE
If you change backup priority of VRRP-e Backup router to be higher than Master router, the slow start timer will not work. The original Master router will take over from the Backup router immediately.

VRRP-extended scale timer

This feature allows you to increase timing sensitivity across all configured or default VRRP-Extended timers. When this command is used, all configured or default VRRP-Extended timers are divided by the value set in the command. For example: a value of 10 divides the timers by a factor of 10. Configuring a value of 10 in a network with all VRRP-Extended values set to their defaults would cause VRRP-Extended instances to send packets every 100ms (instead of every 1 second) and the backup advertisement interval of 60 seconds would be modified to 6 seconds. All other timers would be divided

likewise. This would allow VRRP-Extended instances to converge within a second in the event of a VRRP-Extended master failure (this is since the default dead timer would be 300 ms).

To scale all VRRP-Extended timers by 10, use the following command.

```
device(config)# scale-timer vrrp-extended 10
```

Syntax: [no] scale-timer vrrp-extended scale-factor

The *scale-factor* variable is the number that all VRRP-Extended timers values are divided by. Values can be set from 1 through 10.

NOTE
Increased timing sensitivity as a result of using this command could cause protocol flaps during times of network congestion.

NOTE
This command is not supported in VRRP v2 and VRRP v3.

Enable and Disable password display

By default, the MD5 authentication password key is displayed as dots (...) for in the **show running-configuration** and **show startup-configuration** commands.

For example:

```
device# show running-config
interface ethernet 1/1
...
ipv6 vrrp-extended auth-type md5-auth ********
...
```

Use the **enable password-display** command to display the key password in original form, either encrypted or decrypted.

For example:

```
device# enable password-display
device# show running-config
interface ethernet 1/1
...
ipv6 vrrp-extended auth-type md5-auth 2 $bkciYg==
...
```

Use the **disable password-display** command to display the key password as dots.

Multiple virtual IP address support for VRRP-E

Support for multiple virtual IPv4 addresses per interface is available for a VRRP extended (VRRP-E) router instance. For an interface that has multiple subnet IPv4 addresses, a single VRRP-E provides redundancy when all the IP addresses are configured for the VRRP-E instance.

All virtual IP addresses must be configured before activating the VRRP-E instance. If a virtual IP address is to be added, the VRRP-E router instance must be disabled, and the configuration changes made before reactivating the VRRP-E instance. Between master and backup VRRP-E devices the

virtual IP address configuration must be consistent to allow the generation of the same virtual MAC address for a specific virtual router ID (VRID). Gratuitous ARP messages are sent for all the configured IP addresses of each VRRP-E instance. To avoid regeneration of the virtual MAC address when a lower value virtual IPv4 address is configured, configure a static virtual MAC address for the VRRP-E router instance.

NOTE
Multiple virtual IPv4 address support is not compatible with the VRRP-E scaling feature.

Configuring multiple virtual IP addresses for VRRP-E

Configuring multiple virtual IPv4 addresses for an interface assigned as a VRRP extended (VRRP-E) router instance.

1. From privileged EXEC mode, enter configuration mode by issuing the **configure terminal** command.

```
device# configure terminal
```

2. Globally enable the VRRP-E protocol by entering the **router vrrp-extended** command.

```
device(config)# router vrrp-extended
```

3. Configure the ethernet interface for the VRRP-E instance.

```
device(config)# interface ethernet 1/1
```

4. Configure an IP address on the interface.

```
device(config-if-e1000-1/1)# ip address 10.10.10.1/24
```

5. Repeat Step 4 for all IP addresses on the interface.

```
device(config-if-e1000-1/1)# ip address 10.20.20.1/24
```

6. Configure a VRRP extended instance using a virtual routing ID (VRID).

```
device(config-if-e1000-1/1)# ip vrrp-extended vrid 2
```

7. Configure the device as a backup VRRP-E device for VRID 2.

```
device(config-if-e1000-1/1-vrid-2)# backup
```

8. Configure an optional static MAC address for VRID 2.

```
device(config-if-e1000-1/1-vrid-2)# virtual-mac aaaa.bbbb.cccc
```

9. Configure a virtual IP address for VRID 2.

```
device(config-if-e1000-1/1-vrid-2)# ip-address 10.10.10.10
```

10. Repeat Step 9 to configure all the virtual IP addresses for VRID 2.

```
device(config-if-e1000-1/1-vrid-2)# ip-address 10.20.20.20
```

If you want to add a new virtual IP address after the initial setup, you must first deactivate the virtual router and then reconfigure all the virtual IP addresses.

11. Activate the virtual IP address or addresses for VRID 2.

```
device(config-if-e1000-1/1-vrid-2)# activate
```

The following example configures seven virtual IP addresses for VRRP-E virtual router instance 2.

```
device# configure terminal
device(config)# router vrrp-extended
device(config)# interface ethernet 1/1
device(config-if-e1000-1/1)# ip address 10.10.10.1/24
device(config-if-e1000-1/1)# ip address 10.20.20.1/24
device(config-if-e1000-1/1)# ip address 10.30.30.1/24
device(config-if-e1000-1/1)# ip address 10.40.40.1/24
device(config-if-e1000-1/1)# ip address 10.50.50.1/24
device(config-if-e1000-1/1)# ip address 10.60.60.1/24
device(config-if-e1000-1/1)# ip address 10.70.70.1/24
device(config-if-e1000-1/1)# ip vrrp-extended vrid 2
device(config-if-e1000-1/1-vrid-2)# backup
device(config-if-e1000-1/1-vrid-2)# virtual-mac aaaa.bbbb.cccc
device(config-if-e1000-1/1-vrid-2)# ip-address 10.10.10.10
device(config-if-e1000-1/1-vrid-2)# ip-address 10.20.20.20
device(config-if-e1000-1/1-vrid-2)# ip-address 10.30.30.30
device(config-if-e1000-1/1-vrid-2)# ip-address 10.40.40.40
device(config-if-e1000-1/1-vrid-2)# ip-address 10.50.50.50
device(config-if-e1000-1/1-vrid-2)# ip-address 10.60.60.60
device(config-if-e1000-1/1-vrid-2)# ip-address 10.70.70.70
device(config-if-e1000-1/1-vrid-2)# activate
```

Displaying multiple virtual IP addresses for VRRP-E information

Displays information about the multiple virtual IPv4 addresses that are configured for a VRRP-E instance.

Several options of the **show ip vrrp-extended** command can display information about the multiple virtual IP addresses configured for a VRRP-E instance. Use the steps below in any order.

1. Enter the **show ip vrrp-extended brief** command to display the number of configured virtual IPv4 addresses for each VRRP-E router instance and the virtual IPv4 addresses.

```
device# show ip vrrp-extended brief

Total number of VRRP-Extended routers defined: 3
Flags Codes - P:Preempt 2:V2 3:V3
Short-Path-Fwd Codes - ER: Enabled with revertible option, RT: reverted,
                       NR: not reverted

Intf VRID Curr Flags State  MasterIP BackupIP (No) VirtualIP Short-   Track MCT
          Prio              Address  Address       Address   Path-Fwd VPLS-State
-------------------------------------------------------------------------------
1/1  1    100  P2    Master Local    Unknown  ( 7) 10.10.10.10 Enabled Disable
                                                   10.20.20.20
                                                   10.30.30.30
                                                   10.40.40.40
                                                   10.50.50.50
                                                   10.60.60.60
                                                   10.70.70.70
```

2. Enter the **show ip vrrp-extended** command with a specific VRID and interface to display the number of configured IPv4 addresses for each VRRP-E router instance and the actual IPv4 addresses in a different format with the detailed information about each virtual router instance.

```
device# show ip vrrp-extended vrid 1 ethernet 1/1

Interface 1/1
----------------
 auth-type no authentication
 VRID 1 (index 1)
 interface 1/1
 state master
 administrative-status enabled
 mode non-owner(backup)
 virtual mac 02e0.527d.7c01
```

```
priority 100
current priority 100
track-priority 5
hello-interval 1 sec
backup hello-interval 60 sec
slow-start timer (configured) 0 sec
advertise backup disabled
dead-interval 3500 ms
preempt-mode true
number of configured virtual address 7
virtual ip address 10.10.10.10 10.20.20.20 10.30.30.30 10.40.40.40
                   10.50.50.50 10.60.60.60 10.70.70.70
next hello sent in 1000 ms
Track MCT-VPLS-State: Disable
short-path-forwarding enabled
```

VRRP-E scaling using logical groups

Scaling the number of VRRP extended (VRRP-E) instances up to 4000 instances is allowed using a grouping mechanism. VRRP-E instances are configured into logical groups consistently across all the VRRP-E master and backup devices.

Each VRRP-E logical group is assigned with a group master and the configuration is performed to identify the group master for each VRRP-E virtual router ID (VRID) configured on a device. Group members stop advertising hello messages and they inherit the state of the group master. The significant reduction in load on the CPU when the hello advertisements are not sent or processed, allows more VRRP-E instances to be configured.

A gratuitous ARP router advertisement is sent from the group-master on behalf of its group members once every 30 seconds by default to allow the devices on the network to learn the virtual MAC address of the group master. The gratuitous ARP router advertisement interval can be configured.

NOTE
If the maximum number of VRRP-E instances exist, configuring a gratuitous ARP router advertisement interval lower than 5 seconds can result in high CPU usage.

The configuration for the following commands is inherited from the group master device configuration. VRRP-E ignores the configuration of any parameter that is to be inherited from the group master.

- advertise
- auth-type
- backup
- backup-hello-interval
- dead-interval
- hello-interval
- non-preempt-mode
- use-v2-checksum

The configuration for the following commands is not inherited from the group master. Group members retain their individual configuration parameters from these commands.

- activate
- enable
- ip-address
- ipv6-address
- short-path-forwarding
- virtual-mac

NOTE
The VRRP-E scaling feature is not compatible with the VRRP-E multiple virtual IP addresses feature.

Configuring VRRP-E scaling

Configuring VRRP-E instances into logical groups using a group master and configuring a gratuitous ARP router advertisement interval allows the number of VRRP-E instances to scale up to 4000 instances.

This task has to be repeated for all VRRP-E instances that are to be part of the same group. VRRP-E instances are always physically grouped on the same device and the logical group is maintained consistently across all devices. VRRP-E scaling is supported for IPv4 and IPv6 VRRP-E sessions.

NOTE
The VRRP-E scaling feature is not compatible with the VRRP-E multiple virtual IP addresses feature.

1. From privileged EXEC mode, enter configuration mode by issuing the **configure terminal** command.

```
device# configure terminal
```

2. Globally enable the IPv6 VRRP-E protocol by entering the **ipv6 router vrrp-extended** command.

```
device(config)# ipv6 router vrrp-extended
```

3. Configure the gratuitous ARP interval for the IPv6 VRRP-E instance.

```
device(config-ipv6-VRRP-E-router)# garp-ra-interval 90
```

NOTE
In IPv4 VRRP-E the router prompt does not display the “-VRRP-E-router” text.

4. Configure the ethernet interface for the IPv6 VRRP-E instance.

```
device(config-ipv6-VRRP-E-router)# interface ethernet 1/5
```

5. Configure an IPv6 address on the interface.

```
device(config-if-e1000-1/5)# ipv6 address 3013::2/64
```

6. Configure an IPv6 VRRP extended instance using a virtual routing ID (VRID).

```
device(config-if-e1000-1/5)# ipv6 vrrp-extended vrid 2
```

7. Configure the device as a backup VRRP-E device for VRID 2.

```
device(config-if-e1000-1/5-ipv6-vrid-2)# backup priority 50 track-priority 20
```

8. Configure a virtual link-local IPv6 address for VRID 2.

```
device(config-if-e1000-1/5-ipv6-vrid-2)# ipv6-address fe80::768e:f8ff:fe2a:0099
```

9. Configure a virtual IPv6 address for VRID 2.

```
device(config-if-e1000-1/5-ipv6-vrid-2)# ipv6-address 3013::99
```

10. Configures the VRRP-E master for a group for VRID 2.

```
device(config-if-e1000-1/5-ipv6-vrid-2)# group-master interface ethernet 1/2 vrid
1
```

In this example,

11 Activate the virtual IP address or addresses for VRID 2.

```
device(config-if-e1000-1/5-ipv6-vrid-2)# activate
```

The following example configures virtual router 1 on interface Ethernet 1/2 as the VRRP-E group master of virtual router 2 on interface Ethernet 1/5. Virtual router 2 will stop sending and receiving VRRP-E packets because VRID 1 now maintains the session for virtual router 2 by sending and receiving VRRP-E packets. The interval between gratuitous ARP messages is set to 90 seconds.

```
device# configure terminal
device(config)# router ipv6 vrrp-extended
device(config-ipv6-VRRP-E-router)# garp-ra-interval 90
device(config-ipv6-VRRP-E-router)# interface ethernet 1/5
device(config-if-e1000-1/5)# ipv6 address 3013::2/64
device(config-if-e1000-1/5)# ipv6 vrrp-extended vrid 2
device(config-if-e1000-1/5-ipv6-vrid-2)# backup priority 50 track-priority 20
device(config-if-e1000-1/5-ipv6-vrid-2)# ipv6-address fe80::768e:f8ff:fe2a:0099
device(config-if-e1000-1/5-ipv6-vrid-2)# ipv6-address 3013::99
device(config-if-e1000-1/5-ipv6-vrid-2)# group-master interface ethernet 1/2 vrid 1
device(config-if-e1000-1/5-ipv6-vrid-2)# activate
```

Displaying VRRP-E scaling information

Displays information about VRRP-E group members and group masters configured for the VRRP-E scaling feature.

