



Extreme OS ONE Switching and Routing v22.2.3.0 Layer 2 Configuration Guide

Switching, LAG, MLAG, and Bridge Domain Setup

9041118-00 Rev AA
August 2026



Copyright © 2026 Extreme Networks, Inc. All rights reserved.

Legal Notice

Extreme Networks, Inc. reserves the right to make changes in specifications and other information contained in this document and its website without prior notice. The reader should in all cases consult representatives of Extreme Networks to determine whether any such changes have been made.

The hardware, firmware, software or any specifications described or referred to in this document are subject to change without notice.

Trademarks

Extreme Networks and the Extreme Networks logo are trademarks or registered trademarks of Extreme Networks, Inc. in the United States and/or other countries.

All other names (including any product names) mentioned in this document are the property of their respective owners and may be trademarks or registered trademarks of their respective companies/owners.

For additional information on Extreme Networks trademarks, see: <https://www.extremenetworks.com/about-extreme-networks/company/legal/trademarks>

Open Source Declarations

Some software files have been licensed under certain open source or third-party licenses.

End-user license agreements and open source declarations can be found at: <https://www.extremenetworks.com/support/policies/open-source-declaration/>

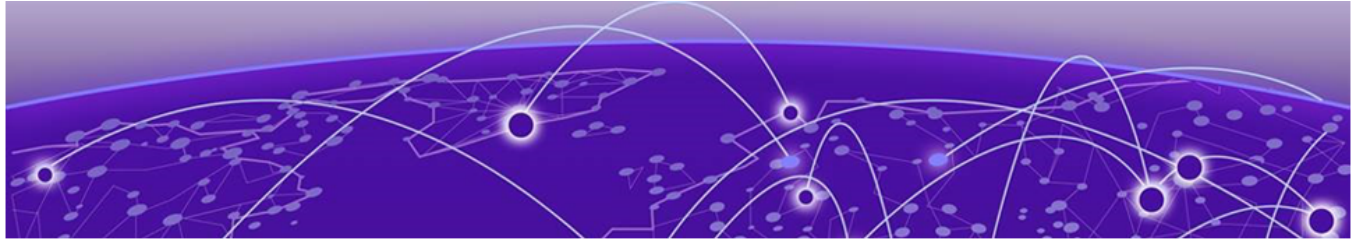
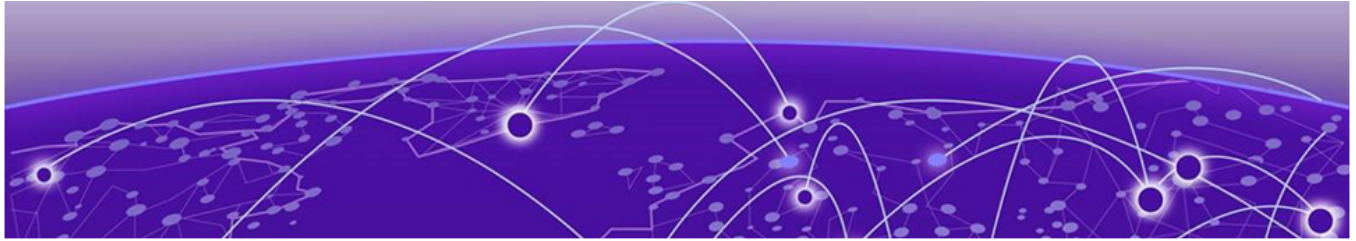


Table of Contents

Abstract.....	vi
Preface.....	vii
Text Conventions.....	vii
Documentation and Training.....	viii
Open Source Declarations.....	ix
Training.....	ix
Help and Support.....	ix
Subscribe to Product Announcements.....	x
Send Feedback.....	x
About This Document.....	11
What's New in This Document	12
Supported Platforms.....	13
Link Aggregation.....	14
Link Aggregation Overview.....	14
Basic LAG Configuration.....	15
Configure a Port Channel Interface.....	15
Delete a Port Channel Interface.....	15
Add a Member Port to a Port Channel.....	16
Delete a Member Port from a Port Channel.....	17
Configure the Minimum Number of LAG Member Links.....	17
Dynamic (LACP) Configuration.....	17
Configure LACP.....	18
Configure LACP System Priority	18
Configure the LACP Interval.....	19
Configure the LACP MAC Address.....	20
Configuring System Priority in LACP.....	22
Supported Show and Clear Commands.....	22
Multi-Chassis Link Aggregation (MLAG).....	25
Overview.....	25
MLAG Limitations.....	26
Other Limitations.....	26
Supported Features.....	26
MLAG Terminology.....	27
MLAG Control Plane.....	27
Inter-Switch Link (ISL)	28
MLAG Peer Role Selection in AUTO Role.....	28
MLAG Control Plane Protocol	28
MLAG Resiliency.....	29
Health Monitoring.....	29
Split-Brain Handling.....	30

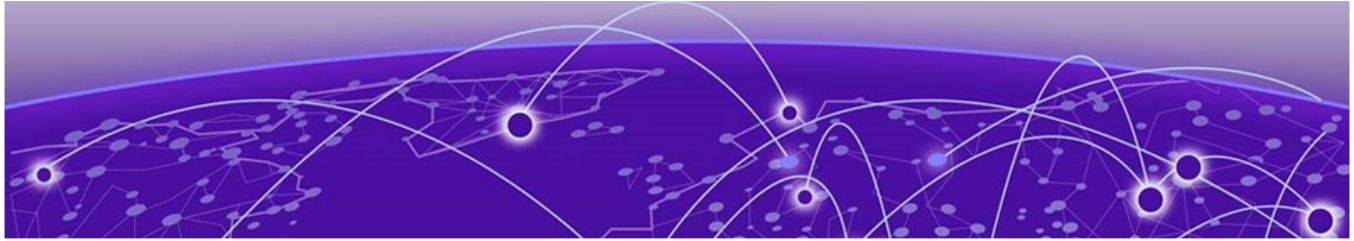
Chassis Manager Responsibilities.....	30
FWD-HAL Responsibilities.....	30
Out-of-Memory Condition.....	30
PSU Failure Detection and Fault Recovery.....	34
Disk Full Detection and Fault Recovery.....	37
Uncontrolled Restart of FWD-HAL.....	42
Uncontrolled Restart of Multiple Services.....	44
I2C and EEPROM Errors on QSFP Modules.....	44
YANG Modules and CLI Commands for MLAG Resiliency.....	46
Using MLAG CLI Commands.....	54
Configure an MLAG Session on Peer Devices.....	57
Monitor MLAG Events and Notifications.....	63
Supported Notifications.....	63
MLAG MIB Definitions.....	63
YANG Module for MLAG Status Retrieval.....	65
Displaying MLAG Addresses in the MAC Address Table for a Bridge Domain.....	66
MLAG Event Log Messages.....	66
RASlogs.....	67
Bridge Domains	69
Bridge Domain Overview.....	69
Bridge Domain Limitations.....	69
Bridge Domain Configuration.....	69
Configure a Subinterface.....	70
Configure a Bridge Domain in VLAN Mode.....	71
Configure a Bridge Domain in Default Mode.....	73
Configure a Static MAC Address.....	74
Configure MAC Learning.....	75
Configure MAC Address Aging.....	77
MAC Movement Detection and Resolution.....	78
MAC Movement Overview.....	78
MAC Movement Detection.....	79
MAC Movement Resolution.....	79
MAC Movement Detection and Resolution Commands.....	81
Multiple Spanning Tree Protocol	86
MSTP Limitations.....	86
CLI Configuration.....	87
Convergence.....	87
Uncontrolled Restart.....	87
MSTP Terminology.....	87
CIST Root Bridge Election.....	88
MSTP Port Roles and Priority Vector.....	89
MSTP Port RolesMSTP Port Roles and Forwarding Behavior.....	89
Priority VectorMSTP Priority Vector Fields and Comparison Order.....	90
MSTP Default Path Cost.....	91
MSTP Port States.....	92
MSTP Timers.....	93
MSTP Configuration.....	94
Enable MSTP.....	95

Add an Interface to MSTP.....	95
YANG Modules and gNMI Operational State Paths for MSTP.....	96
YANG Modules.....	96
gNMI Operational State Paths.....	96
Supported Show Commands.....	97
DHCP Relay.....	99
DHCP Relay Agent Overview.....	99
Purpose of DHCP Relay.....	99
DHCP Relay Protocol Basics.....	99
Key Concepts.....	99
DHCP Relay Features and Capabilities.....	100
Feature Highlights.....	100
Interface and Routing Support.....	100
Helper Address Support.....	100
Gateway Address (giaddr) Handling.....	100
MLAG Redundancy Support.....	101
DHCP Relay in MLAG Deployments.....	101
MLAG Topology Context.....	101
Peer-Local-Physical-giaddr + Option-3-VIP Pattern.....	101
Relay Behavior in MLAG Context.....	102
Expected DORA Event Flow in MLAG.....	102
MLAG Failure Scenarios.....	103
Binding Consistency and Synchronization.....	103
DHCP Relay gNMI Operations.....	103
gNMI Overview for DHCP Relay.....	103
SET Operations (Configuration).....	104
GET Operations (Retrieval).....	105
YANG Path Reference.....	105
DHCP Relay Limitations and Scalability.....	106
Unsupported Features.....	106
Operational Limitations.....	106
Scalability Limits.....	107
Supported Platforms and Interfaces.....	107
Option 82 Limitations.....	107
MLAG-Specific Limitations.....	107
Known Issues.....	107



Abstract

The *Extreme OS ONE Switching and Routing Layer 2 Configuration Guide* version 22.2.2.0 details Layer 2 features and procedures, including static and dynamic LAG (IEEE 802.1AX/LACP), MLAG control-plane design, and bridge domain configuration (VLAN and default modes) for tagged, double-tagged, untagged, and untagged-strict sub-interfaces. This guide is intended for intermediate-to-advanced IT professionals.



Preface

Read the following topics to learn about:

- The meanings of text formats used in this document.
- Where you can find additional information and help.
- How to reach us with questions and comments.

Text Conventions

Unless otherwise noted, information in this document applies to all supported environments for the products in question. Exceptions, like command keywords associated with a specific software version, are identified in the text.

When a feature, function, or operation pertains to a specific hardware product, the product name is used. When features, functions, and operations are the same across an entire product family, such as Extreme Networks switches, the product is referred to as *the switch*.

Table 1: Notes and warnings

Icon	Notice type	Alerts you to...
	Tip	Helpful tips and notices for using the product
	Note	Useful information or instructions
	Important	Important features or instructions
	Caution	Risk of personal injury, system damage, or loss of data
	Warning	Risk of severe personal injury

Table 2: Text

Convention	Description
screen displays	This typeface indicates command syntax, or represents information as it is displayed on the screen.
The words <i>enter</i> and <i>type</i>	When you see the word <i>enter</i> in this guide, you must type something, and then press the Return or Enter key. Do not press the Return or Enter key when an instruction simply says <i>type</i> .
Key names	Key names are written in boldface, for example Ctrl or Esc . If you must press two or more keys simultaneously, the key names are linked with a plus sign (+). Example: Press Ctrl+Alt+Del
<i>Words in italicized type</i>	Italics emphasize a point or denote new terms at the place where they are defined in the text. Italics are also used when referring to publication titles.
NEW!	New information. In a PDF, this is searchable text.

Table 3: Command syntax

Convention	Description
bold text	Bold text indicates command names, keywords, and command options.
<i>italic</i> text	Italic text indicates variable content.
[]	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.
x y	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, such as passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, <i>member[member...]</i> .
\	In command examples, the backslash indicates a “soft” line break. When a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Documentation and Training

Find Extreme Networks product information at the following locations:

[Current Product Documentation](#)

[Release Notes](#)

[Hardware and Software Compatibility](#) for Extreme Networks products

[Extreme Optics Compatibility](#)

[Other Resources](#) such as articles, white papers, and case studies

Open Source Declarations

Some software files have been licensed under certain open source licenses. Information is available on the [Open Source Declaration](#) page.

Training

Extreme Networks offers product training courses, both online and in person, as well as specialized certifications. For details, visit the [Extreme Networks Training](#) page.

Help and Support

If you require assistance, contact Extreme Networks using one of the following methods:

Extreme Portal

Search the GTAC (Global Technical Assistance Center) knowledge base; manage support cases and service contracts; download software; and obtain product licensing, training, and certifications.

The Hub

A forum for Extreme Networks customers to connect with one another, answer questions, and share ideas and feedback. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC.

Call GTAC

For immediate support: (800) 998 2408 (toll-free in U.S. and Canada) or 1 (408) 579 2800. For the support phone number in your country, visit www.extremenetworks.com/support/contact.