Several options of the **show ip vrrp-extended** or **show ipv6 vrrp-extended** commands can display information about the VRRP-E scaling feature configuration. Use the steps below in any order.

1. Enter the **show ip vrrp-extended** or the **show ipv6 vrrp-extended** command with the **vrid** option to display group member information for the VRRP-E scaling feature for a specific VRID. In the example below, output for VRID 1 shows the total number of group members and in which VRIDs the members are configured.

```
device(config)# show ip vrrp-extended vrid 1

VRID 1 (index 1)
 interface 1/1
 state master
. administrative-status enabled
.
.
 group-member count 3
 group-members
   ethernet 1/2 vrid 2
   ethernet 1/2 vrid 3
   ethernet 1/2 vrid 4
```

2. Enter the **show ipv6 vrrp-extended** or the **show ip vrrp-extended** command with a specific VRID and interface to display group master information for the VRRP-E scaling feature for a specific interface. Only partial output is displayed.

```
device# show ipv6 vrrp-extended ve 100 vrid 2

VRID 2 (index 2)
 interface v100
 state backup
.
.
.
 group-master ve 100 vrid 1
```

Alternate VRRPv2 checksum for VRRPv3 IPv4 sessions

If VRRPv3 is configured on a Brocade device in a network with third-party peering devices using VRRPv2-style checksum calculations for IPv4 VRRPv3 sessions, a VRRPv2-style checksum must be configured for VRRPv3 IPv4 sessions on a Brocade device.

VRRPv3 introduced a new checksum method for both IPv4 and IPv6 sessions and this version 3 checksum computation is enabled by default. To accommodate third-party devices that still use a VRRPv2-style checksum for IPv4 VRRPv3 sessions, a command-line interface (CLI) command is available for configuration on a Brocade device. The new checksum method is disabled by default and only applicable to IPv4 VRRPv3 sessions. If configured for VRRPv2 sessions, the VRRPv2-style checksum CLI is accepted, but it has no effect.

Enabling the v2 checksum computation method in a VRRPv3 IPv4 session

Configuring an alternate VRRPv2-style checksum in a VRRPv3 IPv4 session for compatibility with third-party network devices.

VRRPv3 uses the v3 checksum computation method by default for both IPv4 and IPv6 sessions on Brocade devices. Third-party devices may only have a VRRPv2-style checksum computation available for a VRRPv3 IPv4 session. The **use-v2-checksum** command is entered in interface configuration mode.

1. Use the **configure** command to enter global configuration mode.

```
device# configure
```

2. To enable VRRP globally enter the **router vrrp** command.

```
device(config)# router vrrp
```

3. Enter the **interface** command with an interface type and number.

```
device(config)# interface ethernet 2/4
```

4. To configure a VRRP virtual routing ID use the **ip vrrp vrid** command with an associated ID number.

```
device(config-if-e1000-2/4)# ip vrrp vrid 14
```

5. To enable VRRP version 3 (VRRPv3) enter the **version** command with version number of v3.

```
device(config-if-e1000-2/4-vrid-14)# version v3
```

6. To enable v2 checksum computation method in an IPv4 VRRPv3 session, use the **use-v2-checksum** command in the VRRP configuration mode.

```
device(config-if-e1000-2/4-vrid-14)# use-v2-checksum
```

7. Enter the IP address for the interface using the **ip-address** command.

```
device(config-if-e1000-2/4-vrid-14)# ip-address 10.14.14.99
```

8. To activate the interface, enter the **activate** command.

```
device(config-if-e1000-2/4-vrid-14)# activate
```

The following example shows the v2 checksum computation method enabled for an VRRPv3 IPv4 session on a Brocade device.

```
device# config
device(config)# router vrrp
device(config)# ethernet 2/4
device(config-if-e1000-2/4)# ip vrrp vrid 14
device(config-if-e1000-2/4-vrid-14)# version v3
device(config-if-e1000-2/4-vrid-14)# use-v2-checksum
device(config-if-e1000-2/4-vrid-14)# ip-address 10.14.14.99
device(config-if-e1000-2/4-vrid-14)# activate
```

Displaying alternate v2 checksum settings

Displaying the alternate VRRPv2-style checksum for VRRPv3 IPv4 sessions.

The following steps are both optional and can be used to verify that the alternate VRRPv2-style checksum computation command, **use-v2-checksum**, has been set for VRRPv3 IPv4 sessions.

1. Use the **show running-config** command to verify that the **use-v2-checksum** command has been configured for a specified interface. Only part of the output is displayed below.

```
device# show running-config

interface ethernet 2/4
 ip address 10.14.14.2/24
 ip vrrp vrid 14
  backup
  ip-address 10.14.14.99
  use-v2-checksum
  exit
```

2. Use the **show ip vrrp** command with a virtual router ID number to display the current settings of a specific VRRP session including the **use-v2-checksum** command, if configured.

```
device# show ip vrrp vrid 14

Interface 2/4
----------------
auth-type no authentication

VRID 14 (index 1)
 interface 2/4
 state initialize
 administrative-status disabled
 version v3 - use-v2-checksum
 mode non-owner(backup)
 virtual mac 0000.5e00.010e
 priority 100
 current priority 100
 track-priority 1
 hello-interval 1 sec
 backup hello-interval 60 sec
 slow-start timer (configured) 0 sec
 advertise backup disabled
 dead-interval 3500 ms
 preempt-mode true
 ip-address 10.14.14.99
```

Displaying VRRP and VRRP-E information for IPv4

You can display the following information for IPv4 VRRP or VRRP-E:

- Displaying summary information on page 720
- Displaying detailed information on page 721
- Displaying statistics on page 725
- Displaying configuration information for VRRP and VRRP-E on page 727

Displaying summary information

To display summary information for IPv4 VRRP or VRRP-E, enter the following command at any level of the CLI.

```
device(config)# show ip vrrp brief
Total number of VRRP routers defined: 2
Flags Codes - P:Preempt 2:V2 3:V3 S:Short-Path-Fwd
Inte- VRID Current  Flags  State  Master IP       Backup IP       Virtual IP
rface      Priority              Address         Address         Address
-------------------------------------------------------------------------------
1/1   10   255      P2-    Master Local           Unknown         10.30.30.2
1/3   13   100      P2-    Master Local           Unknown         10.13.13.3
```

Syntax: show ip vrrp [**brief** | **ethernet slot/portnum** | **statistics** | **ve num** | **vrid** *id*]

Syntax: show ip vrrp-extended [**brief** | **ethernet slot/portnum** | **statistics** | **ve num** | **vrid** *id*]

The **brief** parameter displays the summary information. If you do not use this parameter, detailed information is displayed instead. Refer to Displaying detailed information on page 721.

The **ethernet***slot*/*portnum* parameter specifies an Ethernet port. If you use this parameter, the command displays IPv4 VRRP or VRRP-E information only for the specified port.

The **ve***num* parameter specifies a virtual interface. If you use this parameter, the command displays IPv4 VRRP or VRRP-E information only for the specified virtual interface.

The **vrid***id* parameter specifies a virtual router ID. If you use this parameter, the command displays IPv4 VRRP or VRRP-E information only for the specified virtual router.

The **statistics** parameter displays statistics. Refer to Displaying statistics on page 725.

This display shows the following information.

TABLE 156 CLI display of VRRP or VRRP-E summary information

This field...	Displays...
Total number of VRRP (or VRRP-Extended) routers defined	The total number of virtual routers configured on this Brocade device. **NOTE** The total applies only to the protocol the Brocade device is running. For example, if the Brocade device is running VRRP-E, the total applies only to VRRP-E routers.
Interface	The interface on which VRRP or VRRP-E is configured. If VRRP or VRRP-E is configured on multiple interfaces, information for each interface is listed separately.
VRID	The ID of the virtual router configured on this interface. If multiple virtual routers are configured on the interface, information for each virtual router is listed in a separate row.
Current Priority	The current VRRP or VRRP-E priority of this Brocade device for the virtual router.

TABLE 156 CLI display of VRRP or VRRP-E summary information (Continued)

This field...	Displays...
Flags	Whether the backup preempt mode is enabled. If the backup preempt mode is enabled, this field contains a "P". If the mode is disabled, this field is blank. P:Preempt 2:V2 3:V3 2: implies VRRP Version2 3: implies VRRP Version3.
Short-Path-Fwd	Displays information about whether VRRP-E Extension for Server Virtualization is enabled or disabled.
State	This Brocade device's VRRP or VRRP-E state for the virtual router. The state can be one of the following: • Init - The virtual router is not enabled (activated). If the state remains Init after you activate the virtual router, make sure that the virtual router is also configured on the other routers and that the routers can communicate with each other. **NOTE** If the state is Init and the mode is incomplete, make sure you have specified the IP address for the virtual router. • Backup - This Brocade device is a Backup for the virtual router. • Master - This Brocade device is the Master for the virtual router.
Master addr	The IP address of the router interface that is currently the Master for the virtual router.
Backup addr	The IP addresses of the router interfaces that are currently Backups for the virtual router.
VIP	The virtual IP address that is being backed up by the virtual router.

Displaying detailed information

To display detailed information for IPv4 VRRP or VRRP-E, enter the following command at any level of the CLI.

```
device(config)# show ip vrrp-extended
Total number of vrrp-extended routers defined: 1
Interface v10
----------------
auth-type no authentication
VRID 10 (index 1)
interface v10
state backup
administrative-status enabled
mode non-owner(backup)
virtual mac 02e0.52a0.c00a
priority 50
current priority 50
track-priority 5
hello-interval 1 sec
backup hello-interval 60 sec
slow-start timer (configured) 30 sec
advertise backup disabled
dead-interval 3600 ms
```

```
preempt-mode true
virtual ip address 10.10.10.254
next hello sent in 1000ms
track-port 1/1 (up)
master router 10.10.10.4 expires in 3.1 sec
short-path-forwarding enabled
```

Syntax: show ip vrrp

Syntax: show ip vrrp-extended

This display shows the following information.

TABLE 157 CLI display of VRRP or VRRP-E detailed information

This field...	Displays...
Total number of VRRP (or VRRP-Extended) routers defined	The total number of virtual routers configured on this Brocade device. **NOTE** The total applies only to the protocol the Brocade device is running. For example, if the Brocade device is running VRRP-E, the total applies only to VRRP-E routers.
Interface parameters	
Interface	The interface on which VRRP or VRRP-E is configured. If VRRP or VRRP-E is configured on multiple interfaces, information for each interface is listed separately.
auth-type	The authentication type enabled on the interface.
Virtual Router parameters	
VRID	The virtual router configured on this interface. If multiple virtual routers are configured on the interface, information for each virtual router is listed separately.
state	This Brocade device's VRRP or VRRP-E state for the virtual router. The state can be one of the following: • initialize - The virtual router is not enabled (activated). If the state remains "initialize" after you activate the virtual router, make sure that the virtual router is also configured on the other routers and that the routers can communicate with each other. **NOTE** If the state is "initialize" and the mode is incomplete, make sure you have specified the IP address for the virtual router. • backup - This Brocade device is a Backup for the virtual router. • master - This Brocade device is the Master for the virtual router.
administrative-status	The administrative status of the virtual router. The administrative status can be one of the following: • disabled - The virtual router is configured on the interface but VRRP or VRRP-E has not been activated on the interface. • enabled - VRRP or VRRP-E has been activated on the interface.