Before contacting Extreme Networks for technical support, have the following information ready:

- Your Extreme Networks service contract number, or serial numbers for all involved Extreme Networks products
- A description of the failure
- A description of any actions already taken to resolve the problem
- A description of your network environment (such as layout, cable type, other relevant environmental information)
- Network load at the time of trouble (if known)
- The device history (for example, if you have returned the device before, or if this is a recurring problem)
- Any related RMA (Return Material Authorization) numbers

Subscribe to Product Announcements

You can subscribe to email notifications for product and software release announcements, Field Notices, and Vulnerability Notices.

1. Go to [The Hub](#).
2. In the list of categories, expand the **Product Announcements** list.
3. Select a product for which you would like to receive notifications.
4. Select **Subscribe**.
5. To select additional products, return to the **Product Announcements** list and repeat steps 3 and 4.

You can modify your product selections or unsubscribe at any time.

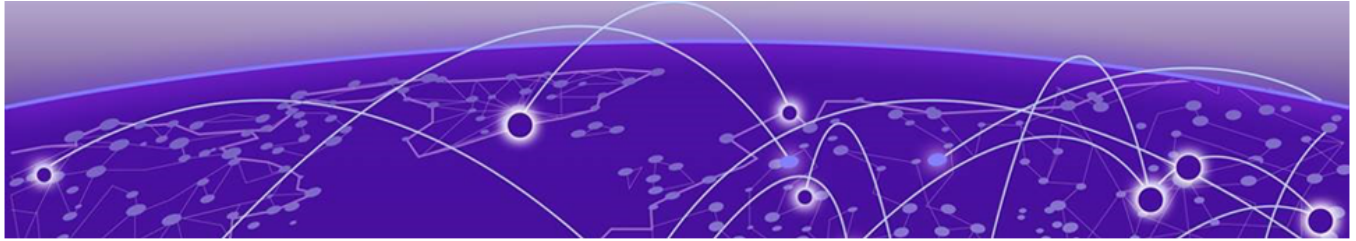
Send Feedback

The User Enablement team at Extreme Networks has made every effort to ensure that this document is accurate, complete, and easy to use. We strive to improve our documentation to help you in your work, so we want to hear from you. We welcome all feedback, but we especially want to know about:

- Content errors, or confusing or conflicting information.
- Improvements that would help you find relevant information.
- Broken links or usability issues.

To send feedback, email us at Product-Documentation@extremenetworks.com.

Provide as much detail as possible including the publication title, topic heading, and page number (if applicable), along with your comments and suggestions for improvement.



About This Document

[What's New in This Document](#) on page 12

[Supported Platforms](#) on page 13

What's New in This Document

The following table describes the information added to this guide for Extreme OS ONE Switching and Routing, release 22.2.3.0:

Feature	Description	Link
PSU Failure Detection	PSU Failure Detection monitors power supply unit status via IPMI commands and triggers a system fault event when the number of available PSUs falls below the minimum required for device operation. When a PSU failure is detected, the system brings down all front-panel interfaces to enable a controlled MLAG failover and maintain network stability.	The following new topics have been added: • PSU Failure Detection and Fault Recovery on page 34 • PSU Failure Detection on page 34 • PSU Fault Recovery on page 36 The following topics are updated: • MLAG Resiliency on page 29 • Post-Reload System Fault Behavior on page 46 • Interface Operational Down Reasons on page 45 • YANG Modules and CLI Commands for MLAG Resiliency on page 46
DHCP Relay Agent	Added support for Dynamic Host Configuration Protocol Relay to enable forwarding of DHCP messages between clients and servers across different IP subnets. Includes global and interface-level configuration support aligned with OpenConfig relay-agent YANG schema, Option 82 (agent information option) support for client identification, per-interface DHCP relay enablement with up to 4 helper addresses, MLAG-aware relay behavior with peer-local giaddr handling, and comprehensive operational visibility through statistics, debugging, and monitoring capabilities.	DHCP Relay on page 99

For additional information, see the *Extreme OS ONE Switching and Routing Release Notes*.

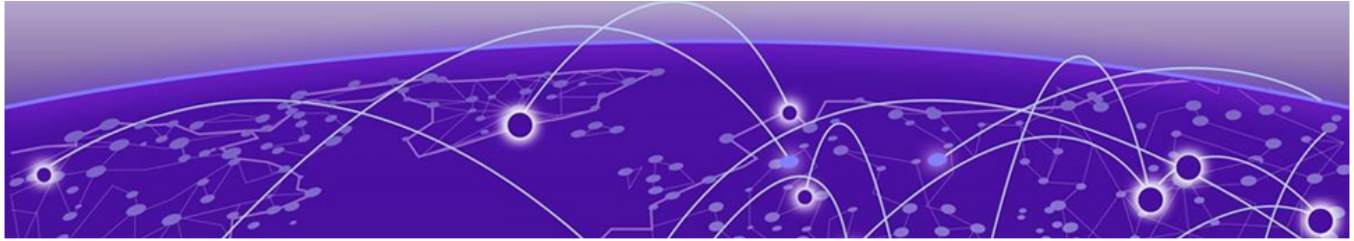
Supported Platforms

Extreme OS ONE Switching and Routing 22.2.3.0 and later releases support Extreme 8520, Extreme 8720, Extreme 8730, and Extreme 8820 hardware platforms.

**Note**

Although this release uses many tested and supported software and hardware configurations, documenting all possible configurations and scenarios is beyond this document's scope.

For information about other releases, see the documentation for those releases.



Link Aggregation

[Link Aggregation Overview](#) on page 14

[Basic LAG Configuration](#) on page 15

[Dynamic \(LACP\) Configuration](#) on page 17

[Supported Show and Clear Commands](#) on page 22

Link Aggregation Overview

You can use link aggregation (LAG) to bundle multiple physical Ethernet links into a single port channel to provide enhanced performance, redundancy, and availability.

We also refer to port channels as link aggregation groups (LAGs). Connected devices, the Spanning Tree Protocol, IEEE 802.1Q VLANs, and so on consider a LAG to be a single link.

When one physical link in the LAG fails, the other links stay up. A failed link causes a small drop in traffic.

When queuing traffic from multiple input sources to the same output port, the system gives all input sources the same weight regardless of whether the input source is a single physical link or a port channel.

Link aggregation has the following benefits:

- Increased bandwidth (the system can change the logical bandwidth dynamically as the demand changes)
- Increased availability
- Load sharing
- Rapid configuration and reconfiguration

Each LAG has the following components:

- Links of the same speed
- A MAC address that differs from the MAC addresses of the LAG's individual member links
- An interface index for each link to identify the link to the neighboring devices
- An administrative key for each link. Only the links with the same administrative key can belong to a LAG. Link Aggregation Control Protocol (LACP) automatically configures an administrative key value equal to the port channel identification number on each link that is configured to use LACP

Extreme OS ONE Switching and Routing supports two LAG types:

- Static — In static link aggregation, the system adds links into a LAG without exchanging control packets between the partner systems. The operational status and administrative state of the link determines the distribution and collection of frames on static links.
- Dynamic — In dynamic link aggregation, LACP negotiates the links included in a LAG. Typically, two partner systems that share multiple physical Ethernet links can aggregate a number of those physical links using LACP. LACP creates a LAG on both partner systems and identifies the LAG by the LAG ID. All links with the same administrative key and all links that you connected to the same partner switch become members of the LAG. LACP continuously exchanges LACP protocol data units (PDUs) to monitor the health of each member link.

Basic LAG Configuration

The topics in this section configure both dynamic (LACP) and static LAG implementations.



Note

You can associate a link with only one port channel.

Configure a Port Channel Interface

Follow this procedure to create a port channel interface at the global configuration level.

1. Access global configuration mode.

```
device# configure terminal
```

2. Create a new port channel interface.

```
device(config)# interface port-channel 1
```

Delete a Port Channel Interface

Follow this procedure to delete a port channel interface and all of its member interfaces.

1. Access global configuration mode.

```
device# configure terminal
```

2. Disable the port channel.

```
device(config)# no interface port-channel 1
```

Add a Member Port to a Port Channel

Follow this procedure to add a member port to a specific port channel interface. If the port channel does not exist, this task creates the port channel and adds a physical interface to it.

1. Access global configuration mode.

```
device# configure terminal
```

2. Add a port channel.

```
device(config)# interface port-channel 1
```

The range is 1 to 255 for a port channel number.

3. Access global configuration mode again.

```
device(config-if-po-1)# exit
```

4. Access interface configuration mode for the physical interface where you want to add to the port channel.

```
device(config)# interface ethernet 0/1
```

5. Enable link aggregation on the physical interface and add it to the port channel. You can specify one of the following modes:

- (For *active* mode) Enable link aggregation on the physical interface and add it to the port channel to enable LACP unconditionally.

```
device(config-if-eth-0/1)# channel-group 1 mode active
```

- (For *passive* mode) Enable link aggregation on the physical interface and add it to the port channel to enable Link Aggregation Control Protocol (LACP) when another LACP device is detected.

```
device(config-if-eth-0/1)# channel-group 1 mode passive
```

- (For *static* mode) Enable link aggregation on the physical interface and add it to the port channel to enable static link aggregation without LACP, which prevents channel formation with other ports that are in active or passive mode.

```
device(config-if-eth-0/1)# channel-group 1 mode on
```

The following example displays the LAG configuration that is running currently on the device. This example configures port channel number 1 and adds Ethernet interface 0/1 to the port channel in active mode:

```
device# show running-config lacp

interface port-channel 1
  no shutdown
  subinterface vlan 1
    ipv4 address 192.0.2.1/24
    ipv6 address 2001:db8:1:1::1/64
  !
!
interface ethernet 0/1
  no shutdown
  channel-group 1 mode active
!
device#
```


Delete a Member Port from a Port Channel

Follow this procedure to delete a member port from a port channel interface at the interface configuration level.

1. Access interface configuration mode for the physical interface that you want to delete from the port channel.

```
device(config)# interface ethernet 0/1
```

2. Delete the port from the port channel interface.

```
device(config-if-eth-0/1)# no channel-group
```

Configure the Minimum Number of LAG Member Links

Follow this procedure to configure the minimum number of Link Aggregation Group (LAG) member links that the Link Aggregation Control Protocol (LACP) bundle can allow.

This configuration lets a port channel operate at a certain minimum bandwidth at all times. If the bandwidth of the port channel drops below the minimum number, then the device declares the port channel operationally DOWN even though it has operationally UP members.

1. Access Global configuration (config) mode.

```
device# configure terminal
```

2. Configure the interface.

```
device(config)# interface port-channel 30
```

3. Configure the minimum number of LAG member links.

```
device(config-if-po-30)# lacp min-links 5
```

The range is 1 to 64. The default is 1.



Note

Always configure **min-links** identically on both ends of the LAG. This prevents asymmetric forwarding and ensures clean failover behavior.

Dynamic (LACP) Configuration

Link Aggregation Control Protocol (LACP) is an IEEE 802.1AX standard that enables two partner systems to negotiate dynamically the attributes of physical links between them to form port channels.

If LACP determines that it can aggregate a link into a Link Aggregation Group (LAG), it puts the link into the LAG. All links in a LAG inherit the same administrative characteristics.

LACP operates in two modes:

- Active: LACP initiates protocol data unit (PDU) exchanges, regardless of whether the partner system sends LACP PDUs.

- Passive: LACP responds to PDUs initiated by its partner system but does not initiate the LACP PDU exchange.

The LACP process collects and distributes Ethernet frames. The collection and distribution process implements the following functionality:

- Inserting and capturing control LACP PDUs
- Restricting the traffic of a conversation to a specific link
- Load balancing links
- Handling dynamic changes in LAG membership

On each port, LACP performs the following tasks:

- Maintains configuration information to control port aggregation
- Exchanges configuration information with other devices to form LAGs
- Attaches ports to and detaches ports from the aggregator when they join or leave a LAG
- Enables or disables an aggregator's frame collection and distribution functions

Configure LACP

1. Access global configuration mode.

```
device# configure terminal
```

2. Enable Link Aggregation Control Protocol (LACP) globally.

```
device(config)# protocol lacp
```

3. (Optional) Disable LACP globally.

```
device(config)# no protocol lacp
```

Configure LACP System Priority

You configure Link Aggregation Control Protocol (LACP) system priority on each device that runs LACP. LACP uses the system priority with the device MAC address to form the system ID and also during negotiation with other devices.

1. Access global configuration mode.

```
device# configure terminal
```

2. Specify the LACP system priority. The range is 1 to 65535. The higher the number, the lower the priority. The default is 32768.

```
device(config)# lacp system-priority 25000
```

3. (Optional) Reset the system priority to the default value.

```
device(config)# no lacp system-priority
```

Configure the LACP Interval

You can configure the protocol data unit (PDU) interval for a port that is a member of a link aggregation group (LAG). This is the interval at which the device sends Link Aggregation Control Protocol (LACP) control packets to an LACP supported interface.

You can choose a fast interval (one second) or a slow interval (30 seconds). The default is a slow interval.

The LACP timeout is triple the LACP interval. This means that for the **fast** setting, the LACP timeout is (1 second) x 3 = 3 seconds. For the **slow** setting, the LACP timeout is (30 seconds) x 3 = 90 seconds.

1. Access global configuration mode.

```
device# configure terminal
```

2. Specify the Ethernet interface for which you want to configure the interval.

```
device(config)# interface ethernet 0/1
```

3. Configure the interval on the interface.

- To set a fast interval (1 PDU per second):

```
device(config-if-eth-0/1)# lacp interval fast
```

- To set a slow interval (1 PDU per 30 seconds):

```
device(config-if-eth-0/1)# lacp interval slow
```

- To restore the default interval (also 1 PDU per 30 seconds):

```
device(config-if-eth-0/1)# no lacp interval
```

The following example adds Ethernet interface 0/1 to a LAG that has port channel number 10, specifies active LACP mode on the interface, and sets the interval of the interface to 1 PDU per second:

```
device# configure terminal
device(config)# interface ethernet 0/1
device(config-if-eth-0/1)# channel-group 10 mode active
device(config-if-eth-0/1)# lacp interval fast
device(config-if-eth-0/1)#
```

The following example displays the LAG configuration that is running currently on the device. This example adds Ethernet interface 0/1 to a LAG in active mode with a fast interval (1 PDU per second):

```
device# show running-config lacp

protocol lacp
lacp system-priority 32768
!
interface port-channel 10
lacp min-links 1
no shutdown
!
interface ethernet 0/1
channel-group 10 mode active
lacp interval fast
no shutdown
!
device#
```

The following example adds Ethernet interface 0/2 to a LAG that has port channel number 20, specifies active LACP mode on the interface, and sets the interval of the interface to 1 PDU per 30 seconds:

```
device# configure terminal
device(config)# interface ethernet 0/2
device(config-if-eth-0/2)# channel-group 20 mode active
device(config-if-eth-0/2)# lacp interval slow
device(config-if-eth-0/2)#
```

The following example displays the LAG configuration that is running currently on the device. This example adds Ethernet interface 0/2 to a LAG in active mode with a show interval (1 PDU per 30 seconds):

```
device# show running-config lacp

protocol lacp
lacp system-priority 32768
!
interface port-channel 20
lacp min-links 1
no shutdown
!
interface ethernet 0/2
channel-group 20 mode active
lacp interval fast
no shutdown
!
device#
```

Configure the LACP MAC Address

You can configure the Link Aggregation Control Protocol (LACP) MAC address of a link aggregation group (LAG).

1. Access global configuration mode.

```
device# configure terminal
```

2. Specify the port channel interface for which you want to configure the MAC address.

```
device(config)# interface port-channel 1
```

3. (Optional) Validate the port channel health and statistics information.

```
device(config)# do show counters interface port-channel 101

Interface Statistics: port-channel 101
  Carrier Transitions: 1
    LastClear: 0s
Input:
  Total pkts: 1477460957
  Broadcast pkts: 87289588
  Discard pkts: 0
  Errors pkts: 0
    FCS Errors: 0
  MCast pkts: 87378080
    Octets: 1126773024923
  UCast pkts: 1302793289
  Runt pkts: 0
  CRC Errors: 0
Input Distribution:
  64 byte pkts: 5868861
```

```

        65-127 byte pkts: 50932300
        128-255 byte pkts: 573604220
        256-511 byte pkts: 66253377
        512-1023 byte pkts: 347495781
        1024-1518 byte pkts: 211842668
        Jumbo pkts: 221463750
Out:
        Total pkts: 2416872288
        Broadcast pkts: 522284647
        Discard pkts: 0
        Errors pkts: 0
        MCast pkts: 272962830
        Octets: 2360836407771
        UCast pkts: 1621624811
Rate Info:
        Input: 116.925313 Mbits/sec, 20726 pkts/sec 0.29% of line-rate
        Output: 288.130020 Mbits/sec, 35007 pkts/sec 0.72% of line-rate
device(config)#

```

4. Configure the system MAC address.

```
device(config-if-po-1)# lacp system-mac 66:fc:1d:1f:5b:85
```

5. (Optional) Reset the LACP system MAC address to the default value. This is a MAC address whose last octet is one more than that of the MAC address of the immediately prior interface.

```
device(config-if-po-1)# no lacp system-mac
```

The following example enables LACP globally on the device and configures a Link Aggregation Group (LAG) with port channel number 10 and a system MAC address of 01:23:45:67:89:ab.

```

device# configure terminal
device(config)# protocol lacp
device(config)# interface port-channel 10
device(config-if-po-10)# lacp system-mac 01:23:45:67:89:ab
device(config-if-po-10)#

```

The following example displays the LAG configuration that the device is running currently. This example enables LACP globally, specifies a LAG with port channel number 10 and a system MAC address of 01:23:45:67:89:ab, and adds Ethernet interface 0/29 to the LAG in active mode:

```

device# show running-config lacp

protocol lacp
interface port-channel 10
lacp system-mac 01:23:45:67:89:ab
no shutdown
!
interface ethernet 0/29
channel-group 10 mode active
no shutdown
!
device#

```

Configuring System Priority in LACP

In Link Aggregation Control Protocol (LACP), the system priority determines which device takes the lead during link aggregation negotiations. You can set a value that influences which device takes the lead in a multi-device setup.

The default is 32768, and the range is 1 to 65535. You must ensure that the value is the same across the device.

To set or unset LACP system priority, use the following command:

```
device(config-if-po-101)# lacp system-priority
```

Supported Show and Clear Commands

- **show interface port-channel:** Displays details for all port channels.

```
device# show interface port-channel

  IFNAME  Po Value from 1-255
  brief   brief
device# show interface port-channel 101

device# show interface port-channel 101
port-channel 101 is up
  MTU 9216 Bytes
  IfIndex 0x4000065
  Mac address is 88:7e:25:d3:da:14
  Port mode is Full Duplex, 40G
  MinLinks is 1
  LagType is LACP
  Active Members in this channel: Eth 0/3
  Members in this channel: Eth 0/3
Statistics
  Carrier Transitions: 1
    LastClear: 0s
Input:
  Broadcast pkts: 87225518
  Discard pkts: 0
  Errors pkts: 0
    FCS Errors: 0
  MCast pkts: 87313943
    Octets: 1126070215752
  UCast pkts: 1301896993
  Unknown Protocols: 0
Out:
  Broadcast pkts: 521900301
  Discard pkts: 0
  Errors pkts: 0
  MCast pkts: 272758354
    Octets: 2359154212029
  UCast pkts: 1620447729

device# show interface port-channel brief

Flags:  M - Redundant Management  P - Performance-Path
Number of interfaces 21
Port    Mtu      Admin-State Oper-State  Speed    Ifindex    Description
-----
Po 10   9216      UP          UP          75G      0x400000a  ISL_Underlay_PO
Po 53   9216      UP          UP          75G      0x4000035  TO-DUT3
```

```

Po 54      9216      UP      UP      75G      0x4000036      TO-DUT4
Po 101     9216      UP      UP      40G      0x4000065      Port-Channel 101
Po 102     9216      UP      UP      20G      0x4000066      Port-Channel 102
Po 127     9216      UP      DOWN    0G      0x400007f      Port-Channel 127
Po 151     9216      UP      UP      100G     0x4000097      Port-Channel 151
Po 152     9216      UP      UP      30G      0x4000098      Port-Channel 152
Po 153     9216      UP      UP      20G      0x4000099      Port-Channel 153
Po 154     9216      UP      UP      10G      0x400009a      Port-Channel 154
Po 155     9216      UP      UP      10G      0x400009b      Port-Channel 155
Po 156     9216      UP      UP      40G      0x400009c      Port-Channel 156
Po 157     9216      UP      UP      100G     0x400009d      Port-Channel 157
Po 158     9216      UP      UP      10G      0x400009e      Port-Channel 158
Po 159     9216      UP      UP      10G      0x400009f      Port-Channel 159
device#

```

- **show counters interface port-channel:** Displays statistics for a specific port channel.

```

device# show counters interface port-channel 101

Interface Statistics: port-channel 101
  Carrier Transitions: 1
    LastClear: 0s
Input:
  Total pkts: 1477460957
  Broadcast pkts: 87289588
  Discard pkts: 0
  Errors pkts: 0
  FCS Errors: 0
  MCast pkts: 87378080
    Octets: 1126773024923
  UCast pkts: 1302793289
  Runt pkts: 0
  CRC Errors: 0
Input Distribution:
  64 byte pkts: 5868861
  65-127 byte pkts: 50932300
  128-255 byte pkts: 573604220
  256-511 byte pkts: 66253377
  512-1023 byte pkts: 347495781
  1024-1518 byte pkts: 211842668
  Jumbo pkts: 221463750
Out:
  Total pkts: 2416872288
  Broadcast pkts: 522284647
  Discard pkts: 0
  Errors pkts: 0
  MCast pkts: 272962830
    Octets: 2360836407771
  UCast pkts: 1621624811
Rate Info:
  Input: 116.925313 Mbits/sec, 20726 pkts/sec 0.29% of line-rate
  Output: 288.130020 Mbits/sec, 35007 pkts/sec 0.72% of line-rate
device#

```

- **show counters lacp:** Displays statistics for member ports of dynamic Link Aggregation Groups (LAGs).

```

device# show counters lacp

Port              in-pkts   out-pkts   TxErr   RxErr   unknownErr   LacpErr
-----
Channel group: 10
ethernet 0/7:1    393       2724       0        0        0            0
ethernet 0/7:2    401       2723       0        0        0            0
ethernet 0/7:3    403       2723       0        0        0            0

```

```

ethernet 0/7:4      396      2723      0      0      0      0
device#

```



Note

The LACP microservice fetches counter statistics for out-pkts from the kernel every 10 sec. This means that counters are refreshed every 10 seconds. In-Pkts are updated every second.

- **show lacp system-identifier:** Displays the unique identifier assigned to a LAG.

```

device# show lacp system-identifier

      System ID: 0x8000, 00:04:96:d6:83:e0
device#

```

- **show lacp interface ethernet x/x:** Displays statistics for specific member ports of dynamic LAGs.

```

device# show lacp interface ethernet 0/7:2

interface Eth 0/7:2 is up
  Channel group is 10 port channel is Po10
  PDUs sent: 2943
  PDUs rcvd: 614
  LACP Rx errors: 0
  LACP Tx errors: 0
  LACP unknown errors: 0
  LACP errors: 0
Local Port: Eth 0/7:2   MAC Address = 00:04:96:d6:83:e0
System Identifier = 80:00:00:04:96:d6:83:e0
Port Identifier = 0x8000, 0x207
Operational key = 10
LACP_Activity = active
LACP_Timeout = short Timeout (1s)
Synchronization = IN_SYNC
Collecting = true
Distributing = true

Partner information
  Partner-id = 80:00:00:04:96:d6:55:1c
  Partner-key = {10, 519}
device#

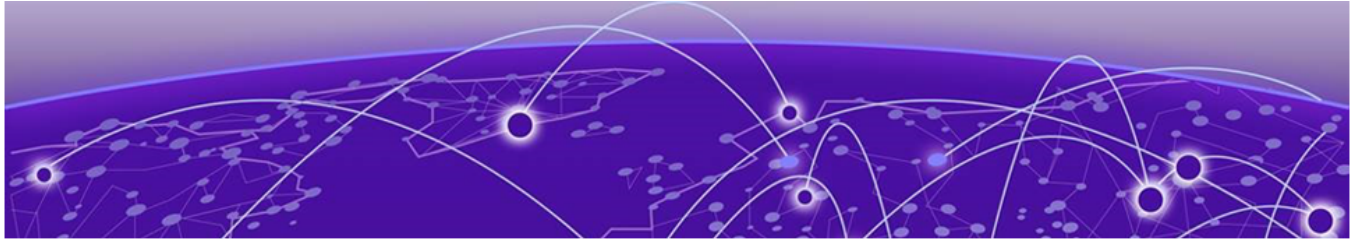
```

- **clear counters lacp:** Clears statistics for member ports of the dynamic LAGs that are configured on the device.

```

device# clear counters lacp
device# clear counters lacp interface port-channel 10-20
device#

```

Multi-Chassis Link Aggregation (MLAG)

[Overview](#) on page 25

[MLAG Limitations](#) on page 26

[MLAG Terminology](#) on page 27

[MLAG Control Plane](#) on page 27

[MLAG Resiliency](#) on page 29

[Using MLAG CLI Commands](#) on page 54

[Configure an MLAG Session on Peer Devices](#) on page 57

[Monitor MLAG Events and Notifications](#) on page 63

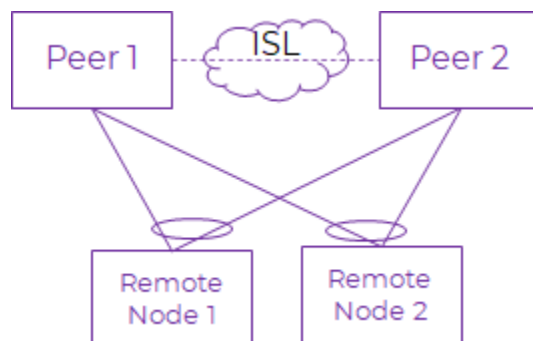
Overview

Multi-Chassis Link Aggregation (MLAG) lets one device (such as a server) connect to two switches at the same time, but it looks like just one connection to the server.