TABLE 157 CLI display of VRRP or VRRP-E detailed information (Continued)

This field...	Displays...
mode	Indicates whether the Brocade device is the Owner or a Backup for the virtual router. **NOTE** If "incomplete" appears after the mode, configuration for this virtual router is incomplete. For example, you might not have configured the virtual IP address that is being backup up by the virtual router.This field applies only to VRRP. All Brocade devices configured for VRRP-E are Backups.
virtual MAC	The virtual IP MAC address that this virtual router is backing up.
priority	The device's preferability for becoming the Master for the virtual router. During negotiation, the router with the highest priority becomes the Master. If two or more devices are tied with the highest priority, the Backup interface with the highest IP address becomes the active router for the virtual router.
current priority	The current VRRP or VRRP-E priority of this Brocade device for the virtual router. The current priority can differ from the configured priority (refer the previous row) for the following reasons: • The virtual router is still in the initialization stage and has not become a Master or Backup yet. In this case, the current priority is 0. • The virtual router is configured with track ports and the link on a tracked interface has gone down. Refer to Track ports and track priority on page 692.
track priority	VRRP-E priority value assigned to the tracked port.
hello-interval	The number of seconds between Hello messages from the Master to the Backups for a given virtual router.
backup hello-interval	The number of seconds between Hello messages from a Backup to the Master.
advertise backup	The IP addresses of Backups that have advertised themselves to this Brocade device by sending Hello messages. **NOTE** Hello messages from Backups are disabled by default. You must enable the Hello messages on the Backup for the Backup to advertise itself to the current Master. Refer to Hello interval on page 707.
dead-interval	The current value of the dead interval. This is the value actually in use by this interface for the virtual router. **NOTE** This field does not apply to VRRP Owners.

TABLE 157 CLI display of VRRP or VRRP-E detailed information (Continued)

This field...	Displays...
preempt-mode	Whether the backup preempt mode is enabled. **NOTE** This field does not apply to VRRP Owners.
virtual ip address	The virtual IP addresses that this virtual router is backing up.
backup router *ip-addr* expires in *time*	The IP addresses of Backups that have advertised themselves to this Master by sending Hello messages. The *time* value indicates how long before the Backup expires. A Backup expires if you disable the advertise backup option on the Backup or the Backup becomes unavailable. Otherwise, the Backup's next Hello message arrives before the Backup expires. The Hello message resets the expiration timer. An expired Backup does not necessarily affect the Master. However, if you have not disabled the advertise backup option on the Backup, then the expiration may indicate a problem with the Backup. **NOTE** This field applies only when Hello messages are enabled on the Backups (using the advertise backup option).
next hello sent in *time*	How long until the Backup sends its next Hello message. **NOTE** This field applies only when this Brocade device is the Master and the Backup is configured to send Hello messages (the advertise backup option is enabled).
master router *ip-addr* expires in *time*	The IP address of the Master and the amount of time until the Master's dead interval expires. If the Backup does not receive a Hello message from the Master by the time the interval expires, either the IP address listed for the Master will change to the IP address of the new Master, or this Brocade device itself will become the Master. **NOTE** This field applies only when this Brocade device is a Backup.
track port	The interfaces that the virtual router's interface is tracking. If the link for a tracked interface goes down, the VRRP or VRRP-E priority of the virtual router interface is changed, causing the devices to renegotiate for Master. **NOTE** This field is displayed only if track interfaces are configured for this virtual router.

TABLE 157 CLI display of VRRP or VRRP-E detailed information (Continued)

This field...	Displays...
short-path-forwarding	Displays information about whether VRRP-E Extension for Server Virtualization is enabled or disabled.

Displaying statistics

To display IPv4 VRRP or VRRP-E statistics, enter the following command.

```
device# show ip vrrp-extended statistics
Global VRRP-Extended statistics
-------------------------------
- received vrrp-extended packets with checksum errors = 0
- received vrrp-extended packets with invalid version number = 0
- received vrrp-extended packets with unknown or inactive vrid = 1480
Interface v10
----------------
VRID 1
- number of transitions to backup state = 1
- number of transitions to master state = 1
- total number of vrrp-extended packets received = 0
  . received backup advertisements = 0
  . received packets with zero priority = 0
  . received packets with invalid type = 0
  . received packets with invalid authentication type = 0
  . received packets with authentication type mismatch = 0
  . received packets with authentication failures = 0
  . received packets dropped by owner = 0
  . received packets with ip ttl errors = 0
  . received packets with ip address mismatch = 0
  . received packets with advertisement interval mismatch = 0
  . received packets with invalid length = 0
- total number of vrrp-extended packets sent = 2004
  . sent backup advertisements = 0
  . sent packets with zero priority = 0
- received arp packets dropped = 0
- received proxy arp packets dropped = 0
- received ip packets dropped = 0
```

Syntax: show ip vrrp statistics

Syntax: show ip vrrp-extended statistics

The **statistics** parameter displays the following information.

The **received vrrp packets with checksum errors** shows the number of packets that is contained in checksum errors.

The **received vrrp packets with invalid version number** shows the number of packets with invalid versions.

The **received vrrp packets with unknown or inactive vrid** shows the number of packets that contain virtual routers that are not configured on the device or its interface

Displaying VRRP and VRRP-E information for IPv6

You can display the following information for IPv6 VRRP or VRRP-E:

- Displaying summary information on page 726
- Displaying detailed information on page 726

- Displaying statistics on page 727
- Displaying configuration information for VRRP and VRRP-E on page 727

Displaying summary information

To display summary information for IPv6 VRRP or VRRP-E, enter the following command at any level of the CLI.

```
device(config)# show ipv6 vrrp-extended brief
Total number of VRRP routers defined: 1
Flags Codes - P:Preempt 2:V2 3:V3 S:Short-Path-Fwd
Intf  VRID CurrPrio Flags State  Master-IPv6 Backup-IPv6 Virtual-IPv6
                                      -Address                -Address              -
Address
------------------------------------------------------------------
1/3   23   100      P3-                                Master
Local             3013::2                3013::99
```

Syntax: show ipv6 vrrp [**brief** | **ethernet slot/portnum** | **statistics** | **ve num** | **vrid** *id*]

Syntax: show ipv6 vrrp-extended [**brief** | **ethernet slot/portnum** | **statistics** | **ve num** | **vrid** *id*]

The **brief** parameter displays the summary information. If you do not use this parameter, detailed information is displayed instead. Refer to Displaying detailed information on page 721.

The **ethernet***slot*/*portnum* parameter specifies an Ethernet port. If you use this parameter, the command displays IPv6 VRRP or VRRP-E information only for the specified port.

The **ve***num* parameter specifies a virtual interface. If you use this parameter, the command displays IPv6 VRRP or VRRP-E information only for the specified virtual interface.

The **vrid***id* parameter specifies a virtual router ID. If you use this parameter, the command displays IPv6 VRRP or VRRP-E information only for the specified virtual router.

The **statistics** parameter displays statistics. Refer to Displaying statistics on page 725.

Displaying detailed information

To display detailed information for IPv6 VRRP or VRRP-E, enter the following command at any level of the CLI

```
device(config)# show ipv6 vrrp
Total number of VRRP routers defined: 1
Interface 1/3
---------------
auth-type no authentication
VRID 13 (index 2)
 interface 1/3
 state master
 administrative-status enabled
 version v3
 mode non-owner(backup)
 virtual mac 0000.5e00.0217
 priority 100
 current priority 100
 track-priority 1
 hello-interval 1000 ms
 backup hello-interval 60000 ms
 advertise backup disabled
 dead-interval 3000 ms
 preempt-mode true
 ipv6-address 3013::1
 next hello sent in 700 ms
 short-path-forwarding disabled
```

Syntax: show ipv6 vrrp

Syntax: show ipv6 vrrp-extended

Displaying statistics

To display IPv6 VRRP or VRRP-E statistics, enter the following command.

```
device# show ipv6 vrrp statistics
Global IPv6 VRRP statistics
-------------------------------
- received vrrp packets with checksum errors = 0
- received vrrp packets with invalid version number = 0
- received vrrp packets with unknown or inactive vrid = 0
Interface 1/3
----------------
VRID 13
- number of transitions to backup state = 1
- number of transitions to master state = 1
- total number of vrrp packets received = 0
. received backup advertisements = 19
. received packets with zero priority = 0
. received packets with invalid type = 0
. received packets with invalid authentication type = 0
. received packets with authentication type mismatch = 0
. received packets with authentication failures = 0
. received packets dropped by owner = 0
. received packets with ttl errors = 0
. received packets with ipv6 address mismatch = 0
. received packets with advertisement interval mismatch = 0
. received packets with invalid length = 0
- total number of vrrp packets sent = 1175
. sent backup advertisements = 0
. sent packets with zero priority = 0
- received neighbor solicitation packets dropped = 0
- received proxy neighbor solicitation packets dropped = 0
- received ipv6 packets dropped = 0
```

Syntax: show ipv6 vrrp statistics

Syntax: show ipv6 vrrp-extended statistics

Displaying configuration information for VRRP and VRRP-E

To display the current configuration information for IPv4 VRRP or VRRP-E and IPv6 VRRP or VRRP-E, enter the following command at any level of the CLI.

```
device(config-if-e10000-1/3)# show run
...
router vrrp
...
interface ethernet 1/1
 port-name Port111
 enable
 ip address 10.30.30.2/28
 ip vrrp vrid 10
  owner
  ip-address 10.30.30.2
  activate
!
...
interface ethernet 1/3
 enable
 ip address 10.13.13.2/24
 ip vrrp vrid 13
  version v3
  backup
  ip-address 10.13.13.3
```

```
  activate
...
```

Syntax: show run

Clearing VRRP or VRRP-E statistics

To clear IPv4 VRRP or VRRP-E statistics, enter the following command at the Privileged EXEC level or any configuration level of the CLI.

```
device(config)# clear ip vrrp statistics
```

Syntax: clear ip vrrp statistics

Syntax: clear ip vrrp-extended statistics

To clear IPv6 VRRP or VRRP-E statistics, enter the following command at the Privileged EXEC level or any configuration level of the CLI.

```
device(config)# clear ipv6 vrrp statistics
```

Syntax: clear ipv6 vrrp statistics

Syntax: clear ipv6 vrrp-extended statistics

Configuration examples

The following sections contain the CLI commands options for implementing the VRRP and VRRP-E configurations shown in Standard VRRP on page 688 and Overview of VRRP-E on page 694.

VRRP example for IPv4

To implement the IPv4 VRRP configuration shown in Standard VRRP on page 688, enter the following commands.

Configuring router1

To configure VRRP Router1, enter the following commands.

```
device(config)# router vrrp
device(config)# interface ethernet 1/6
device(config-if-e10000-1/6)# ip address 10.53.5.1
device(config-if-e10000-1/6)# ip vrrp vrid 1
device(config-if-e10000-1/6-vrid-1)# owner track-priority 20
device(config-if-e10000-1/6-vrid-1)# track-port ethernet 2/4
device(config-if-e10000-1/6-vrid-1)# ip-address 10.53.5.1
device(config-if-e10000-1/6-vrid-1)# activate
```

NOTE
When you configure the Master (Owner), the address you enter with the **ip-address** command must already be configured on the interface.

The **ip vrrp owner** command specifies that this router owns the IP address you are associating with the virtual router. Since this router owns the IP address, this router is the default Master router and its VRRP priority is thus 255.

NOTE
If VRRP is not configured globally, then you will see the response "Invalid input..." when you try to create a VRRP instance.

Configuring router2

To configure Router2 in Standard VRRP on page 688 after enabling VRRP, enter the following commands.

```
device(config)# router vrrp
device(config)# interface ethernet 1/5
device(config-if-e10000-1/5)# ip address 10.53.5.3
device(config-if-e10000-1/5)# ip vrrp vrid 1
device(config-if-e10000-1/5-vrid-1)# backup priority 100 track-priority 19
device(config-if-e10000-1/5-vrid-1)# track-port ethernet 3/2
device(config-if-e10000-1/5-vrid-1)# ip-address 10.53.5.1
device(config-if-e10000-1/5-vrid-1)# activate
```

The **backup** command specifies that this router is a VRRP Backup for virtual router VRID1. The IP address entered with the **ip-address** command is the same IP address as the one entered when configuring Router1. In this case, the IP address cannot also exist on Router2, but the interface on which you are configuring the virtual router Backup must have an IP address in the same subnet. By entering the same IP address as the one associated with this virtual router on the Owner, you are configuring the Backup to back up the address, but you are not duplicating the address.

NOTE
When you configure a Backup router, the router interface on which you are configuring the virtual router must have a real IP address that is in the same subnet as the address associated with the virtual router by the Owner. However, the address cannot be the same.

The **priority** parameter establishes the router's VRRP priority in relation to the other VRRP routers in this virtual router. The**track-priority** parameter specifies the new VRRP priority that the router receives for this virtual router if the interface goes down. Refer to Track ports and track priority on page 692.

The **activate** command activates the virtual router configuration on this interface. The interface does not provide backup service for the virtual IP address until you activate the VRRP configuration.

Syntax: router vrrp

Syntax: ip vrrp vrid vrid

Syntax: owner [**track-priority value**]

Syntax: backup [**priority value**] [**track-priority value**]

Syntax: track-port ethernet slot/portnum ve num

Syntax: ip-address ip-addr

Syntax: activate

VRRP example for IPv6

To implement the VRRP configuration for IPv6, enter the following commands.

Configuring router1

To configure VRRP Router1, enter the following commands.

```
device(config)# ipv6 router vrrp
device(config)# interface ethernet 1/6
device(config-if-e10000-1/6)# ipv6 address 1414:1414:1414::1/64
device(config-if-e10000-1/6)# ipv6 vrrp vrid 1
device(config-if-e10000-1/6-vrid-1)# owner track-priority 20
device(config-if-e10000-1/6-vrid-1)# track-port ethernet 2/4
device(config-if-e10000-1/6-vrid-1)# ipv6-address 1414:1414:1414::1/64
device(config-if-e10000-1/6-vrid-1)# activate
```

NOTE
When you configure the Master (Owner), the address you enter with the **ipv6-address** command must already be configured on the interface.

The **ipv6 vrrp owner** command specifies that this router owns the IP address you are associating with the virtual router. Since this router owns the IP address, this router is the default Master router and its VRRP priority is thus 255.

Configuring router2

To configure VRRP Router2, enter the following commands.

```
device(config)# ipv6 router vrrp
device(config)# interface ethernet 1/5
device(config-if-e10000-1/5)# ipv6 address 1414:1414:1414::2/64
device(config-if-e10000-1/5)# ipv6 vrrp vrid 1
device(config-if-e10000-1/5-vrid-1)# backup priority 100 track-priority 19
device(config-if-e10000-1/5-vrid-1)# track-port ethernet 3/2
device(config-if-e10000-1/5-vrid-1)# ipv6-address 1414:1414:1414::1/64
device(config-if-e10000-1/5-vrid-1)# activate
```

The **backup** command specifies that this router is a VRRP Backup for virtual router VRID1. The IP address entered with the **ipv6-address** command is the same IP address as the one entered when configuring Router1. In this case, the IP address cannot also exist on Router2, but the interface on which you are configuring the virtual router Backup must have an IP address in the same subnet. By entering the same IP address as the one associated with this virtual router on the Owner, you are configuring the Backup to back up the address, but you are not duplicating the address.

NOTE
When you configure a Backup router, the router interface on which you are configuring the virtual router must have a real IP address that is in the same subnet as the address associated with the virtual router by the Owner. However, the address cannot be the same.

The **priority** parameter establishes the router's VRRP priority in relation to the other VRRP routers in this virtual router. The **track-priority** parameter specifies the new VRRP priority that the router receives for this virtual router if the interface goes down.

The **activate** command activates the virtual router configuration on this interface. The interface does not provide backup service for the virtual IP address until you activate the VRRP configuration.

Syntax: ipv6 router vrrp

Syntax: ipv6 vrrp vrid vrid

Syntax: owner [**track-priority value**]

Syntax: backup [**priority value**] [**track-priority value**]

Syntax: track-port ethernet slot/portnum ve num

Syntax: ipv6-address ip-addr

Syntax: activate

VRRP-E example for IPv4

To implement the IPv4 VRRP-E configuration shown in Overview of VRRP-E on page 694, configure the VRRP Routers as shown in the following sections.

Configuring router1

To configure VRRP Router1 in Overview of VRRP-E on page 694, enter the following commands.

```
device(config)# router vrrp-extended
device(config)# interface ethernet 1/6
device(config-if-e10000-1/6)# ip address 10.53.5.2/24
device(config-if-e10000-1/6)# ip vrrp-extended vrid 1
device(config-if-e10000-1/6-vrid-1)# backup priority 110 track-priority 20
device(config-if-e10000-1/6-vrid-1)# track-port ethernet 2/4
device(config-if-e10000-1/6-vrid-1)# ip-address 10.53.5.254
device(config-if-e10000-1/6-vrid-1)# activate
VRRP router 1 for this interface is activating
device(config-if-e10000-1/6-vrid-1)# exit
device(config)# interface ethernet 1/6
device(config-if-e10000-1/6)# ip vrrp-extended vrid 2
device(config-if-e10000-1/6-vrid-1)# backup priority 100 track-priority 20
device(config-if-e10000-1/6-vrid-1)# track-port ethernet 2/4
device(config-if-e10000-1/6-vrid-1)# ip-address 10.53.5.253
device(config-if-e10000-1/6-vrid-1)# activate
VRRP router 2 for this interface is activating
```

NOTE
The address you enter with the **ip-address** command cannot be the same as a real IP address configured on the interface.

NOTE
If VRRP is not configured globally, then you will see the response "Invalid input..." when you try to create a VRRP instance.

Configuring router2

To configure Router2, enter the following commands.

```
device(config)# router vrrp-extended
device(config)# interface ethernet 5/1
device(config-if-e10000-5/1)# ip address 10.53.5.3/24
device(config-if-e10000-5/1)# ip vrrp-extended vrid 1
device(config-if-e10000-5/1-vrid-1)# backup priority 100 track-priority 20
device(config-if-e10000-5/1-vrid-1)# track-port ethernet 3/2
device(config-if-e10000-5/1-vrid-1)# ip-address 10.53.5.254
device(config-if-e10000-5/1-vrid-1)# activate
device(config-if-e10000-5/1-vrid-1)# exit
device(config)# interface ethernet 5/1
device(config-if-e10000-5/1)# ip vrrp-extended vrid 2
device(config-if-e10000-5/1-vrid-1)# backup priority 110 track-priority 20
device(config-if-e10000-5/1-vrid-1)# track-port ethernet 2/4
device(config-if-e10000-5/1-vrid-1)# ip-address 10.53.5.253
device(config-if-e10000-5/1-vrid-1)# activate
```

The **backup** command specifies that this router is a VRRP-E Backup for virtual router VRID1. The IP address entered with the **ip-address VRRP-E** command is the same IP address as the one entered when configuring Router1. In this case, the IP address cannot also exist on Router2, but the interface on which you are configuring the virtual router Backup must have an IP address in the same subnet. By entering the same IP address as the one associated with this virtual router on the Owner, you are configuring the Backup to back up the address, but you are not duplicating the address.

NOTE
When you configure a Backup router, the router interface on which you are configuring the virtual router must have a real IP address that is in the same subnet as the address associated with the virtual router by the Owner. However, the address cannot be the same.

The **priority** parameter establishes the router's VRRP-E priority in relation to the other VRRP-E routers in this virtual router. The **track-priority** parameter specifies the new VRRP-E priority that the router receives for this virtual router if the interface goes down. Refer to Track ports and track priority on page 692.

The **activate** command activates the virtual router configuration on this interface. The interface does not provide backup service for the virtual IP address until you activate the VRRP-E configuration. Alternatively, you can use the **enable** command. The **activate** and **enable** commands do the same thing.

Syntax: [no] router vrrp-extended

Syntax: [no] ip vrrp-extended vrid vrid

Syntax: [no] backup [**priority value**] [**track-priority value**]

Syntax: [no] track-port ethernet slot/portnum ve num

Syntax: [no] ip-address ip-addr

Syntax: [no] activate

VRRP-E example for IPv6

To implement the IPv6 VRRP-E configuration, configure the VRRP routers as shown in the following sections.

Configuring router1

To configure VRRP Router1, enter the following commands.

```
device(config)# ipv6 router vrrp-extended
device(config)# interface ethernet 1/6
device(config-if-e10000-1/6)# ipv6 address 1414:1414:1414::3/64
device(config-if-e10000-1/6)# ipv6 vrrp-extended vrid 1
device(config-if-e10000-1/6-vrid-1)# ipv6-address 1414:1414:1414::45
device(config-if-e10000-1/6-vrid-1)# activate
VRRP router 1 for this interface is activating
device(config-if-e10000-1/6-vrid-1)# exit
device(config)# interface ethernet 1/6
device(config-if-e10000-1/6)# ipv6 vrrp-extended vrid 2
device(config-if-e10000-1/6-vrid-1)# ipv6-address 1414:1414:1414::44
device(config-if-e10000-1/6-vrid-1)# activate
VRRP router 2 for this interface is activating
```

NOTE
The address you enter with the **ipv6-address** command cannot be the same as a real IP address configured on the interface.

NOTE
If VRRP is not configured globally, then you will see the response "Invalid input..." when you try to create a VRRP instance.

Configuring router2

To configure Router2, enter the following commands.

```
device(config)# ipv6 router vrrp-extended
device(config)# interface ethernet 5/1
device(config-if-e10000-5/1)# ipv6 address 1414:1414:1414::4/64
device(config-if-e10000-5/1)# ipv6 vrrp-extended vrid 1
device(config-if-e10000-5/1-vrid-1)# ipv6-address 1414:1414:1414::45
device(config-if-e10000-5/1-vrid-1)# activate
device(config-if-e10000-5/1-vrid-1)# exit
device(config)# interface ethernet 5/1
device(config-if-e10000-5/1)# ipv6 vrrp-extended vrid 2
device(config-if-e10000-5/1-vrid-1)# ipv6-address 1414:1414:1414::44
device(config-if-e10000-5/1-vrid-1)# activate
```

The **backup** command specifies that this router is a VRRP-E Backup for virtual router VRID1. The IP address entered with the **ipv6-address VRRP-E** command is the same IP address as the one entered when configuring Router1. In this case, the IP address cannot also exist on Router2, but the interface on which you are configuring the virtual router Backup must have an IP address in the same subnet. By entering the same IP address as the one associated with this virtual router on the Owner, you are configuring the Backup to back up the address, but you are not duplicating the address.

NOTE
When you configure a Backup router, the router interface on which you are configuring the virtual router must have a real IP address that is in the same subnet as the address associated with the virtual router by the Owner. However, the address cannot be the same.

The **priority** parameter establishes the router's VRRP-E priority in relation to the other VRRP-E routers in this virtual router.

The **activate** command activates the virtual router configuration on this interface. The interface does not provide backup service for the virtual IP address until you activate the VRRP-E configuration. Alternatively, you can use the **enable** command. The **activate** and **enable** commands do the same thing.