A LAG (link aggregation group) normally connects to one switch, so if that switch fails, the entire group goes down. MLAG splits the connections split across two switches that work together. The switches use an Inter-Switch Link (ISL) to communicate with each other to act like a single system. This means that if one switch fails, the other one keeps the connection alive to provide redundancy and more bandwidth.

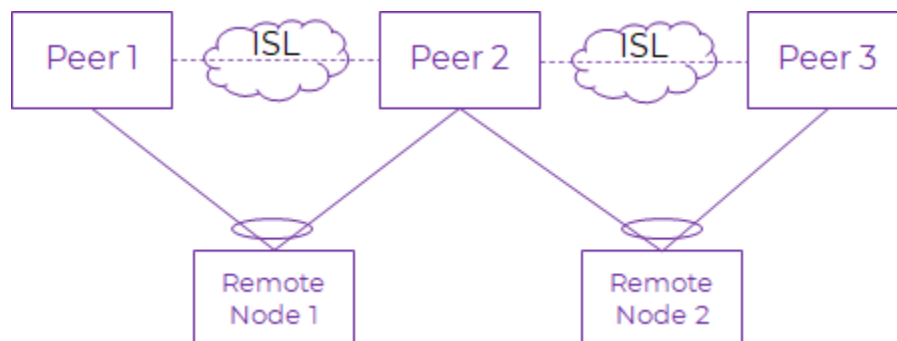
The Extreme OS ONE Switching and Routing Layer 2 MLAG control plane protocol (MCP) synchronizes MAC and ARP data between the MLAG peers for resiliency and faster convergence.

MLAG establishes the data plane using a Virtual eXtensible Local Area Network (VxLAN) tunnel between MLAG peers as in the following figure:



MLAG Limitations

Multichassis Link Aggregation Group (MLAG) supports only one peer device. It does *not* support the following topology, which contains multiple peers:



Other Limitations

- You cannot configure MLAG keep-alive source-interface as Virtual Ethernet (VE).
- Only the default VRF supports MLAG primary keep-alive.
- Ethernet MLAG interfaces do not support MLAG ID configuration as AUTO.
- Copper ports can remain operationally UP during a Forwarding Hardware Abstraction Layer (FWD-HAL) microservice crash.
- Recommendation: Physical interface or Port-channel used as underlay for VxLAN tunnel or ISL interface should not be part of any Bridge-domain or sub-interface configuration.

Supported Features

MLAG supports only system fault generation for the following features:

- Uncontrolled restart of the Forwarding Hardware Abstraction Layer (FWD-HAL) microservice
- Out of memory conditions
- PSU failure detection

MLAG Terminology

MLAG	Multichassis Link Aggregation Group
ISL	Inter-Switch Link. This link connects the two physical devices that form an MLAG. The ISL is the dedicated, high-bandwidth connection between two MLAG peer devices. It synchronizes control traffic, such as MAC addresses and IGMP states, and carries data traffic if one switch loses its downstream link, ensuring high availability. An ISL allows two physically separate switches to act as a single logical entity to downstream devices, providing a unified control plane. The link transfers critical control information and to bridge data traffic between peers during failures or specific hash scenarios. You generally configure an ISL as a direct, high-speed, port-channel connection between the two switches, separate from traffic directed to servers.
VxLAN	Virtual eXtensible LAN
MCP	MLAG Control Protocol
MLAG VLANs	VLANs that the MLAG peers share. You configure these VLANs explicitly in the bridge domain configuration of the MLAG.
Member PO	A port channel (PO) that is a member of an MLAG group. These are the ports that belong to the port channel that forms the MLAG link. These links connect a device to both of the MLAG peer devices. For example, a server might have two NICs, with each connected to a different device, and the system bundles these NICs into a port channel that is a member of the MLAG. This makes the server connection appear as a single, highly available, and high bandwidth link.
Non-MLAG link	A standard network link that is not part of an MLAG. In contrast to MLAG, a non-MLAG link is a simple connection between two devices or a standard Link Aggregation Group (LAG) that combines only multiple links into a single logical one between two devices. A non MLAG link is any link that does not use this specific technology.
BD	Bridge Domain
VNI	Virtual Network Identifier
VE	Virtual Ethernet interface

MLAG Control Plane

Multichassis Link Aggregation Group (MLAG) establishes a primary keepalive session based on a gRPC Remote Procedure Call (gRPC) connection using port 4012 between the peers using the peer IP address of the configured primary keepalive and the IP of source interface. The primary keepalive session helps to establish an initial connection. If the Bidirectional Forwarding Detection (BFD) service is available, the system offloads the session to BFD for further management such as monitoring.

MLAG establishes a secondary keepalive session using a TCP connection using port 4000 between the peers using the IP address of the configured secondary keepalive and the IP of source interface if manually configured. By default, a secondary keepalive establishes a connection through the management interface automatically.

Inter-Switch Link (ISL)

ISL for is a Virtual eXtensible LAN (VxLAN) tunnel created between the Multi-Chassis Link Aggregation (MLAG) peers. The destination IP address of the ISL tunnel is the peer IP address of the MLAG peer. You configure the ISL source IP address on the source interface of the primary keepalive session.

The underlay interface carrying the traffic is any physical port or port channel Layer 3 interface between the MLAG peers. By default, the system extends all MLAG VLANs or bridge domains to the MLAG peer.

By default, the system configures VLAN to VxLAN Network Identifier (VLAN-VNI) mapping automatically for the ISL VxLAN tunnel. Because you can use a single VLAN-VNI mapping domain, any change to this mapping under the overlay gateway changes the mapping for the ISL and affects its traffic temporarily.

ISL does not support Virtual Ethernet (VE) interfaces.

MLAG Peer Role Selection in AUTO Role

The device elects the MLAG peer role based on the source IP address of the primary keepalive in the AUTO role. Otherwise, the device elects the peer with the highest primary keepalive source IP address as PRIMARY and uses the lowest as the BACKUP peer. You should configure the device so that if a peer has an AUTO role, the other peer also has an AUTO role.

MLAG Control Plane Protocol

MLAG peers send health check (HC) messages periodically. These messages travel over a gRPC Remote Procedure Call (gRPC) connection to 4012 port number of the peer. Peers send health check hellos every keepalive interval, where the keepalive interval is a configurable parameter with a range of 100 ms to 1000000 ms, defaulting to 300 ms.

You must configure the primary keepalive for peer liveness detection. MLAG declares the peer DOWN if both the primary and secondary keepalive statuses are DOWN.

In addition to keepalive messages, MLAG Control Plane Protocol (MCP) also synchronizes MAC and Address Resolution Protocol (ARP) messages and exchanges system parameters. These include the management IP address, configured bridge domains, and the system MAC address.

The primary keepalive service establishes the initial connection between peers and then offloads them to Bidirectional Forwarding Detection (BFD) for ongoing monitoring. BFD uses an MLAG profile to configure transmission intervals and detection multipliers. By default, BFD operates with a 300 millisecond interval and a detection multiplier of 3.

MLAG Resiliency

MLAG resiliency enhances the fault tolerance of a Multichassis Link Aggregation Group (MLAG) by detecting and responding to critical system failures such as out of memory condition, uncontrolled restart of the Forwarding Hardware Abstraction Layer (FWD-HAL) service of the primary MLAG device, PSU failure, or Quad Small Form-factor Pluggable (QSFP) Inter-Integrated Circuit (I2C) errors on the primary MLAG device. MLAG resiliency feature ensures continuous network availability.

The primary objective is to prevent split brain mode during system faults and maintain acceptable convergence times.

Health Monitoring

The monitoring service contains the health monitoring logic, which acts as a gRPC Remote Procedure Call (gRPC) server. It listens for system fault events from various services including memory and CPU utilization issues. When it detects a fault, it publishes it on the message bus to make it available to all subscribed services (including MLAG).

Use the following command to check the local peer health.

```
device# curl 0:9004/show-peerdB

Dumping MLAG Config Data Structures:
*****

-----
Keepalive Interval      : ---
Keepalive Delay        : ---
Bringup Delay          : 30
Multiplier             : ---
Role                   : ---
Mac                    : ---
Mgmt IP                : 10.38.59.158 Idx: 22000001
System MAC             : 00:16:3e:54:e1:00
BringUpDelayTmrSt      : false
Local MaintenanceMode  : Disabled
Local Health           : Healthy
device#
```

Use the following command to check the remote peer health.

```
device# show mlag peer

Peer dut3
=====
Peer State           : UP
MCP State            : UP
Role                 : BACKUP
Elected MAC         : 02:00:22:33:44:55
Extend Bridge Count  : 46
Peer Exception       : Peer Under Unhealthy State
device#
```

Split-Brain Handling

A split-brain scenario occurs when the primary keepalive link is down but the secondary keepalive remains active. In this case, the Multichassis Link Aggregation Group (MLAG) backup peer typically shuts down all its client ports to prevent network inconsistencies. However, if the backup peer has already received a peer unhealthy notification, it does not shut down its client ports to avoid traffic blackholing.

Chassis Manager Responsibilities

In certain situations, such as a kernel panic, Forwarding Hardware Abstraction Layer (FWD-HAL) service restart, or broken communication between MLAG peers, the peer health information might not arrive successfully at the peer.

FWD-HAL Responsibilities

The MLAG service communicates this information to the peer using its keepalive service when the monitoring services declare that the system is healthy.

Out-of-Memory Condition

When system memory falls below a certain threshold, the kernel might reload the device. To prevent this, the monitor-svc service takes the following actions:

1. Monitors the system memory every five seconds
2. Calculates the available memory using the MemAvailable field from /proc/meminfo
3. Generates a system fault if the available memory falls below 1 GB
4. Clears a fault when the available memory exceeds 1.5 GB

Out of Memory Handling

A device under an Out of Memory (OOM) condition responds by coordinating with platform scripts and Multichassis Link Aggregation Group (MLAG) services. It listens for OOM alerts from the monitoring service and, when triggered, brings down all physical interfaces and notifies the MLAG peer devices of an unhealthy state to prevent a split brain scenario. Once the system recovers, it helps restore network connectivity by bringing up interfaces and re-enabling MLAG functions.

```
device# show interface brief
```

```
Flags: M - Redundant Management P - Performance-Path
```

```
Number of interfaces 124
```

Port Description	Mtu	Admin-State	Oper-State	Speed	Ifindex
-----	-----	-----	-----	-----	-----
Int 0(P)	9216	UP	UP	10G	0x21000000
Internal 0					
Eth 0/1	9216	UP	DOWN (SYSTEM_FAULT)	100G	0x1000020
towards-Leaf3					
Eth 0/2	9216	UP	DOWN (SYSTEM_FAULT)	100G	0x1000040
towards-Leaf4					
Eth 0/3	9216	DOWN	DOWN	100G	0x1000060
Ethernet 0/3					

```

Eth 0/4(M)          9216    UP        UP                100G    0x1000080
Ethernet 0/4
Eth 0/5:1           9216    UP        DOWN (SYSTEM_FAULT) 25G     0x10000a1
Ethernet 0/5:1
Eth 0/5:2           9216    UP        DOWN (SYSTEM_FAULT) 25G     0x10000a2
Ethernet 0/5:2
Eth 0/5:3           9216    DOWN     DOWN              25G     0x10000a3
Ethernet 0/5:3
Eth 0/5:4           9216    DOWN     DOWN              25G     0x10000a4
Ethernet 0/5:4
Eth 0/6             9216    UP        DOWN (SYSTEM_FAULT) 100G    0x10000c0
Ethernet 0/6
Eth 0/7             9216    UP        DOWN (SYSTEM_FAULT) 100G    0x10000e0
Ethernet 0/7
Eth 0/8             9216    UP        DOWN (SYSTEM_FAULT) 100G    0x1000100
Ethernet 0/8
Eth 0/9             9216    UP        DOWN (SYSTEM_FAULT) 100G    0x1000120
Ethernet 0/9
Eth 0/10            9216    UP        DOWN (SYSTEM_FAULT) 100G    0x1000140
Ethernet 0/10
Eth 0/11            9216    UP        DOWN (SYSTEM_FAULT) 100G    0x1000160
Ethernet 0/11

:

:

:

Eth 0/31:1          9216    UP        DOWN (SYSTEM_FAULT) 25G     0x10003e1
member port of po 78
Eth 0/31:2          9216    UP        DOWN (SYSTEM_FAULT) 25G     0x10003e2
member port of po 78
Eth 0/31:3          9216    UP        DOWN (SYSTEM_FAULT) 25G     0x10003e3
member port of po 78
Eth 0/31:4          9216    UP        DOWN (SYSTEM_FAULT) 25G     0x10003e4
member port of po 78
Eth 0/32:1          9216    UP        DOWN (SYSTEM_FAULT) 25G     0x1000401
member port of po 78
Eth 0/32:2          9216    UP        DOWN (SYSTEM_FAULT) 25G     0x1000402
member port of po 78
Eth 0/32:3          9216    UP        DOWN (SYSTEM_FAULT) 25G     0x1000403
member port of po 78
Eth 0/32:4          9216    UP        DOWN (SYSTEM_FAULT) 25G     0x1000404
member port of po 78
Po 18               9216    UP        DOWN              0G      0x4000012
Port-Channel 18
Po 28               9216    UP        DOWN              0G      0x400001c
Port-Channel 28
Po 78               9216    DOWN     DOWN              0G      0x400004e    Po b/w
9150-1, Leaf3, Leaf4
Po 101              9216    UP        DOWN              0G      0x4000065    Port-
Channel 101
Po 102              9216    UP        DOWN              0G      0x4000066    Port-
Channel 102
Tu ISL_10.7.8.7     UP        DOWN              0x3100001
Tunnel ISL_10.7.8.7
Ve 101              9216    UP        DOWN (NO_ACTIVE_MEMBERS) 0x5000065    VE
101

Device# show system health

-----
DEGRADED-COMPONENTS

```

```
-----
Component: Memory
Reason: insufficient memory available (< 953.7 MiB)
Time: 2025-02-03 15:04:05
device#
```

With an OOM condition, MLAG communicates unhealthy peer information to the peer via the keepalive and system services.

When MLAG clears an OOM condition, it starts the bringup delay timer. Once the timer expires, MLAG brings up the interfaces and uplink track interfaces and sends a Node Health notification to the peer.

Split Brain Handling

A split brain scenario occurs when the primary keepalive link is down but the secondary keepalive remains active. In this case, the MLAG backup peer typically shuts down all its client ports to prevent network inconsistencies. However, if the backup peer has already received a peer unhealthy notification, it will not shut down its client ports to avoid traffic blackholing.

Fault Flapping Prevention

To avoid continuous flapping between low memory and sufficient memory, the system monitors the number of flaps. If this number exceeds 5 flaps, the fault is not cleared.

OOM Event Update

An OOM event updates specific key paths with OOM event information. For example, an OOM event updates the following key path.

```
key /components/component[name=degraded-component]
{
  "name": "degraded-component",
  "subcomponents": {
    "subcomponent": [
      {
        "name": "Memory",
        "state": {
          "name": "Memory"
        }
      }
    ]
  }
}

key /components/component[name=Memory]
{
  "name": "Memory",
  "properties": {
    "property": [
      {
        "name": "reason",
        "state": {
          "configurable": false,
          "value": "insufficient memory available (less than 953.7 MiB)"
        }
      },
      {
```



```

    "name": "timestamp",
    "state": {
      "configurable": false,
      "value": "2025-02-03 06:59:36"
    }
  },
  {
    "name": "count",
    "state": {
      "configurable": false,
      "value": "0"
    }
  }
],
"state": {
  "name": "Memory",
  "type": "health"
}
}

```

Show System Health CLI

Use the **show system health** command to display information about the OOM event. This includes the following details:

- Memory usage statistics
- Degraded components (memory)
- Reason for degradation (insufficient memory available)

```

device# show system health

BIOS Vendor: SeaBIOS
BIOS Version: 1.13.0-lubuntu1.1
BIOS Date: 04/01/2014
Kernel Version: 5.10.210-yocto-standard
Kernel Arch: x86_64
CPU Procs: 6
OS: linux
Platform: alpine
Platform Version: 3.20.3
Memory Total: 7.63GB
Memory Free: 0.91GB
Memory Used: 6.48GB
Memory Used (%): 84.00%
CPU Model: Intel Xeon Processor (Cascadelake)
Cpu Load: 37
Primary Rootfs Disk Total: 16.37GB
Primary Rootfs Disk Free: 11.70GB
Primary Rootfs Disk Used: 3.82GB
Secondary Rootfs Disk Total: 17.00GB
System Uptime: 4m31s
RAM Caches(kB): 911252

RAM Free(kB): 227316
Ram Used(kB): 6842488
RAM Buffers(kB): 23052
Cpu SoftIRQ time: 0
Cpu System time: 3
Cpu IOWait time: 7
Cpu User time: 26
USRDATA Partition Used: 22.00%
APPDATA Partition Used: 8.00%
Secondary Rootfs Disk Used: 3.90GB

```

```

Disk Written(kB): 0
Secondary Rootfs Disk Free: 12.00GB
CONFIG Partition Used: 1.00%
IAH Partition Used: 1.00%
Disk Read(kB): 4184

-----
DEGRADED-COMPONENTS
-----

Component: Memory
Reason: insufficient memory available (less than 953.7 MiB)
Time: 2025-02-03 06:59:36
device#

```

PSU Failure Detection and Fault Recovery

Multi-Chassis Link Aggregation (MLAG) provides resiliency against power supply unit (PSU) failures. When the number of online PSUs drops below the minimum required for the platform, the system transitions all front-panel and MLAG member interfaces to an operationally down state to prevent traffic black-holing. After PSU redundancy is restored, the system applies a bring-up delay before recovering MLAG member interfaces to ensure that forwarding state synchronization with the peer node completes before client traffic resumes.

PSU Failure Detection

The chassis manager (chassis-mgr) monitors the status of all installed power supply units (PSUs) through IPMI commands. Each device has a minimum required number of operational PSUs.

When the number of online PSUs drops below this minimum, chassis-mgr reports a system fault to the monitoring service (monitor-svc). The system performs the following actions when it detects a PSU fault:

1. Publishes the PSU fault event on the message bus
2. Brings all front-panel interfaces operationally DOWN with the reason code `SYSTEM_FAULT`



Note

Port-channels also transition to operationally DOWN when all member ports enter `SYSTEM_FAULT`.

3. Communicates the unhealthy peer state to the MLAG peer through the keepalive and system services
4. Recognizes the unhealthy state and avoids entering split-brain mode.

When a PSU fault occurs, all front-panel Ethernet ports (if administratively UP), port-channels, and the ISL tunnel transition to operationally DOWN with the reason code `SYSTEM_FAULT`:

```

device# show interface brief

```

Port	Mtu	Admin-State	Oper-State	Speed	Ifindex	Description
Eth 0/1	9216	UP	DOWN (SYSTEM_FAULT)	100G	0x1000020	Ethernet 0/1
Eth 0/2	9216	UP	DOWN (SYSTEM_FAULT)	100G	0x1000040	Ethernet 0/2
Eth 0/3	9216	UP	DOWN (SYSTEM_FAULT)	100G	0x1000060	Ethernet 0/3

Eth 0/4	9216	UP	DOWN (SYSTEM_FAULT)	100G	0x1000080	Ethernet 0/4
Po 64	9216	UP	DOWN	0G	0x4000040	Port-Channel 64
Tu ISL_10.20.20.2		UP	DOWN		0x3100001	Tunnel ISL_10.20.20.2

ISL Tunnel Behavior

When the system detects a PSU fault, the interface manager brings the ISL VxLAN tunnel operationally DOWN along with all front-panel interfaces. The MLAG peer detects the ISL loss and redirects traffic to its local paths. The ISL tunnel recovers automatically after the system clears the PSU fault and the bring-up delay timer expires.

Minimum PSU Requirements

Each device requires a minimum number of operational PSUs. For more information, see the hardware installation documentation.

Monitoring Service Update

The monitor-svc updates the following SDB path when a PSU fault occurs:

```
key /components/component[name=degraded-component]
{
  "name": "degraded-component",
  "subcomponents": {
    "subcomponent": [
      {
        "name": "PSU",
        "state": {
          "name": "PSU"
        }
      }
    ]
  }
}

key /components/component[name=PSU]
{
  "name": "PSU",
  "properties": {
    "property": [
      {
        "name": "reason",
        "state": {
          "configurable": false,
          "value": "insufficient number of power supply units in the device"
        }
      },
      {
        "name": "timestamp",
        "state": {
          "configurable": false,
          "value": ""
        }
      },
      {
        "name": "count",
        "state": {
          "configurable": false,
          "value": "0"
        }
      }
    ]
  }
}
```

```
    ]
  }
}
```

Monitoring Service Delete Notification

To clear the fault, monitor-svc sends a delete notification on the subcomponent:

```
key /components/component[name=degraded-component]
{
  "name": "degraded-component",
  "subcomponents": {
    "subcomponent": []
  }
}
```

Show System Health CLI

Use the **show system health** command to verify PSU fault status:

```
device# show system health
-----
DEGRADED-COMPONENTS
-----
Component: PSU
Reason: Insufficient online PSUs
Time: 2026-08-10 19:43:32
System Fault Triggered: True
```

PSU Fault Recovery

When the system restores PSU redundancy, MLAG uses a bring-up delay timer to ensure that MAC and ARP synchronization completes before restoring client traffic.

The system performs the following steps when a PSU returns to service:

1. Detects the PSU coming online through the chassis manager
2. Clears the system fault condition on the monitoring service
3. Brings up all non-MLAG front-panel interfaces and uplink track ports
4. Starts the 90-second PSU Failure Bring-Up Delay timer
5. Brings up the MLAG member interfaces after the timer expires
6. Notifies the peer node that the local node is healthy

Consider the following during PSU recovery:

1. During the 90-second bring-up delay, MLAG member interfaces remain operationally DOWN even though the PSU fault has cleared. The delay allows the system to complete hardware programming and synchronize forwarding state with the peer node.
2. Non-MLAG interfaces and uplink track ports recover immediately when the fault clears.
3. If the ISL or MCP peer link is down when the timer expires, MLAG member interfaces remain operationally DOWN until all dependencies are healthy.

Disk Full Detection and Fault Recovery

Multi-Chassis Link Aggregation (MLAG) resiliency protects your network against disk space exhaustion on critical partitions. When disk space on the rootfs or usrdata partition becomes critically low, MLAG automatically transitions all front-panel and MLAG member interfaces to an operationally inactive state to prevent traffic black-holing and system instability. You can configure MLAG to continuously monitor disk space utilization and provide escalating warning levels before triggering a system fault.

Disk Full Detection

The chassis manager (chassis-mgr) continuously monitors available disk space on critical partitions (rootfs and usrdata) using default monitoring intervals. The system transitions between states based on disk usage thresholds and absolute free-space limits.

When disk space on rootfs or usrdata becomes critically low, the system exhibits the following behaviors:

- Cannot write system and daemon logs
- Daemon crashes and kernel module loading failures
- CLI unresponsiveness
- Traffic disruption
- System instability or uncontrolled state

Disk Space Monitoring States

The system defines four states based on disk usage and available free space:

Table 4: Disk Space Monitoring States

State	Description	Behavior	Condition	Monitoring Interval
Normal	When the disk space usage of rootfs partition is less than 70%, the system operates in Normal state.	System is in a healthy state	Disk usage (rootfs or usrdata) < 70%	10 seconds
Warning	When the disk space usage of rootfs partition is greater than or equal to 70%, the system enters Warning state.	• A warning log is displayed to indicate the user about disk space status • User is notified to take corrective action	Disk usage (rootfs or usrdata) ≥ 70%	5 seconds

Table 4: Disk Space Monitoring States (continued)

State	Description	Behavior	Condition	Monitoring Interval
Severe	If disk space continues to decrease and rootfs partition usage exceeds 80%, the system enters Severe State	- User is notified to reduce unnecessary disk writes - High severity warning logs are generated - Immediate cleanup actions are recommended	Disk usage (rootfs or usrdata) $\geq$ 80%	2 seconds
Critical	If disk space drops below critical levels (free space in rootfs goes under 1.5GB or usrdata under 2GB), the system enters Critical State.	- System fault event is triggered - All front panel interfaces are brought down with the reason code SYSTEM_FAULT - MLAG is notified of the unhealthy state to prevent split-brain scenarios	Available free space in rootfs < 1.5GB OR Available free space in usrdata < 2GB	Immediate fault handling

System Behavior on Critical Disk Full

When a critical disk full condition is detected, all front-panel Ethernet ports (if administratively UP), port-channels, and the ISL tunnel transition to operationally DOWN with the reason code SYSTEM_FAULT:

```
device# show interface brief
Port Mtu Admin-State Oper-State Speed Ifindex Description
Eth 0/1 9216 UP DOWN (SYSTEM_FAULT) 100G 0x1000020 Ethernet 0/1
Eth 0/2 9216 UP DOWN (SYSTEM_FAULT) 100G 0x1000040 Ethernet 0/2
Eth 0/3 9216 UP DOWN (SYSTEM_FAULT) 100G 0x1000060 Ethernet 0/3
Eth 0/4 9216 UP DOWN (SYSTEM_FAULT) 100G 0x1000080 Ethernet 0/4
Po 64 9216 UP DOWN 0G 0x4000040 Port-Channel 64
Tu ISL_10.20.20.2 UP DOWN 0x3100001 Tunnel ISL_10.20.20.2
device#
```

Disk Full Recovery

When the system detects that disk space has been freed and is no longer in a critical state, the following actions occur:

- Clears the system fault condition on the monitoring service
- Brings up all non-MLAG front-panel interfaces and uplink track ports
- Starts a bring-up delay timer to ensure system stability
- Brings up the MLAG member interfaces after the timer expires
- Notifies the peer node that the local node is healthy

Users should proactively clear old files when Warning (70%) or Severe (80%) state is detected, before the system reaches Critical state.