Syntax: [no] ipv6 router vrrp-extended

Syntax: [no] ipv6 vrrp-extended vrid vrid

Syntax: [no] backup [**priority value**] [**track-priority value**]

Syntax: [no] ipv6-address ip-addr

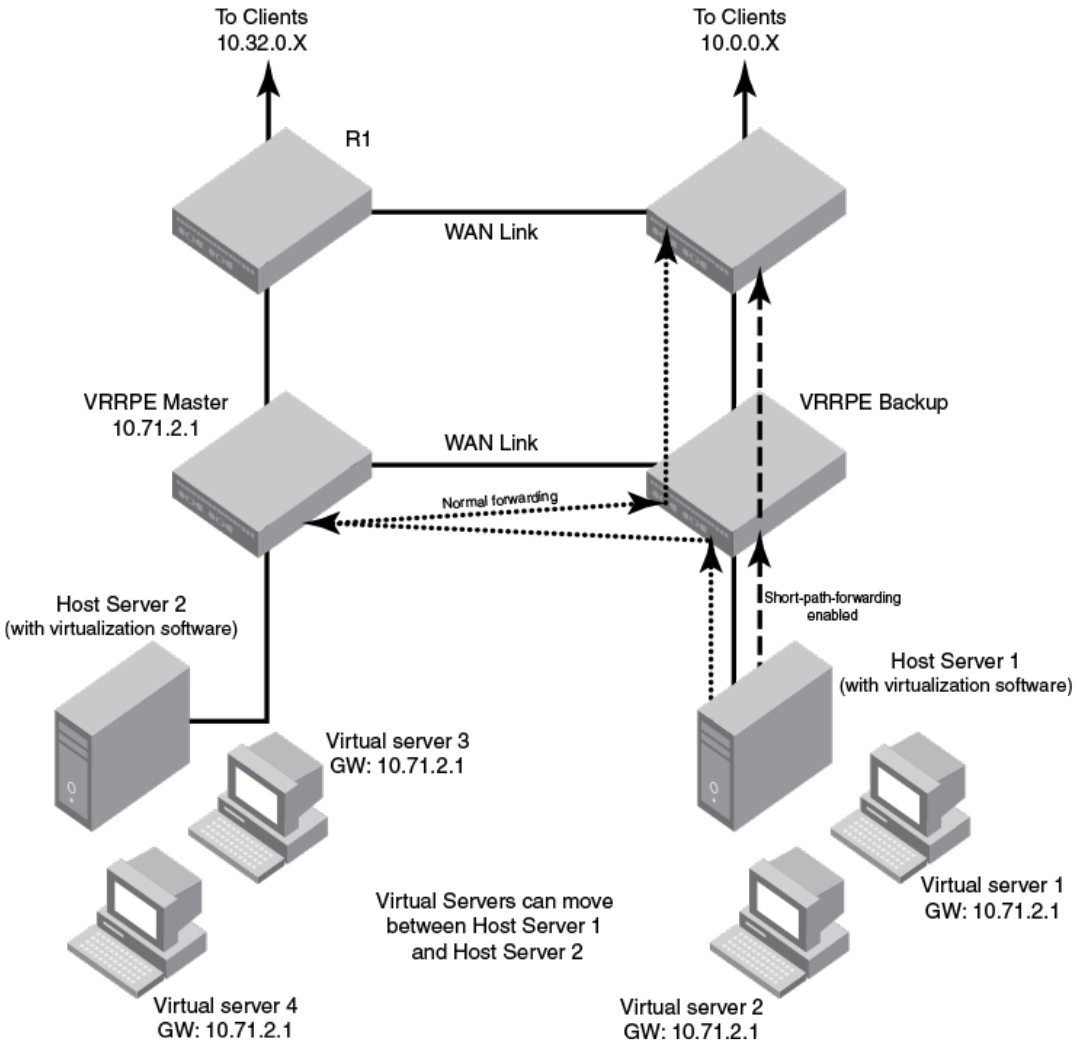
Syntax: [no] activate

VRRP-E Extension for Server Virtualization

VRRP-E is enhanced with the VRRP-E extension for Server Virtualization feature so that the Brocade device attempts to bypass the VRRP-E master router and directly forward packets to their destination through interfaces on the Backup router.

Figure 41 shows an example of VRRP-E Extension for Server Virtualization. As shown, the virtual servers are dynamically moved between Host Server 1 and Host Server 2. Each time the virtual server is activated, it can be on a different Host Server, and sometimes the traffic crosses the WAN two times before it reaches the client. For example, in the VRRP-E implementation (without VRRP-E Extension for Server Virtualization), traffic from virtual server 1 to the client at 10.0.0.X was switched to the VRRP-E master router, then routed back to VRRP-E Backup router, and then routed to the client (the normal forwarding path, dotted lines).

FIGURE 41 Short path forwarding



VRRP-E Extension for server virtualization configuration example

Under the VRRP-E VRID configuration level, there is an option to enable short-path-forwarding.

To enable **short-path-forwarding,** enter the following commands.

```
device(config)# router vrrp-extended
device(config)# interface ve 10
device(config-vif-10)# ip address 10.10.10.25/24
device(config-vif-10)# ip vrrp-extended vrid 10
device(config-vif-10-vrid-10)# backup priority 50
device(config-vif-10-vrid-10)# ip-address 10.10.10.254
device(config-vif-10-vrid-10)# short-path-forwarding
device(config-vif-10-vrid-10)# activate
```

Syntax: [no] short-path-forwarding

Packets from the local subnet of the virtual IP address

If VRRP-E Extension for Server Virtualization is enabled, any packets coming from the local subnet of the virtual IP address will be routed to the VRRP-E master router. This is for the routes whose next-hop gateway is the master router at the Backup router. These routes are routed to the WAN instead of switching them to the master router. The new behavior includes all the packets sent to the virtual IP address that were intended for the master router, such as Telnet, ping and traceroute packets. With VRRP-E Extension for Server Virtualization enabled these packets are now routed instead of switched. Traceroute output will show one extra hop for the source IP subnet that displays the Backup router interface IP address.

The following is an example of the traceroute command output with VRRP-E Extension for Server Virtualization enabled.

```
C:\ >traceroute 10.10.10.254
Tracing route to 10.10.10.254 over a maximum of 30 hops
1    <1 ms    <1 ms    <1 ms  10.10.10.25
 2    <1 ms    <1 ms    <1 ms  10.10.10.254
Trace complete.
```

The following is an example of the traceroute command output without VRRP-E Extension for Server Virtualization enabled:

```
C:\ >traceroute 10.10.10.254
Tracing route to 10.10.10.254 over a maximum of 30 hops
1     8 ms     8 ms     8 ms  10.10.10.254
Trace complete.
```

IPv4 VRF support

VRRP-E Extension for Server Virtualization supports IPv4 VRF forwarding.

Configuration considerations

Since the VRRP-E Extension for Server Virtualization enabled port will route all the traffic sent to the VRRP-E MAC, ACL and PBR features configured on the port will be applied to this traffic.

VRRP-E Extension for Server Virtualization can be dynamically enabled or disabled before or after VRRP-E is activated. No system reload is required.

Although it is not required, it is recommended that interfaces on different routers with the same VRID have the same short-path-forwarding configuration. This will ensure that the short-path forwarding behavior is still retained after the failover. Different VRIDs can be configured differently.

Review the SYSLOG to prevent issues such as the following.

When the VRRP master is being reconfigured from VRRP-e to VRRP, the ve interface IPv6 address is removed and then re-added, at the time of re-adding the address. This happens to be the virtual IPv6 address of the VRRP-e, and it will trigger a fault due to the failed duplicate address detection since the VRRP Backup still running VRRP-e. The error is shown on SYSLOG although the "ipv6 address ..." command is not completely rejected. This results in a VRRP operation error. Watch for a "duplucate address error" in SYSLOG and make the necessary configuration adjustment as required.

GPRS Tunneling Protocol

- GPRS Tunneling Protocol Overview.....737
- GPRS Tunneling Protocol Filtering and Load-balancing.....737

GPRS Tunneling Protocol Overview

Use GPRS Tunneling Protocol (GTP) to selectively filter and load-balance on GTP fields for GTP packets on GTP enabled ports.

The GPRS core network provides mobility management, session management and transport for Internet Protocol packet services in GSM and WCDMA networks. GPRS is similar to a GSM network except for two new nodes (SGSN and GGSN) and a piece of hardware (PCU).

- The Packet Control Unit (PCU) differentiates data destined for the standard GSM network or Circuit Switched Data and data destined for the GPRS network or Packet Switched Data.
- The Serving GPRS Support Node (SGSN) takes care of some important tasks, including routing, handover and IP address assignment.
- The Gateway GPRS Support Node (GGSN) is the last node in the GPRS network before a connection between an ISP or corporate network's router occurs.

The connection between these two GPRS Support Nodes is made with a protocol called GPRS Tunnelling Protocol (GTP).

NOTE
The GTP is only supported on Gen2+ cards.

GPRS Tunneling Protocol Filtering and Load-balancing

The GPRS Tunneling Protocol (GTP) Filtering and Load-balancing feature allows you to selectively filter and load-balance on GTP fields for GTP packets on GTP enabled ports.

This feature supports GTP packets and incorporates the Tunnel Endpoint Identifier (TEID) in the Trunk/ECMP hashing logic.

About GTP load-balance configuration

The GTP profile configuration allows the user to specify the desired interfaces that will need to process GTP packets, and what options are needed to allow the NetIron device to look deeper into the packet.

NOTE
This command can only be applied to Gen2+ cards, except 24x10G cards.

Ports may be specified using the ports command under the GTP profile. Command syntax is as follows.

Syntax: ports all-ethernet |**ethernet***slot/port* **ethernet**|**to***slot/port*

Additionally the GTP profile allows the user to enable extra information into the load-balancing hash algorithm via the following commands:

Brocade(config-gtp-gtp2)#**load-balance port-gtpc-teid-hash-ena**

The **load-balance port-gtpc-teid-hash-ena** command enables the GTPc packets to add the TEID field into the hash mechanism for the ports of the profile.

Brocade(config-gtp-gtp2)#**load-balance port-gtpu-innerl3-hash-ena**

The command **load-balance port-gtpu-innerl3-hash-ena** enables the tunneled l3 and l4 information from GTPu packets to be added into the hash mechanism for the ports of the profile.

Brocade(config-gtp-gtp2)#**load-balance port-gtpu-teid-hash-ena**

The command **load-balance port-gtpu-teid-hash-ena** enables the TEID field from GTPu packets to be added into the hash mechanism for the ports of the profile.

Brocade(config-gtp-gtp2)#**ingress-inner-filter**

The **ingress-inner-filter** command changes the behavior of ACL's and policy based routing (PBR) on the GTP profile ports. This enables the ACL or PBR to match on the inner l3 and l4 header information of GTPu packets.

NOTE
Up to 16 profiles may be configured on the system. Profile names my be up to 64 characters long. Valid profile ID's are 1 to 16.

Example GTP Configuration

Following is an example of creating a GTP configuration. The following examples show GTP configuration and related show output.

```
Brocade(config)#gtp gtp2 12
Brocade(config-gtp-gtp2)#ports ethernet 10/1 to 10/4
Brocade(config-gtp-gtp2)#load-balance port-gtpc-teid-hash-ena
Brocade(config-gtp-gtp2)#load-balance port-gtpu-teid-hash-ena
Brocade(config-gtp-gtp2)#load-balance port-gtpu-innerl3-hash-ena
Brocade(config-gtp-gtp2)#ingress-inner-filter
```

Example of GTP Configuration show output

Following example is how the GTP profile looks like after the above configuration.

```
Brocade#show gtp
Total no. of GTP profiles :: 1

==========GTP gtp2 (12)============
GTP configuration
Port count :: 4
  Ports :: eth 10/1 to 10/4

Loadbalance hashing options
        GTPC TEID Hash enabled          :: Yes
        GTPU Inner L3 Hash enabled      :: Yes
        GTPU TEID Hash enabled          :: Yes
Ingress Inner L4 filter enabled         :: Yes
```

Enable masking of the TEID (Tunnel endpoint identifier) field for GTP packets

To mask information that has been added to the LAG hash algorithm for GTP profile ports, the following commands are available. Each masks the indicated field inside the tunneled L3 and L4 headers, or the GTP header.

By default, the TEID field will not be masked and will be used in hashing calculations only for GTP packets on GTP enabled ports. For non-GTP enabled interfaces, it will not be used, even if GTP packets are received on them. This does not affect the non-GTP packets.

The masking options are available to mask certain fields while calculating the hash for load-balancing. For example, the following command will mask the src-ip address for GTP Ipv4 packets.

```
Brocade(config)#load-balance mask gtp ipv4 src-ip
```

Basic command format is **load-balance mask gtp ipv4**|**ipv6**|**teid** *field option slot*|**all**

The following are the **load-balance mask gtp** options available.

- **ipv4** Mask IPv4 header fields
- **ipv6** Mask IPv6 header fields
- **teid** Mask TEID information in Trunk/ECMP hash

The following are the **load-balance mask gtp ipv4** options available.

- **dst-ip** Mask Destination IP address
- **dst-l4-port** Mask Destination L4 port
- **protocol** Mask IP protocol id
- **src-ip** Mask Source IP address
- **src-l4-port** Mask Source L4 port

The following are the **load-balance mask gtp ipv6** options available.

- **dst-ip** Mask Destination IP address
- **dst-l4-port** Mask Destination L4 port
- **next-hdrl**Mask next header id
- **src-ip** Mask Source IP address
- **src-l4-port** Mask Source L4 port

The following are the **load-balance mask gtp ipv4 dst-ip** options available.

- **DECIMAL** Slot Number
- **all** All slots

MPLS unknown label handling

This task shows how to use MPLS unknown label handling.

If the MPLS unknown label handling is turned on, all unknown MPLS packets entering the specified ingress interface will be stripped of their MPLS labels, and sent to specified egress interface.

NOTE
This command can only be applied to Gen2+ cards, except 24x10G cards.