Remove low priority files like Techsupport, log files, and core dumps from the rootfs and usrdata partition.

If rootfs size becomes less than 500 MB, k3s will start restarting services based on their usage. This in turn may lead to reboot of the device if 2 or more services get restarted.

When disk full condition is detected, automatic tech support collection is not triggered for the following reasons:

- Techsupport files will be always stored in usrdata partition (/mnt/onl/usrdata).
- If the disk is already full, there will be insufficient space to generate and store the tech support archive file.
- Attempting to collect tech support on a full disk could fail due to lack of space and worsen the condition.

To diagnose disk usage when disk full condition is reported, a log indicating the files/directories consuming the highest disk space will be displayed. Run the following commands to determine the disk usage:

1. "du -sh " → To check the size of a file or directory.
2. "df -h" → To check the overall disk usage.
3. "du -sh /* 2> /dev/null | sort -hr | head -20" → To check which directories are consuming more space in rootfs.
4. "du -sh /mnt/onl/usrdata/* 2>/dev/null | sort -hr | head -20" → To check which directories are consuming more space in usrdata partition.

Users should proactively clear old files when Warning (70%) or Severe (80%) state is detected, before the system reaches Critical state.

Show System Health CLI

Use the show system health command to verify disk full fault status. The output displays disk usage information for all monitored partitions:

```
device# show system health

BIOS Vendor: American Megatrends Inc.
BIOS Version: 3A01
```



```
BIOS Date: 07/03/2020
OS: linux
Platform: alpine
Platform Version: 3.23.3
Kernel Version: 6.6.111-yocto-standard
Kernel Arch: x86_64
CPU Procs: 8
CPU Model: Intel(R) Atom(TM) CPU C3758 @ 2.20GHz
Memory Free: 28.79GB
Memory Used: 1.77GB
Memory Used (%): 5.00
Memory Total: 31.19GB
Primary Rootfs Disk Total: 9.75GB
Primary Rootfs Disk Free: 4.30GB
Primary Rootfs Disk Used: 4.93GB
Secondary Rootfs Disk Total: 9.8GB
Secondary Rootfs Disk Used: 5.0GB
Secondary Rootfs Disk Free: 1.4GB
USRDATA Partition Total: 114GB
USRDATA Partition Free: 107GB
USRDATA Partition Used: 1.0GB
USRDATA Partition Used: 2.00%
APPDATA Partition Used: 14.00%
CONFIG Partition Used: 1.00%
IAH Partition Used: 1.00%

-----
DEGRADED-COMPONENTS
-----
Component: Disk-space
Reason: insufficient disk space on / (available: 1.4G (1491738624 bytes), required: 1.5G)
Time: 2026-08-28 09:51:41
System Fault Triggered: True
```

*Disk Full Event Log Messages***Table 5:**

Event Type	Log ID	Full Message	Condition Triggered
Disk full detection	27017	2026-05-12 16:08:40.2292 monitorsvc {"Level":"error","LogID":"27017","Msg":"Disk full condition detected: insufficient disk space on / (available: 1.38 GB, required: 1.50 GB)"} }	Detection of disk full condition
System fault (Disk failure)	27012	2026-05-12 16:08:40.2297 monitorsvc: {"Level":"error","LogID":"27012","Msg":"received system fault event: insufficient disk space on / (available: 1.38 GB, required: 1.50 GB)"} }	System fault triggered by Disk failure

Uncontrolled Restart of FWD-HAL

When the Forwarding Hardware Abstraction Layer (FWD-HAL) application microservice restarts, the Multichassis Link Aggregation Group (MLAG) service relies on health monitoring updates from the state database (SDB) that the monitoring service publishes. If the monitoring service marks a peer as unhealthy, MLAG receives this notification and promptly informs its peer using the keepalive service. When the system is healthy again, MLAG communicates this recovery to the peer to ensure that the health status of each peer remains synchronized.

When FWD-HAL terminates ungracefully, the device reloads after saving tech support information and gracefully terminating services. To minimize traffic disruption, the monitor-svc microservice detects the crash, generates a system fault, and notifies services. Also, the system disables TX on Quad Small Form-factor Pluggable (QSFP) interfaces by using a host script to bring down the front panel link. This does not affect copper ports (it disables only QSFP interfaces).

For example, the monitoring service notifies this event in the following key path:

```
key /components/component[name=degraded-component]
{
  "name": "degraded-component",
  "subcomponents": {
    "subcomponent": [
      {
        "name": "Service",
```

```

    "state": {
      "name": "Service"
    }
  }
]
}
}

key /components/component[name=Service]
{
  "name": "Service",
  "properties": {
    "property": [
      {
        "name": "reason",
        "state": {
          "configurable": false,
          "value": "service fwd-hal not healthy"
        }
      },
      {
        "name": "timestamp",
        "state": {
          "configurable": false,
          "value": "2025-02-03 07:06:27"
        }
      },
      {
        "name": "count",
        "state": {
          "configurable": false,
          "value": "0"
        }
      }
    ]
  },
  "state": {
    "name": "Service",
    "type": "health"
  }
}

```

System Fault Notification

System fault notification updates specific key paths with fault information. The **show system health** command displays the reason for degradation (service fwd-hal not healthy).

```

device# show system health

-----
DEGRADED-COMPONENTS
-----
Component: Service
Reason: service fwd-hal not healthy
Time: 2025-02-03 07:06:27

```

Uncontrolled Restart of Multiple Services

When two or more services terminate ungracefully within 10 minutes, the device reboots automatically. To aid traffic convergence and minimize disruption, Multichassis Link Aggregation Group (MLAG) generates a system fault.

I2C and EEPROM Errors on QSFP Modules

The system detects Quad Small Form-factor Pluggable (QSFP) presence, types, and capabilities via Inter-Integrated Circuit (I2C) requests to the Complex Programmable Logic Device (CPLD) via QSFP EEPROM. However, these transactions can fail due to hardware issues with the QSFP modules or the I2C signal path. In some cases, repeated errors might lock the I2C controller and cause broader failures in QSFP management.

The two types of I2C errors are EEPROM I2C and CPLD I2C errors. These errors affect QSFP management only, not the data plane or traffic flow.

EEPROM I2C Error

This error indicates a permanent hardware fault preventing communication with the port EEPROM, likely because of a failed EEPROM device or damaged I2C signal path. The port remains in an error state until you physically repair or replace it, because reloads or power cycles cannot not resolve the issue.

CPLD I2C Error

This error indicates a CPLD-managed I2C communication failure, usually because faulty QSFP modules disrupt the I2C bus. Recovery involves replacing the affected QSFP modules and reloading the device to reset the CPLD.

Solution

When the system detects an I2C error at the port, BCM APIs enforce port shutdown, which deactivates the laser and transmitter, which triggers the peer device to redirect traffic. The peer device then senses a loss-of-signal (LOS) and shifts traffic to standby links.

If this port is the sole remaining ISL member, all front-panel interfaces go operational-down state to enable traffic handover to the MLAG peer without ongoing redundancy.

If I2C errors exceed the threshold, all front-panel interfaces go operational down state, which triggers a controlled MLAG failover.

Operator Guidance

- EEPROM I2C Errors: Remove or replace the faulty QSFP module or port hardware to recover. Device reloads, power cycles, or admin state changes do not work.
- CPLD I2C Errors: Remove or replace the affected QSFP modules, then reload the device to recover. Admin state changes are not enough.

Interface Operational Down Reasons

Ports show specific reasons for being down:

- I2C_ERROR: When an Inter-Integrated Circuit (I2C) issue occurs on the port, it goes down until a device reload. Admin flaps do not bring it up.
- SYSTEM_FAULT:
 - I2C errors below threshold and Inter-Switch Link (ISL) tunnel are not impacted. Affected ports show an I2C_ERROR.

```
device# show int brief
Port      Mtu    Admin-State Oper-State      Speed  Ifindex
Description
Eth 0/1   9216   UP          DOWN (I2C_ERROR) 100G   0x1000020  Ethernet
0/1
Eth 0/2   9216   UP          UP               100G   0x1000040  Ethernet
0/2
Eth 0/3   9216   UP          UP               100G   0x1000060  Ethernet
0/3
Eth 0/4   9216   UP          DOWN (I2C_ERROR) 100G   0x1000080  Ethernet
0/4
Tu ISL_10.1.2.2      UP          UP               0x3100001  Tunnel
ISL_10.1.2.2
device#
```

- I2C errors exceed threshold and ISL tunnel is impacted. The affected ports show a SYSTEM_FAULT error.

```
device# show int brief
Port      Mtu    Admin-State Oper-State      Speed  Ifindex
Description
Eth 0/1   9216   UP          DOWN (SYSTEM_FAULT) 100G   0x1000020
Ethernet 0/1
Eth 0/2   9216   UP          DOWN (SYSTEM_FAULT) 100G   0x1000040
Ethernet 0/2
Eth 0/3   9216   UP          DOWN (SYSTEM_FAULT) 100G   0x1000060
Ethernet 0/3
Eth 0/4   9216   UP          DOWN (SYSTEM_FAULT) 100G   0x1000080
Ethernet 0/4
Tu ISL_10.1.2.2      UP          DOWN              0x3100001
Tunnel ISL_10.1.2.2
device#
```

- Out-of-memory condition: All front panel Ethernet ports (if admin UP) go down with a SYSTEM_FAULT error.

```
device# show int brief
Port      Mtu    Admin-State Oper-State      Speed  Ifindex  Description
Eth 0/1   9216   UP          DOWN (SYSTEM_FAULT) 100G   0x1000020  Ethernet 0/1
Eth 0/2   9216   UP          DOWN (SYSTEM_FAULT) 100G   0x1000040  Ethernet 0/2
Eth 0/3   9216   UP          DOWN (SYSTEM_FAULT) 100G   0x1000060  Ethernet 0/3
Eth 0/4   9216   UP          DOWN (SYSTEM_FAULT) 100G   0x1000080  Ethernet 0/4
device#
```

- PSU failure condition: When the online PSU count drops below the device minimum, all front-panel Ethernet ports (if admin UP) transition to DOWN with a SYSTEM_FAULT error.

```
device# show interface brief
Port      Mtu    Admin-State Oper-State      Speed  Ifindex  Description
Eth 0/1   9216   UP          DOWN (SYSTEM_FAULT) 100G   0x1000020  Ethernet 0/1
```

```

Eth 0/2    9216  UP          DOWN (SYSTEM_FAULT) 100G 0x1000040 Ethernet 0/2
Eth 0/3    9216  UP          DOWN (SYSTEM_FAULT) 100G 0x1000060 Ethernet 0/3
Eth 0/4    9216  UP          DOWN (SYSTEM_FAULT) 100G 0x1000080 Ethernet 0/4
Po 64      9216  UP          DOWN                0G   0x4000040 Port-Channel 64
Tu ISL_10.20.20.2  UP      DOWN                0x3100001 Tunnel
ISL_10.20.20.2
device#

```

Post-Reload System Fault Behavior

If a switch reloads after a system fault, and Inter-Integrated Circuit (I2C) errors persist, the system remains in system fault state if any port has an I2C error, regardless of whether it is an Inter-Switch Link (ISL) port or not. This only applies if the reload happens after the initial system fault detection.

Interface Manager Restart

If the interface manager restarts, ports in Inter-Integrated Circuit (I2C) error or the system in the fault state retain their state after restart.

YANG Modules and CLI Commands for MLAG Resiliency

YANG Path

The monitor-svc microservice updates the state database (SDB) with system fault information in the following paths:

1. Degraded Component: Lists all subcomponents in a degraded state: `/components/component[name=degraded-component]`
2. Subcomponent Details: Provides the reason, timestamp, and count for each subcomponent (Memory, Service and Inter-Integrated Circuit (I2C)): `/components/component[name=Memory]`

Multichassis Link Aggregation Group (MLAG) sends a notification on the subcomponent to clear the event. MLAG updates `/components/component[name=degraded-component]` with an empty subcomponent list.