Enter the **mpls-unknown-label-forward** command to direct all unknown MPLS packets to the specified interface.

The command format is **mpls-unknown-lable-forward ingress** ingress port **egress** egress port

```
Brocade(config)#mpls-unknown-label-forward ingress 10/8 egress 8/1
Reload required. Please write memory and then reload the system.
Failure to reload could cause system instability on failover.
Newly configured mpls catch-all value will not take effect during hitless-reload.
Brocade(config)#
```

Enabling internal loopback

The internal loopback feature allows the configured interface to redirect packets that would normally egress the interface and be physically loop backed via a short fiber, or specialized hardware device, to the ingress of the interface without a physical device.

This behavior is supported on the following modules:

- BR-MLX-40Gx4-M
- BR-MLX-10Gx20
- BR-MLX-100Gx2-CFP2

NOTE
Usage of an optic in interfaces that are configured as internal loopback is not supported.

The following example shows the command to enable this:

```
Brocade(config)#int e 1/1
Brocade(config-if-e10000-1/1)#loopback system
```

GTP Profile configuration commands

Use the following commands to maintain GTP hashing and filtering configurations in unique GTP profiles. A maximum of 16 profiles are supported.

Naming a GTP profile

Maintain GTP hashing and filtering configurations in unique GTP profiles. A maximum of sixteen profiles are supported.

Syntax **gtp_profile** { **ASCII string** | **Decimal** }

Parameters

ASCII string

Name of the GTP profile

Decimal

ID of the GTP profile.

Modes This command operates under the configuration mode (config).

Usage Guidelines Use this command to maintain and create GTP hashing and filtering configurations in unique GTP profiles. A maximum of sixteen profiles are supported.

Examples

```
Brocade (config)#gtp_profile gtp1
```

History

Release version	Command history
NetIron 5.7.00	This command was introduced.

Adding port list to enable GTP with this profile

The following section describes how to add a port list to enable GTP in a profile.

Syntax **ports** { **slot/port** | **all** }

Parameters

slot/port

The port to add to the enable GTP list.

All

Enable all ports.

Modes This command operates under the GTP configuration mode (config-gtp).

Usage Guidelines If the port is LAG, you will need to specify the primary port of the LAG. For LAG, this profile will be applied to all ports of the LAG. When new ports are added to LAG, they will inherit the same profile. If a secondary port specified, it will be rejected.

Examples

```
Brocade(config-gtp-gtp1)# ports 1/1
```

History

Release version	Command history
5.7.00	This command was introduced.

Show GTP

Displays GTP profile information.

Syntax **show gtp** { **Id** | **Name** | **Interface**}

Parameters

ID

Shows GTP by ID.

Name

Shows GTP by name.

Interface

Show GTP interfaces.

Examples Example of the **Show GTP ID** command output.

```
Brocade#show gtp id 12
Total no. of GTP profiles :: 1
===========GTP gtp2 (12)============
GTP configuration
Port count :: 4
  Ports :: eth 10/1 to 10/4

Loadbalance hashing options
        GTPC TEID Hash enabled          :: Yes
        GTPU Inner L3 Hash enabled      :: Yes
        GTPU TEID Hash enabled          :: Yes
Ingress Inner L4 filter enabled         :: Yes
```

Example of the **Show GTP name** command output.

```
Brocade#show gtp name gtp2
Total no. of GTP profiles :: 1
===========GTP gtp2 (12)============
GTP configuration
Port count :: 4
  Ports :: eth 10/1 to 10/4

Loadbalance hashing options
        GTPC TEID Hash enabled          :: Yes
        GTPU Inner L3 Hash enabled      :: Yes
        GTPU TEID Hash enabled          :: Yes
Ingress Inner L4 filter enabled         :: Yes
```

Example of the **Show GTP interface** command output.

```
Brocade#show gtp interfaces
        Slot/Interface          GTP Profile Name
            10/1               gtp2
            10/2               gtp2
            10/3               gtp2
            10/4               gtp2
```

History

Release version	Command history
NetIron 5.7.00	This command was introduced.

Show loadbalance mask-options command

Use this command to display the load balance masking information.

Syntax **Show loadbalance mask-options** { **gtp**}

Examples The following is an example of the **show loadbalance mask-options gtp** command output.

```
Brocade# show load-balance mask-options gtp

Mask GTP options -

  Mask GTP TEID is enabled on -
  No Slots

  Mask GTP IPv4 Source IP address is enabled on -
  Slot 1

  Mask GTP IPv4 Destination IP address is enabled on -
  All Slots

  Mask GTP IPv4 Destination L4 port  is enabled on -
  No Slots

  Mask GTP IPv4 Source L4 port is enabled on -
  No Slots

  Mask GTP IPv6 Source IP address is enabled on -
  No Slots

  Mask GTP IPv6 Destination IP address is enabled on -
  No Slots

  Mask GTP IPv6 Source L4 port is enabled on -
  No Slots

  Mask GTP IPv6 Destination L4 port is enabled on -
  No Slots

  Mask GTP IPv6 Next Header information is enabled on -
  No Slots
```

History

Release version	Command history
NetIron 5.7.00	This command was introduced.

Enabling and Disabling MAC movement syslog on MLX device

The following command enable or disable the display of MAC movement syslog messages on Brocade NetIron MLX Series device.

Syntax: [no] mac-move-det-syslog

```
Brocade(config)#mac-move-det-syslog
```

The **no** form of the command disables the display of MAC movement syslog messages.

The following example shows the MAC movement syslog message output when **mac-move-det-syslog** command is used.

```
Brocade(config)#mac-move-det-syslog
Brocade(config)#show arp
Total number of ARP entries: 2
(In all VRFs)

Entries in default routing instance:
IP          Address             MAC Address         Type            Age      Port
(Vpls-Id, Vlan)/ Vpls-Id:Peer

1      19.19.19.1           0010.9400.0606      Dynamic           1               1/24
2      172.26.67.1          0024.381c.b900      Dynamic           1
mgmt1
Brocade(config)#
Brocade(config)#
Brocade#
SYSLOG: <12>Sep 25 02:43:07 IP/ARP: IP address 19.19.19.1 MAC movement detected,
changed from MAC 0010.9400.0606 / port 1/24 to MAC 0010.9400.0001 / port 1/24

Brocade#
Brocade#
Brocade#
Brocade(config)#show arp
Total number of ARP entries: 2
(In all VRFs)

Entries in default routing instance:
      IP Address             MAC Address         Type        Age              Port/
Port (Vpls-Id, Vlan)/
1     19.19.19.1           0010.9400.0001      Dynamic     1               1/24
2     172.26.67.1         0024.381c.b900      Dynamic     2            mgmt1
Brocade(config)#
Brocade(config)#
SYSLOG: <12>Sep 25 02:43:40 IP/ARP: IP address 19.19.19.1 MAC movement detected,
changed from MAC 0010.9400.0001 / port 1/24 to MAC 0010.9400.0606 / port 1/24
```

Use **[no] mac-move-det-syslog** command to disable the MAC movement syslog.

The following example shows the MAC movement syslog message output when **[no] mac-move-det-syslog** command is used.

```
Brocade(config)#no mac-move-det-syslog
Brocade(config)#
Brocade(config)#
Brocade(config)#show arp
Total number of ARP entries: 2
(In all VRFs)

Entries in default routing instance:
      IP Address             MAC Address         Type        Age              Port
(Vpls-Id, Vlan)/ Vpls-Id:Peer

1     19.19.19.1           0010.9400.0001      Dynamic     1               1/24
2     172.26.67.1          0024.381c.b900      Dynamic     2            mgmt1
```

```
Brocade#
Brocade#
```

Routing commands

- default-link-metric....748
- ipv6 nd ra-dns-server....749
- ipv6 nd ra-domain-name....750
- isis reverse-metric....752
- mac-move-det-syslog....755
- neighbor next-hop-self (BGP)....757
- reverse-metric....758
- show metro mp-vlp-queue....761
- system-max ip-vrf-route....763
- system-max ipv6-vrf-route....764

default-link-metric

Configures the metric value globally on all active IPv4 IS-IS interfaces.

Syntax **default-link-metric** *value* [**level-1** | **level-2**]

no default-link-metric *value* [**level-1** | **level-2**]

Command Default The **default-link-metric** command is disabled by default.

Parameters

default-link-metric

Specifies the global default-link-metric parameter for an IPv4 IS-IS unicast address family configuration.

value

Specifies the default-link-metric value in metric style and configurable range. The metric style consists of narrow or wide style. The narrow metric range is from 1 - 63. The wide metric range is from 1 - 16777215. If you change the metric style configuration, the default-link-metric value will also change. The new default-link-metric value is equal to the minimum of the configured value, and the maximum value supported by the new metric style. For example, if the metric style changes from a wide metric to a narrow metric, and the default-link-metric value is greater than 63, the default-link-metric value changes to 63 because it is the maximum value supported in the narrow metric style. When the metric style changes from a narrow metric to a wide metric, there is no change to the default-link-metric value.

level-1 | level-2

Specifies the IS-IS routing parameter as level-1 or level-2. You can choose to configure the default-link-metric parameter as either level-1 or level-2. If the IS-IS routing parameter is not configured, the default-link-metric value is applied to both level-1 and level-2.

Modes IPv4 IS-IS unicast address family configuration level.

Usage Guidelines Use the **default-link metric** *value* command to change the metric value globally on all active IPv4 IS-IS interfaces. The **default-link metric** *value* command is useful when you have a common IS-IS metric value on all IS-IS interfaces, other than the default metric value of 10. The command enables the metric value for IPv4 routes per address family configuration. Use the **no** form of the command to reset the metric value to the default value 10. The **default-link metric** *value* command is not applicable to MPLS IS-IS shortcuts and tunnel interfaces.

You can change the metric value for a specific interface using the **isis metric** command or the **isis ipv6** command. The **isis metric** command configuration takes precedence over the **default-link metric** *value* command configuration.

During switchover or hitless upgrade, the IS-IS default link metric configuration is not affected. Backward compatibility is not supported.

NOTE
The **default-link metric** *value* command is supported on the Brocade NetIron XMR Series, the Brocade MLX Series, and the Brocade NetIron CER Series and Brocade NetIron CES Series platforms.

Examples The following example configures the IS-IS default link metric value to 30 for an IPv4 address family. The default-link-metric value of 30 is applied to both level-1 and level-2.

```
device(config)#router isis
device(config-isis-router)#address-family-ipv4 unicast
device(config-isis-router-ipv4u)#default-link-metric 30
device(config-isis-router-ipv4u)#
```

The following example configures the IS-IS default link metric value to 30 for level-1, and the IS-IS default link metric value of 40 to level-2.

```
device(config)#router isis
device(config-isis-router)#address-family-ipv4 unicast
device(config-isis-router-ipv4u)#default-link-metric 30 level-1
device(config-isis-router-ipv4u)#default-link-metric 40 level-2
```

Use the **show isis** command to display the configuration for the IS-IS default link metric value.

```
device(config)#show isis
…..
Default redistribution metric: 0
Default link metric for level-1: 33 (conf)/ 33 (adv)
Default link metric for level-2: 5 (conf)/ 5 (adv)
Protocol Routes redistributed into IS-IS:
…..
```

History

Release version	Command history
Multi-Service Ironware R05.7.00	This command was introduced.

ipv6 nd ra-dns-server

Advertises the recursive Domain Name System (DNS) server address and the lifetime multiplier information to IPv6 hosts in the Router Advertisement (RA) message. The **no** form of this command disables the advertisement of the specified server address in the RA message.

Syntax **ipv6 nd ra-dns-server** *ipv6-address* [**lifetime-multiplier** *decimal*]

no ipv6 nd ra-dns-server *ipv6-address* [**lifetime-multiplier** *decimal*]

Command Default By default, the recursive DNS server address and the lifetime multiplier information is not configured.

Parameters ***ipv6-address***

Specifies the global IPv6 address of the DNS server.

lifetime-multiplier ***decimal*** **Specifies the percentage value of maximum router advertisement interval. The maximum router advertisement interval is the maximum time that can be allowed between sending unsolicited RA messages for DNS name resolution. The lifetime-multiplier** ***decimal***

value is calculated as percentage of the RA lifetime. The maximum router advertisement interval percentage range is 100 through 200% and the default value is 200%.

Modes Global configuration mode

Interface configuration mode

Usage Guidelines You can configure a maximum of four recursive DNS server addresses and corresponding lifetime multiplier values at a given instance.