Yang Path Updates for Monitor-SVC

- Memory issues: Updates degraded-component with Memory subcomponent and reason

```

key /components/component[name=degraded-component]
{
  "name": "degraded-component",
  "subcomponents": {
    "subcomponent": [
      {
        "name": "Memory",
        "state": {
          "name": "Memory"
        }
      }
    ]
  }
}

```

```

key /components/component[name=Memory]
{
  "name": "Memory",
  "properties": {
    "property": [
      {
        "name": "reason",
        "state": {
          "configurable": false,
          "value": "insuffiecient memory available (less than 953.7 MiB)"
        }
      },
      {
        "name": "timestamp",
        "state": {
          "configurable": false,
          "value": ""
        }
      },
      {
        "name": "count",
        "state": {
          "configurable": false,
          "value": "0"
        }
      }
    ]
  }
}

```

To clear the event delete notification will be sent on the subcomponent.

```

key /components/component[name=degraded-component]
{
  "name": "degraded-component",
  "subcomponents": {
    "subcomponent": []
  }
}

```

- Service restarts: Updates degraded-component with Service subcomponent and reason

```

key /components/component[name=degraded-component]
{
  "name": "degraded-component",
  "subcomponents": {
    "subcomponent": [
      {
        "name": "Service",
        "state": {
          "name": "Service"
        }
      }
    ]
  }
}

key /components/component[name=Service]
{
  "name": "Service",
  "properties": {
    "property": [
      {

```

```

    "name": "reason",
    "state": {
      "configurable": false,
      "value": "service fwd-hal not healthy"
    }
  },
  {
    "name": "timestamp",
    "state": {
      "configurable": false,
      "value": "2025-02-03 07:06:27"
    }
  },
  {
    "name": "count",
    "state": {
      "configurable": false,
      "value": "0"
    }
  }
]
},
"state": {
  "name": "Service",
  "type": "health"
}
}

```

- I2C errors:
 - Updates degraded-component with I2C subcomponent
 - Tracks count, affected interfaces, and timestamp

```

module extreme-i2c-fault {

  yang-version "1";

  namespace "http://extremenetworks.com/yang/i2c-fault";

  prefix "extr-i2c-fault";

  // import some basic types
  import openconfig-system { prefix oc-sys; }
  import openconfig-extensions { prefix oc-ext; }
  import openconfig-types { prefix oc-types; }
  import openconfig-interfaces { prefix oc-if; }

  organization
    "Extreme Networks, Inc.";

  contact
    "Extreme Networks, Inc."
    http://www.extremenetworks.com ";

  description
    "This module defines configuration and operational state data
    for I2C/EEPROM fault handling. It provides a configurable threshold
    to trigger system-wide interface shutdown when a specified number
    of interfaces experience I2C/EEPROM hardware errors, preventing
    traffic blackholing.";

  oc-ext:openconfig-version "0.1.0";

  revision "2025-11-24" {
    description

```



```

        "Initial revision for I2C fault handling feature";
        reference
            "0.1.0";
    }

    // grouping statements

    grouping i2c-fault-config {
        description
            "Configuration data for I2C fault handling";

    leaf threshold {
        type uint16 {
            range "0..80";
        }
        description
            "Number of interfaces with I2C/EEPROM faults required to
            trigger shutdown of all front-panel ports. When the number
            of faulted interfaces reaches or exceeds this threshold, the
            system will disable TX on all ports to prevent traffic
            blackholing.";
    }
    }
    grouping i2c-fault-state {
        description
            "Operational state data for I2C fault handling";

    leaf count {
        type uint16;
        description
            "Current number of interfaces experiencing I2C/EEPROM faults.
            This counter includes all interfaces where persistent I2C
            read errors have been detected, regardless of interface type.";
    }
    leaf system-fault-triggered {
        type boolean;
        description
            "Indicates whether the threshold has been exceeded and global
            interface shutdown has been triggered. Once set to true, this
            flag remains true until system reboot, even if individual
            interface faults are cleared.";
    }
    leaf timestamp {
        type oc-types:timeticks64;
        description
            "Timestamp of when the threshold was last exceeded and global
            shutdown was triggered. Format: YYYY-MM-DD HH:MM:SS";
    }
    leaf-list interfaces {
        type oc-if:base-interface-ref;
        description
            "List of interface names currently experiencing I2C/EEPROM
            faults. Format: interface names as shown in 'show interface
            brief' (e.g., 'ethernet 0/1', 'ethernet 0/5').";
    }
    }
    grouping i2c-fault-top {
        description
            "Top-level grouping for I2C fault configuration
            and state";
    }

```

```

container i2c-fault {
  description
    "Configuration and state for I2C/EEPROM fault handling.
    This feature monitors I2C/EEPROM hardware errors on interfaces
    and triggers a protective shutdown of all ports when a
    configurable threshold is exceeded, preventing traffic loss.";
  container config {
    description
      "Configuration parameters for I2C fault handling";
    uses i2c-fault-config;
  }
  container state {
    config false;
    description
      "Operational state for I2C fault handling";
    uses i2c-fault-config;
    uses i2c-fault-state;
  }
}

// augment statements

augment "/oc-sys:system" {
  description
    "Adds I2C fault configuration to the openconfig-system
    model";

  uses i2c-fault-top;
}
}

FWD-HAL will notify using following keypaths -

key /components/component[name=degraded-component]
{
  "name": "degraded-component",
  "subcomponents": {
    "subcomponent": [
      {
        "name": "I2C",
        "state": {
          "name": "I2C"
        }
      }
    ]
  }
}

/components/component[name=I2C]
  /properties/property[name=reason]/state/value = "I2C errors on 3 interfaces"
  /properties/property[name=timestamp]/state/value = "2025-02-15 10:23:45"
  /properties/property[name=count]/state/value = 3
  /properties/property[name=interfaces]/state/value = "ethernet 0/1,ethernet
0/2,ethernet 0/6"

```

- PSU failure: The chassis manager updates the degraded-component path with the PSU subcomponent and reason when online PSUs fall below the device minimum.

```
key /components/component[name=degraded-component]
{
  "name": "degraded-component",
  "subcomponents": {
    "subcomponent": [
      {
        "name": "PSU",
        "state": {
          "name": "PSU"
        }
      }
    ]
  }
}

key /components/component[name=PSU]
{
  "name": "PSU",
  "properties": {
    "property": [
      {
        "name": "reason",
        "state": {
          "configurable": false,
          "value": "insufficient number of power supply units in the device"
        }
      },
      {
        "name": "timestamp",
        "state": {
          "configurable": false,
          "value": ""
        }
      },
      {
        "name": "count",
        "state": {
          "configurable": false,
          "value": "0"
        }
      }
    ]
  }
}
```

MLAG clears the PSU fault by sending a delete notification on the subcomponent:

```
key /components/component[name=degraded-component]
{
  "name": "degraded-component",
  "subcomponents": {
    "subcomponent": []
  }
}
```

CLI Commands

For details about MLAG commands, see the *Extreme OS ONE Switching and Routing Command Reference Guide*.

show Command

When an Out of Memory (OOM), PSU fault, or PSU recovery condition occurs:

- Front panel Ethernet ports become operationally down with the reason code `SYSTEM_FAULT` (if administratively UP).
- **show int brief** displays the interface status, including OOM affected ports.
- **show system health** displays the degraded components, such as memory, service restart, I2C errors, or PSU fault with the reason for the degradation.

Enhanced show mlag peer Command

- Displays peer health state.
- A new leaf `/mlag/peers/peer[name=%v]/exception-state` updates peer exception information.

The following example output shows Peer State (UP) and Peer Exception (Peer Under Maintenance Mode, Unhealthy State):

```
device# show mlag peer

Peer towards-leaf
=====
Peer State           : UP
MCP State            : UP
Role                 : BACKUP
Elected MAC         : 02:00:22:33:44:55
Extend Bridge Count  : 46
Peer Exception       : Peer Under Maintenance Mode, Unhealthy State
device#
```

The following example output confirms that the peer no longer reports an unhealthy exception:

```
device# show mlag peer

Peer peer01
=====
Peer State           : UP
MCP State            : UP
Role                 : PRIMARY
Elected MAC         : 50:0a:9c:97:90:00
Extend Bridge Count  : 1614
Peer Exception       : None
```

RASLog Messages

- MLAG peer unhealthy: LogID:17014 MLAG Peer 10.2.5.5 is Unhealthy
- MLAG peer healthy: LogID:17014 MLAG Peer 10.2.5.5 is Healthy

MLAG Resiliency Event Log Messages

Event Type	Log ID	Full Message	Condition Triggered
MLAG peer unhealthy	17014	LogID:17014 Msg:MLAG Peer 10.2.5.5 is Unhealthy	Peer health state changes
MLAG peer healthy	17014	LogID:17014 Msg:MLAG Peer 10.2.5.5 is Healthy	Peer health state changes
PSU_FAIL BringUp Delay start	17016	Level:info LogID:17016 Msg:Timer PSU_FAIL BringUp Delay Started for 90 Seconds	MLAG starts the PSU_FAIL BringUp Delay timer after PSU recovery
PSU_FAIL BringUp Delay expiry	17017	Level:info LogID:17017 Msg:Timer PSU_FAIL BringUp Delay Expired	The PSU_FAIL BringUp Delay timer expires and the system restores interfaces
I2C CPLD error	5004	2025-12-05 04:15:57.2860 fwd-hal[99]: Level:error LogID:5004 Topic:1 Msg:I2C errors found during accessing CPLD for the port and it may shutdown, Reboot the device to recover. slot:0 port:14	CPLD I2C error detected
I2C EEPROM error	5005	2025-12-05 04:10:06.9816 fwd-hal[99]: Level:error LogID:5005 Topic:1 Msg:I2C errors found during accessing EEPROM for the port and it may shutdown, Remove and/or replace the pluggable media on this port and reboot the device to recover. port:13 slot:0	EEPROM I2C error detected
PSU minimum violation	5411	Level:error LogID:5411 Msg:Required minimum PSUs are not Up	The number of online PSUs drops below the device minimum

Event Type	Log ID	Full Message	Condition Triggered
PSU minimum restored	5412	Level:error LogID:5412 Msg:Required minimum PSUs are Up	The number of online PSUs meets or exceeds the device minimum
System fault (I2C error)	25039	2025-12-09 20:15:11.8566 interface-mgr[7]: Level:error LogID:25039 Msg:System fault detected due to I2C error	System fault triggered by I2C error

Using MLAG CLI Commands

This task uses the Multichassis Link Aggregation Group (MLAG) CLI commands to configure MLAG interfaces, keepalive behavior, and peer attributes, and then verify MLAG status and learned MAC addresses. For details on syntax and command parameters, see the *Extreme OS ONE Switching and Routing Command Reference Guide*.

1. Clear existing MLAG peer statistics so that new counters reflect only the current session.

```
device# clear counters mlag peer m12345
```

Replace *m12345* with the name of the MLAG peer.

2. Configure MLAG interfaces and their identifiers.

- a. In MLAG configuration mode, add an Ethernet interface and assign an MLAG ID.

```
device# configure terminal
device(config)# mlag
device(config-mlag)# interface ethernet 0/1 id 234
```

The ID value can be in the range 0 to 4294967295 or set to **auto**.

- b. Add one or more port channel interfaces as MLAG interfaces.

```
device(config-mlag)# interface port-channel 10 id 1020
```



Note

MLAG port channels do not support minimum links configuration. Configuring minimum links on MLAG port channels can disrupt traffic during failure events such as link down or port channel related triggers.

3. Set MLAG keepalive timers.

```
device(config-mlag)# keepalive interval 1000 delay 100 multiplier 4
```

The keepalive command supports the following parameters:

- **interval** (0 to 1000000 ms)
- **delay** (100 to 1000000 ms)
- **multiplier** (1 to 50)

Use the **no** form of this command to restore the default values.

4. Configure the global MLAG MAC address and device role.

- a. Specify the MLAG MAC address.

```
device(config-mlag)# mac a001.a002.a003
```

Use 48-bit dotted notation (xxxx.xxxx.xxxx).

- b. Set the MLAG role for the device.

```
device(config-mlag)# role primary
```

You can specify **auto**, **primary**, or **backup**. The **auto** keyword chooses the role based on the keepalive behavior.

5. Configure the MLAG peer and the extended bridge domain behavior.

- a. Enter MLAG peer configuration mode and specify a peer name.

```
device(config-mlag)# peer peer01
device(config-mlag-peer-peer01)#
```

- b. Extend a subset of bridge domains to the peer.

```
device(config-mlag-peer-peer01)# extend-bd subset 1,5,100-200
```

The range for the subset is 1 to 8192.