NOTE
The **ipv6 nd ra-dns-server** command at the interface configuration level takes precedence over global configuration. In other words, if at least one DNS server address is configured on an interface, it will override other DNS server address configurations at the global configuration.

Examples The following examples configure the recursive DNS address for a lifetime-multiplier value of 200.

```
device(config)# ipv6 nd ra-dns-server 2001:DC8:200::3 lifetime 200
device(config-if-e10000-1/10)# ipv6 nd ra-dns-server 2001:DC8:200::3 lifetime 200
```

ipv6 nd ra-domain-name

Configures the domain name of the Domain Name System (DNS) suffix and the lifetime multiplier information to IPv6 hosts in the Router Advertisement (RA) message. The **no** form of this command disables the advertisement of the specified domain name of DNS suffix in the RA message.

Syntax **ipv6 nd ra-domain-name** *string* [**lifetime-multiplier** *decimal*]

no ipv6 nd ra-domain-name *string* [**lifetime-multiplier** *decimal*]

Parameters ***string***
Specifies the domain name of the DNS suffix.

lifetime-multiplier ***decimal***
Specifies the percentage value of maximum router advertisement interval. The maximum router advertisement interval is the maximum time that can be allowed between sending unsolicited RA messages for DNS name resolution. The **lifetime-multiplier** *decimal* **value is calculated as percentage of** the RA lifetime. **The maximum router advertisement interval percentage range is 100 through 200% and the default value is 200%.**

Modes Global configuration mode

Interface configuration mode

Usage Guidelines You can configure a maximum of four different domain names of DNS suffix and corresponding lifetime multiplier values at a given instance.

The domain name of a DNS suffix at the global configuration level is used on all IPv6 routed interfaces that do not have a domain name of DNS suffix configured on them.

NOTE
The **ipv6 nd ra-domain-name** command at the interface configuration takes precedence over global configuration. In other words, if at least one DNS server address is configured on an interface, it will override other DNS server address configurations at the global configuration.

Examples The following examples configure the domain names of a DNS suffix for a lifetime-multiplier value of 200.

```
device (config)# ipv6 nd ra-domain-name brocade.com lifetime 200
device (config-if-e10000-1/10)# ipv6 nd ra-domain-name brocade.com lifetime 200
```

History

Release	Command History
Multi-Service IronWare R05.5.00	This command was introduced.

isis reverse-metric

Configures the reverse metric value on a single IS-IS interface.

Syntax **isis reverse-metric** [*value*] [**whole-lan**] [**te-def-metric**]

no isis reverse-metric [*value*] [**whole-lan**] [**te-def-metric**]

Command Default The **isis reverse-metric** command is disabled by default.

Parameters

isis reverse-metric

Specifies the reverse metric parameter at the interface level.

value

Specifies the reverse metric value in metric style. The metric style consists of narrow or wide style. The narrow metric range is from 1 - 63. The wide metric range is from 1 - 16777215. The default value is 16777214 irrespective of the metric style configured. If the reverse-metric value is configured, the local LSP is updated with the sum of the default metric and the reverse metric value. When the IS-IS neighbor router receives the reverse metric value through the IS hello, the neighbor router updates the cost to reach the original IS-IS router with the sum of default metric and the reverse metric value. This helps in shifting traffic to the other alternate paths.

whole-lan

Specifies changing the reverse metric parameter for the entire LAN. The **whole-lan** option indicates the whole LAN bit in the flag. If the **whole-lan** option is enabled, the configured reverse metric value affects the entire LAN. If the **whole-lan** option is not enabled, the reverse metric value affects only the neighbor router. This option takes effect only on the multi-access LAN. IS-IS point-to-point interfaces are not affected when the **whole-lan** option is enabled.

te-def-metric

Specifies setting the TE default metric sub-TLV. If the **te-def-metric** option is enabled, the router sends a TE default metric sub-TLV within the reverse-metric TLV.

Modes IS-IS interface level.

Usage Guidelines Use the **isis reverse-metric** command when you are performing network maintenance operations, such as software upgrades, at the link level. When maintenance operations are performed, the link undergoing maintenance should not be used by the neighbor routers to forward transit traffic. In order to shift traffic away from the link undergoing maintenance, configure the **isis reverse-metric** command on the maintenance link. The router undergoing maintenance first advertises a reverse metric TLV in a IS-IS hello PDU to its neighbor router on a point-to-point or multi-access link. When the neighbor router receives a high reverse metric value, the router selects alternate paths to forward traffic while maintenance is going on. The neighbor router adds the reverse metric TLV to its own TE default metric sub-TLV and recalculates its SPF tree and route topology. The neighbor router floods the new LSP containing the extended IS reachability TLV throughout the domain. Traffic gradually shifts onto alternate paths away from the link between the maintenance router and the neighbor router as nodes in the IS-IS domain receive the new LSP. Once the maintenance is complete, you can remove the **isis reverse-metric** command configuration on the link, and the reverse metric TLV in the IS-IS hello PDU is no longer advertised to the neighbor router. The IS-IS neighbor router reverts back to its original IS-IS metric, and the traffic switches to the original IS-IS link to reach its destination.

In a multi-access link, the IS-IS DIS router adds the reverse metric TLV value to each node's default metric value in the pseudonode LSP when the whole-lan flag is set. All non-DIS nodes ignore the

reverse metric TLV. If multiple neighbor routers advertise the reverse metric TLV with the whole LAN flag set, the neighbor router with the highest MAC address takes precedence, and the value advertised by that neighbor is updated in the pseudonode LSP for all neighbors. If some neighbor routers do not set the whole LAN flag, then the reverse metric TLV value advertised by the neighbor router is updated in the pseudonode LSP for that neighbor only.

The S flag is set when the sender of the reverse metric TLV signals to the neighbor router to use the TE sub-tlv for the default metric (sub-tlv type 18) in the reverse metric TLV. When the receiving router finds the S flag set in the reverse metric TLV, the router searches for the TE sub-tlv. The router adds the default metric value in the TE sub-tlv to the configured TE default metric value and recalculates the CSPF.

The **no** form of the command, specified with the configured value, resets the metric value to the default value of 16777214. The **no isis reverse-metric** command removes the entire reverse metric configuration.

NOTE
The **isis reverse-metric** *value* command is supported on the Brocade NetIron XMR Series, the Brocade MLX Series, and the Brocade NetIron CER Series and Brocade NetIron CES Series platforms.

Examples The following example configures the reverse metric value to 40 on a single IS-IS interface level. The **whole-lan** option is enabled to include the entire LAN.

```
device(config)#interface ethernet 2/2
device(config-if-e1000-2/2)# isis reverse-metric ?
DECIMAL           Narrow metric range 1-63, Wide metric range 1-16777214,
                  Default is 16777214
te-def-metric   Update TE default metric sub-tlv
whole-lan       Change metric for whole LAN
device(config-if-e1000-2/2)# isis reverse-metric 40 ?
te-def-metric   Update TE default metric sub-tlv
whole-lan       Change metric for whole LAN
<cr>
device(config-if-e1000-2/2)#isis reverse-metric 40 whole-lan
device(config-if-e1000-2/2)#
```

Use the **show isis** command to display the configuration of the reverse metric value at the global level. The reverse metric value and flags are highlighted in the output.

```
device(config)#show isis
IS-IS Routing Protocol Operation State: Enabled
 IS-Type: Level-1-2
 System ID: aaaa.bbbb.cccc
 Manual area address(es):
   49.2211
 Level-1-2 Database State: On
 Administrative Distance: 115
 Maximum Paths: 4

 ISIS Global Reverse Metric 40
 ISIS Global Reverse Metric Flags: W S
```

Use the **show isis interface** command to display the configuration of the reverse metric value at the interface level. The reverse metric value and flags are highlighted in the output.

```
device(config)#show isis interface
Total number of IS-IS Interfaces: 1

Interface: eth 1/1
   Circuit State: DOWN Circuit Mode: LEVEL-1-2
   Circuit Type: BCAST Passive State: FALSE
   Circuit Number: 1, MTU: 1500
   Level-1 Auth-mode: None
   Level-2 Auth-mode: None
   Level-1 Metric: 10, Level-1 Priority: 64
   Level-1 Hello Interval: 10 Level-1 Hello Multiplier: 3
   Level-1 Designated IS: MLX-2-01 Level-1 DIS Changes: 1
   Level-2 Metric: 10, Level-2 Priority: 64
   Level-2 Hello Interval: 10 Level-2 Hello Multiplier: 3
   Level-2 Designated IS: MLX-2-01 Level-2 DIS Changes: 1

   IP Enabled: TRUE
   IPv6 Enabled: FALSE
   MPLS TE Enabled: FALSE
   ISIS Reverse Metric 40
   ISIS Reverse Metric Flags: W S
   LDP-SYNC: Disabled, State: -
```

History

Release version	Command history
Multi-Service Ironware R05.7.00	This command was introduced.

mac-move-det-syslog

The following command enable or disable the display of MAC movement syslog messages on Brocade NetIron MLX Series device.

Syntax **[no] mac-move-det-syslog**

Command Default By default this command is enabled.

Modes Global configuration mode.

Usage Guidelines The **no** form of the command disables the display of MAC movement syslog messages.

Examples The following example shows the MAC movement syslog message output when **mac-move-det-syslog** command is used.

```
Brocade(config)#mac-move-det-syslog
Brocade(config)#show arp
Total number of ARP entries: 2
(In all VRFs)

Entries in default routing instance:
IP          Address             MAC Address         Type            Age      Port
(Vpls-Id, Vlan)/ Vpls-Id:Peer

1      19.19.19.1           0010.9400.0606      Dynamic           1               1/24
2      172.26.67.1          0024.381c.b900      Dynamic           1
mgmt1
Brocade(config)#
Brocade(config)#
Brocade#
SYSLOG: <12>Sep 25 02:43:07 IP/ARP: IP address 19.19.19.1 MAC movement detected,
changed from MAC 0010.9400.0606 / port 1/24 to MAC 0010.9400.0001 / port 1/24

Brocade#
Brocade#
Brocade#
Brocade(config)#show arp
Total number of ARP entries: 2
(In all VRFs)

Entries in default routing instance:
      IP Address              MAC Address         Type        Age              Port/
Port (Vpls-Id, Vlan)/
1     19.19.19.1           0010.9400.0001      Dynamic     1               1/24
2     172.26.67.1         0024.381c.b900      Dynamic     2            mgmt1
Brocade(config)#
Brocade(config)#
SYSLOG: <12>Sep 25 02:43:40 IP/ARP: IP address 19.19.19.1 MAC movement detected,
changed from MAC 0010.9400.0001 / port 1/24 to MAC 0010.9400.0606 / port 1/24
```

Use **[no] mac-move-det-syslog** command to disable the MAC movement syslog.

The following example shows the MAC movement syslog message output when **[no] mac-move-det-syslog** command is used.

```
Brocade(config)#no mac-move-det-syslog
Brocade(config)#
Brocade(config)#
Brocade(config)#show arp
Total number of ARP entries: 2
(In all VRFs)

Entries in default routing instance:
      IP Address              MAC Address         Type        Age              Port
(Vpls-Id, Vlan)/ Vpls-Id:Peer

1     19.19.19.1           0010.9400.0001      Dynamic     1               1/24
```

```
2     172.26.67.1          0024.381c.b900      Dynamic     2            mgmt1
Brocade#
Brocade#
```

History

Release version	Command history
Multi-Service IronWare Release 5.7.00	This command was introduced.

neighbor next-hop-self (BGP)

Causes the device to list itself as the next hop in updates that are sent to the specified neighbor.

Syntax **neighbor** *ip-address* | *ipv6-address* | *peer-group-name* **next-hop-self** [**always**]

no neighbor *ip-address* | *ipv6-address* | *peer-group-name* **next-hop-self**

Command Default None.

Parameters

ip-address
IPv4 address of the neighbor.

ipv6-address
IPv6 address of the neighbor.

peer-group-name
Peer group name configured by the **neighbor** *peer-group-name*.

always
Enables this feature for route reflector (RR) routes.

Modes BGP configuration mode.

Usage Guidelines Use the no form of this command to remove this configuration at BGP level.

Use this command to cause the device to list itself as the next hop in updates that are sent to the specified neighbor.

Examples This example configures the device to list itself as the next hop in updates sent to a neighbor with the IP address 10.157.22.26.

```
device# config
device(config)# router bgp
device(config-bgp-router)# neighbor 10.157.22.26 next-hop-self
```

This example configures the device to list itself as the next hop in updates sent to a neighbor that is a route-reflector client of the device.

```
device# config
device(config)# router bgp
device(config-bgp-router)# neighbor 10.157.22.26 next-hop-self always
```

History

Release version	Command history
Not available	This command was introduced.
Not available	This command was enhanced to support IPv6 addresses.

reverse-metric

Configures the reverse metric value at the IS-IS router level.

Syntax **reverse-metric** [*value*] [**whole-lan**] [**te-def-metric**]

no reverse-metric [*value*] [**whole-lan**] [**te-def-metric**]

reverse-metric tlv-type[*value*]

no reverse-metric tlv-type[*value*]

Command Default The **reverse-metric** command is disabled by default.

Parameters

reverse-metric

Specifies the reverse metric parameter at the IS-IS router level.

value

Specifies the reverse metric value in metric style. The metric style consists of narrow or wide style. The narrow metric range is from 1 - 63. The wide metric range is from 1 - 16777215. The default value is 16777214 irrespective of the metric style configured. If the reverse-metric value is configured, the local LSP is updated with the sum of the default metric and the reverse metric value. When the IS-IS neighbor router receives the reverse metric value through the IS hello, the neighbor router updates the cost to reach the original IS-IS router with the sum of default metric and the reverse metric value. This helps in shifting traffic to the other alternate paths.

whole-lan

Specifies changing the reverse metric parameter for the entire LAN. The **whole-lan** option indicates the whole LAN bit in the flag. If the **whole-lan** option is enabled, the configured reverse metric value affects the entire LAN. If the **whole-lan** option is not enabled, the reverse metric value affects only the neighbor router. This option takes effect only on the multi-access LAN. IS-IS point-to-point interfaces are not affected when the **whole-lan** option is enabled.

te-def-metric

Specifies setting the TE default metric sub-TLV. If the **te-def-metric** option is enabled, the router sends a TE default metric sub-TLV within the reverse-metric TLV.

tlv-type *value*

Specifies the TLV type for the reverse metric parameter. The TLV type can only be configured at the IS-IS router level. The **tlv-type** *value* parameter must be configured in the range of unassigned IS-IS TLV values. The **tlv-type** *value* parameter should not be configured with existing IS-IS TLV types. The default value is 254.

Modes IS-IS router level.

Usage Guidelines Use the **reverse-metric** command when you are performing network maintenance operations, such as software upgrades, on an IS-IS router node. When maintenance operations are performed, the router undergoing maintenance should not be used by the neighbor routers to forward transit traffic. In order to shift traffic away from the router undergoing maintenance, configure the **reverse-metric** command on the maintenance router. The router undergoing maintenance first advertises a reverse metric TLV in a IS-IS hello PDU to its neighbor router on a point-to-point or multi-access link. When the neighbor router receives a high reverse metric value, the router selects alternate paths to forward traffic while

maintenance is going on. The neighbor router adds the reverse metric TLV to its own TE default metric sub-TLV and recalculates its SPF tree and route topology. The neighbor router floods the new LSP containing the extended IS reachability TLV throughout the domain. Traffic gradually shifts onto alternate paths away from the link between the maintenance router and the neighbor router as nodes in the IS-IS domain receive the new LSP. Once the maintenance is complete, you can remove the **reverse-metric** command configuration from the router, and the reverse metric TLV in the IS-IS hello PDU is no longer advertised to the neighbor router. The IS-IS neighbor router reverts back to its original IS-IS metric, and the traffic switches to the original IS-IS router to reach its destination.

In a multi-access link, the IS-IS DIS router adds the reverse metric TLV value to each node's default metric value in the pseudonode LSP when the whole-lan flag is set. All non-DIS nodes ignore the reverse metric TLV. If multiple neighbor routers advertise the reverse metric TLV with the whole LAN flag set, the neighbor router with the highest MAC address takes precedence, and the value advertised by that neighbor is updated in the pseudonode LSP for all neighbors. If some neighbor routers do not set the whole LAN flag, then the reverse metric TLV value advertised by the neighbor router is updated in the pseudonode LSP for that neighbor only.

The S flag is set when the sender of the reverse metric TLV signals to the neighbor router to use the TE sub-tlv for the default metric (sub-tlv type 18) in the reverse metric TLV. When the receiving router finds the S flag set in the reverse metric TLV, the router searches for the TE sub-tlv. The router adds the default metric value in the TE sub-tlv to the configured TE default metric value and recalculates the CSPF.

The **no** form of the command, specified with the configured value, resets the metric value to the default value of 16777214. The **no reverse-metric** command removes the entire reverse metric configuration.

NOTE
The **reverse-metric** *value* command is supported on the Brocade NetIron XMR Series, the Brocade MLX Series, and the Brocade NetIron CER Series and Brocade NetIron CES Series platforms.

Examples The following example configures the reverse metric value to 50 at the router level. The **whole-lan** option is enabled to include the entire LAN.

```
device(config)#router isis
device(config-isis-router)#reverse-metric ?
DECIMAL           Narrow metric range 1-63, Wide metric range 1-16777214,
                  Default is 16777214
te-def-metric   Update TE default metric sub-tlv
tlv-type        Configure reverse metric TLV type
whole-lan       Change metric for whole LAN
device(config-isis-router)#reverse-metric 50 ?
te-def-metric   Update TE default metric sub-tlv
whole-lan       Change metric for whole LAN
<cr>
device(config-isis-router)#reverse-metric 50 whole-lan
device(config-isis-router)#
```

The following example configures the reverse metric TLV type in the range of unassigned IS-IS TLV values.

```
device(config-isis-router)#reverse-metric tlv-type ?
DECIMAL   Configure in the range of unassigned ISIS TLV values
device(config-isis-router)#reverse-metric tlv-type 230
device(config-isis-router)#
```

Use the **show isis config** command to display the configuration of the reverse metric value at the router level. The reverse metric value and the parameters, **whole-lan** and **te-def-metric** are highlighted in the output.

```
device(config)#show isis config
    router isis
 net 49.2211.aaaa.bbbb.cccc.00
 reverse-metric 50 whole-lan te-def-metric
 address-family ipv4 unicast
 exit-address-family

 address-family ipv6 unicast
 exit-address-family
```

History

Release version	Command history
Multi-Service Ironware R05.7.00	This command was introduced.

show metro mp-vlp-queue

Displays priority information about management processor virtual line card (MP-VLP) queues on Brocade NetIron CER Series devices.

Syntax **show metro mp-vlp-queue**

Modes Privileged EXEC mode

Usage Guidelines Use this command to view statistics about messages from the MP are that are queued in the VLP to dequeue.

NOTE
If the Dequeue Time is less than 1 millisecond, it is not recorded in the **show metro mp-vlp-queue** statistics. The corresponding timestamp is also not recorded. The initial timestamp is shown as "0000.00.00-00:00:00.000".

Command Output The **show metro mp-vlp-queue** command displays the following information:

Output field	Description
MP => VLP Queue	The queue priority: high, medium, or low.
Queue Size	The maximum amount of packet counts that the queue can handle at a given time.
Total Pkt Count	The total count of messages queued in each queue.
Current Pkt Count	The count of messages queued at a specific moment in each queue.
Pkt High WM	The maximum messages reached in the queue at any point of time.
Pkt drop Count	The amount of messages that were dropped because the queue was full.
Dequeue High WM(msec)	The longest period of time, in milliseconds, that a message remained in that queue.
Timestamp Pkt High WM(High)	The timestamp for the time when the high water mark for the number of messages in the high priority queue is reached.
Timestamp Pkt High WM(Medium)	The timestamp for the time when the high water mark for the number of messages in the medium priority queue is reached.
Timestamp Pkt High WM(Low)	The timestamp for the time when the high water mark for the number of messages in the low priority queue is reached.
Timestamp Dequeue Time HWM(High)	The timestamp for the time when the most delay is observed in the high priority queue.
Timestamp Dequeue Time HWM(Medium)	The timestamp for the time when the most delay is observed in the medium priority queue.

Output field	Description
Timestamp Dequeue Time HWM(Low)	The timestamp for the time when the most delay is observed in the low priority queue.

Examples

This example shows sample output from the **show metro mp-vlp-queue** command. Three MP-VLP queues are shown with priority High, Medium and Low. The messages from the MP are queued in these queues for the VLP to dequeue.

```
LP-1# show metro mp-vlp-queue

MP => VLP Queue       :         High       Medium          Low
Queue Size            :         2000         2000         2000
Total Pkt Count       :      2160279            0     61210672
Current Pkt Count     :            0            0            0
Pkt High WM           :           13            0         1992
Pkt drop count        :            0            0            0
Dequeue Time HWM(msec):        12000            0        12675

Timestamp Pkt High WM(High)       : [          13]: 2015.02.25-08:07:16.533
Timestamp Pkt High WM(Medium)     : [           0]: 0000.00.00-00:00:00.000
Timestamp Pkt High WM(Low)        : [        1992]: 2015.02.25-08:07:17.223

Timestamp Dequeue Time HWM(High)  : [       12000]: 2015.02.25-08:07:17.230
Timestamp Dequeue Time HWM(Medium): [           0]: 0000.00.00-00:00:00.000
Timestamp Dequeue Time HWM(Low)   : [       12675]: 2015.02.25-08:07:17.800
```

This example shows sample output from the **show metro mp-vlp-queue** command after statistics have been cleared using the **clear metro mp-vlp-queue** command.

```
LP-1# show metro mp-vlp-queue

MP => VLP Queue       :         High       Medium          Low
Queue Size            :         2000         2000         2000
Total Pkt Count       :            0            0            0
Current Pkt Count     :            0            0            0
Pkt High WM           :            0            0            0
Pkt drop count        :            0            0            0
Dequeue Time HWM(msec):            0            0            0

Timestamp Pkt High WM(High)       : [           0]: 0000.00.00-00:00:00.000
Timestamp Pkt High WM(Medium)     : [           0]: 0000.00.00-00:00:00.000
Timestamp Pkt High WM(Low)        : [           0]: 0000.00.00-00:00:00.000

Timestamp Dequeue Time HWM(High)  : [           0]: 0000.00.00-00:00:00.000
Timestamp Dequeue Time HWM(Medium): [           0]: 0000.00.00-00:00:00.000
Timestamp Dequeue Time HWM(Low)   : [           0]: 0000.00.00-00:00:00.000
```

History

Release version	Command history
5.8.00a	This command was introduced.

system-max ip-vrf-route

Configures the maximum number of IPv4 routes that can be created per VRF instance.

Syntax **system-max ip-vrf-route***value*

no system-max ip-vrf-route*value*

Command Default By default, the maximum number of IPv4 routes per VRF instance are not configured.

Parameters *value*

The number of of IPv4 routes that can be created per VRF instance. The values are 128 through 524288. The default value is 1024.

Modes Interface configuration mode.

Usage Guidelines Use the **no** form of the command to reset the maximum number of IPv4 routes that was configured for a VRF instance.

Examples The following example configures 200 IPv4 routes per VRF instance.

```
device# configure terminal
device(config)# system-max ip-vrf-route 200
```

History

Release version	Command history
5.8.00	This command was modified.

system-max ipv6-vrf-route

Configures the maximum number of IPv6 routes that can be created per VRF instance.

Syntax **system-max ipv6-vrf-route***value*

no system-max ipv6-vrf-route*value*

Command Default By default, the maximum number of IPv6 routes per VRF instance are not configured.

Parameters *value*

The number of of IPv6 routes that can be created per VRF instance. The values are 1024 through 131072. The default value is 8192.

Modes Interface configuration mode.

Usage Guidelines Use the **no** form of the command to reset the maximum number of IPv6 routes that was configured for a VRF instance.

Examples The following example configures 4000 IPv6 routes per VRF instance.

```
device# configure terminal
device(config)# system-max ipv6-vrf-route 4000
```

History

Release version	Command history
5.8.00	This command was modified.

Index

B

Border Gateway Protocol 4 (BGP4)
- basic configuration tasks 40
- changing the metric used for route redistribution 61
- graceful restart 33
- MED favoring 67
- parameter changes 38
- parameters 38

C

command
- med-missing-as-worst 67
- network 59

CPU utilization
- displaying statistics 336

G

Global Config
- snmp-server community 397

O

OSPF
- resetting 185

R

Router
- default-metric command
  - default-metric 61
- med-missing-as-worst 67

S

show command
- show ipv6 ospf virtual-link 603
- show process cpu 336