- c. Optionally, extend all bridge domains except those specified.

```
device(config-mlag-peer-peer01)# extend-bd except 1001,1301
```

6. Configure the MLAG peer keepalive address type, destination, source interface, and keepalive type.

- a. Create or select a keepalive instance for the peer.

```
device(config-mlag-peer-peer01)# keepalive keepalive01
device(config-mlag-peer-keepalive-keepalive01)#
```

- b. Specify the address type for the keepalive destination.

```
device(config-mlag-peer-keepalive-keepalive01)# address-type manual
```

Use **auto** to derive the management based keepalive automatically or **manual** to specify explicit addresses.

- c. Configure the keepalive destination address.

```
device(config-mlag-peer-keepalive-keepalive01)# destination 12.1.1.1
```

You can specify an IPv4 or an IPv6 destination.

- d. Select the source interface for the keepalive traffic.

```
device(config-mlag-peer-keepalive-keepalive01)# source-interface port-channel 10
```

You can specify an Ethernet, loopback, management, or port channel interface.

- e. Set the keepalive type.

```
device(config-mlag-peer-keepalive-keepalive01)# type primary
```

You can specify **primary** or **secondary**. Typically, the primary keepalive uses the ISL path, and the secondary uses the management network.

7. (Optional) Verify the MLAG peer status, interfaces, MAC learning, and counters.

- a. Display MLAG counters for the peer and keepalive sessions.

```
device(config-mlag-peer-keepalive-keepalive01)# do show counters mlag peer

Peer peer01
=====
Transitions      : 2
```

```

No. of Restarts : 2
Last clear      : 0000-00-00 00:00:00

    Keepalive keepalive01
    -----
    Transitions      : 1
    Receive           : 8 (BFD)
    Transmit          : 8 (BFD)
    Last clear        : 0000-00-00 00:00:00
device(config-mlag-peer-keepalive-keepalive01)#

```

- b. Display MAC addresses that the system learned via MLAG for a specific bridge domain.

```

device(config-mlag-peer-keepalive-keepalive01)# do show mac-address-table bridge-
domain 100 mlag

Total number of Mac Entries: 2
Hardware Status Codes - #:Failed
Mac-Address              Type              Interface
-----
00:10:94:10:01:01        MLAG              ethernet 0/1
device(config-mlag-peer-keepalive-keepalive01)#

```

- c. Check the overall MLAG configuration.

```

device(config-mlag-peer-keepalive-keepalive01)# do show mlag

Role                : PRIMARY
MAC                 : 00:10:94:10:01:01
Bring up delay      : 1000 seconds
Keepalive Delay     : 100 ms
Tx Interval         : 1000 ms
Multiplier          : 4
No. of Peers        : 1
No. of Interfaces   : 2
Uplink-track Interfaces : ethernet 0/1, port-channel 10

Peer Info:
=====
Peer                State
-----
peer01              UP

Interfaces Info:
=====
Id      Interface      Peer                Local/Remote State  Exceptions
-----
234     ethernet 0/1     peer01              UP
peer01
peer01              UP

Interfaces Info:
=====
Id      Interface      Peer                Local/Remote State  Exceptions
-----
1020    port-channel 10     peer01              UP
device(config-mlag-peer-keepalive-keepalive01)#

```

- d. Check the MLAG interface summary.

```

device(config-mlag-peer-keepalive-keepalive01)# do show mlag interfaces

Interface Identifier: 234
Interface: ethernet 0/1
Peer                Lcl/Rmt State Rmt Cfg/Act Member Up Count  Exceptions
-----
peer01              UP / UP        1 / 1          11

Interface Identifier: 1020

```



```

Interface: port-channel 10
Peer          Lcl/Rmt State Rmt Cfg/Act Member Up Count Exceptions
-----
peer01        UP / UP      2 / 2              16
device(config-mlag-peer-keepalive-keepalive01)#

```

- e. Display detailed information about MLAG peers and their keepalives.

```

device(config-mlag-peer-keepalive-keepalive01)# do show mlag peer

Peer peer01
=====
Peer State           : UP
MCP State            : UP
Role                 : PRIMARY
Elected MAC         : 50:0a:9c:97:90:00
Extend Bridge Count  : 1614
Peer Exception       : None

Keepalive keepalive01
-----
Destination Address: 12.1.1.1
State               : UP
Type                : PRIMARY
Address Type        : MANUAL
Source Interface    : port-channel 10 ( 10.1.1.0 )
First Rx            : 0000-00-00 00:00:00
Last Rx             : 0000-00-00 00:00:00
device(config-mlag-peer-keepalive-keepalive01)#

```

Configure an MLAG Session on Peer Devices

You can configure Multichassis Link Aggregation Group (MLAG) sessions on peer devices.

Follow this procedure to configure an MLAG session on peer devices.

1. Access global configuration mode.

```
device# configure terminal
```

2. Add an MLAG session.

```
device# (config)# mlag
device# (config-mlag)# peer mlag-peer
```

3. (Optional) Verify the MLAG session configuration on the peer devices.

```
device# show running-config mlag
mlag
peer mlag-peer
device# show running-state mlag
mlag
peer mlag-peer
```

4. Configure uplink track ports between MLAG peers and an external router.

```

device(config-mlag)# uplink-track interface ( ethernet IFNAME | port-channel
PONUMBER )
EX:
(device(config-mlag)# uplink-track interface
ethernet      Ethernet
port-channel  Port-channel

device(config-mlag)# uplink-track interface ethernet
IFNAME  Interface name in slot/port or slot/port:breakout format i.e slot/

```

```

port:<channel range>
device(config-mlag)# uplink-track interface ethernet 0/1:1

device(config-mlag)# uplink-track interface port-channel
    PORANGE Value from 1-255, Example: 1. Range Example: 1-3,5,7-9
device(config-mlag)# uplink-track interface port-channel 200

```

You configure uplink track ports between MLAG peers and an external router. This command helps reduce convergence for reload cases by diverting traffic to alternate paths. After reload, the device puts uplink track interfaces in the Oper down (MLAG_SHUT) state until the Bring Up delay timer expires.

A split brain scenario makes uplink track interfaces Oper down (MLAG_SHUT) on a secondary MLAG peer along with MLAG interfaces.

5. Clear the counters of an MLAG peer.

```
device# clear counters mlag peer m12345
```

6. Extend the bridge domains across the MLAG peer.

The following example configures the subset in extended bridge domain mode with the range of bridge domains. This example extends the bridge domains with IDs 1, 5, and 100 to 200 to other MLAG peers.

```

device# configure terminal
device(config)# mlag
device(config-mlag)# peer peer01
device(config-mlag-peer-peer01)# extend-bd subset 1,5,100-200

```

The following example configures all the bridge domains except the configured bridge domains.

```
device(config-mlag-peer-mlag-peer)# extend-bd except 1-3,5,7-9
```

7. Configure an MLAG interface.

The following example configures an Ethernet interface on port 1 in slot 0 and specifies 234 as the MLAG identifier.

```

device# configure terminal
device(config)# mlag
device(config-mlag)# interface ethernet 0/1 id 234

```

The following is an example command of adding port channels.

```

device(config-mlag)# interface port-channel 1 id auto
device(config-mlag)# interface port-channel 10 id 1020

```

8. Specify the delay, interval, and multiplier settings for MLAG keepalive transmissions.

The following example sets the interval to 1000 milliseconds, the delay to 100 milliseconds, and the multiplier to 4.

```

device# configure terminal
device(config)# mlag
device(config-mlag)# keepalive interval 1000 delay 100 multiplier 4

```

9. Specify the MLAG peer keepalive name for MLAG keepalive transmissions and enter MLAG peer keepalive configuration mode.

The following example configures an MLAG peer named peer01 and a keepalive named keepalive01.

```

device# configure terminal
device(config)# mlag
device(config-mlag)# peer peer01
device(config-mlag-peer-peer01)# keepalive keepalive01

```

10. Specify the IP address for the keepalive destination of the MLAG peer.

The following example configures an MLAG peer named peer01 and configures IPv4 address 12.1.1.1 as its keepalive destination.

```
device# configure terminal
device(config)# mlag
device(config-mlag)# peer peer01
device(config-mlag-peer-peer01)# keepalive keepalive01
device(config-mlag-peer-keepalive-keepalive01)# destination 12.1.1.1
```

11. Specify the MLAG MAC address.

The following example configures the MLAG MAC address on the device.

```
device# configure terminal
device(config)# mlag
device(config-mlag)# mac a001.a002.a003
```

12. Set the MLAG bringup delay.

```
device# configure terminal
device(config)# mlag
device(config-mlag)# bringup-delay 100
```

13. Specify the type of role for MLAG.

The following example sets the MLAG role as primary.

```
device# configure terminal
device(config)# mlag
device(config-mlag)# role primary
```

14. Display statistics for MLAG.

The following example displays statistics for the MLAG peer.

```
device# show counters mlag peer

Peer dut3
=====
Transitions      : 2
show counters mlag
No. of Restarts  : 2
Last clear       : 0000-00-00 00:00:00

Keepalive kp-primary
-----
Transitions : 1
Receive     : 8 (BFD)
Transmit     : 8 (BFD)
Last clear  : 0000-00-00 00:00:00

Keepalive kp-secondary
-----
Transitions : 1
Receive     : 199603
Transmit     : 199603
Last clear  : 0000-00-00 00:00:00
device#
```

15. Specify the source interface for the MLAG peer keepalive.

The following example configures an MLAG peer named peer01 and configures an Ethernet port named 0/1 as its keepalive source interface.

```
device# configure terminal
device(config)# mlag
device(config-mlag)# peer peer01
device(config-mlag-peer-peer01)# keepalive keepalive01
device(config-mlag-peer-keepalive-keepalive01)# source-interface ethernet 0/1
```

16. Specify the MLAG peer keepalive type.

The following example configures an MLAG peer named peer01 and configures it to use a primary keepalive type.

```
device# configure terminal
device(config)# mlag
device(config-mlag)# peer peer01
device(config-mlag-peer-peer01)# keepalive keepalive01
device(config-mlag-peer-keepalive-keepalive01)# type primary
```

17. Display details about the MLAG interfaces

The following example displays information about all configured MLAG interfaces on the device.

```
device(config-mlag-peer-keepalive-keepalive01)# do show mlag interfaces

Interface Identifier: 1010
  Interface: ethernet 0/13:1
Peer□   Lcl/Rmt State Rmt Cfg/Act Member Up Count Exceptions
-----
dut3           UP / UP           1 / 1           1

Interface Identifier: 4294967295
  Interface: ethernet 0/17:1
Peer□   Lcl/Rmt State Rmt Cfg/Act Member Up Count Exceptions
-----
dut3           UP / UP           1 / 1           1

Interface Identifier: 1103
  Interface: port-channel 103
Peer□   Lcl/Rmt State Rmt Cfg/Act Member Up Count Exceptions
-----
dut3           UP / UP           3 / 3           1

Interface Identifier: 2000
  Interface: port-channel 104
Peer   Lcl/Rmt State Rmt Cfg/Act Member Up Count Exceptions
-----
dut3           UP / UP           1 / 1           1
```

18. Display Bidirectional Forwarding Detection (BFD) details about the MLAG primary keepalive state on the device.

The following example displays BFD details on the configured MLAG peer on the device.

```
device(config-mlag-peer-keepalive-keepalive01)# do show mlag peer bfd

Peer mlag-peer
=====
Keepalive one
-----
Profile           : default
Session ID        : 1
Session Status    : Up
```

19. Display details about the MLAG peers that are configured on the device.

The following example displays information about the configured MLAG peer on the device.

```
device(config-mlag-peer-keepalive-keepalive01)# do show mlag peer

Peer dut4
=====
Peer State           : UP
MCP State            : UP
```

```

Role                : PRIMARY
Elected MAC        : 02:00:22:33:44:55
Extend Bridge Count : 72
Peer Exception      : None
  Keepalive kp-primary
  -----
  Destination Address: 12.1.1.1
  State              : UP
  Type              : PRIMARY
  Address Type       : MANUAL
  Source Interface   : port-channel 20 ( 11.x.x.x )
  First Rx           : 2025-06-24 15:34:32
  Last Rx            : 2025-06-24 15:34:33

```

20. Display information about the configured MLAG peer extended bridges on the device.

```

device(config-mlag-peer-keepalive-keepalive01)# do show mlag peer extended-bridges

Peer dut3
=====
No. of Extend Bridge: 34
Extend Bridges :
1,100,200-202,205,209,215-216,300-302,400,402,500-506,1001,2001,2500-2501,3001-3002,300
5-3
006,4000-4003,8192

```

21. Display information about the MLAG configuration.

The following example displays information about the MLAG configuration on the device.

```

device(config-mlag-peer-keepalive-keepalive01)# do show mlag

Role                : BACKUP
MAC                 : none
Bring up delay      : 100 seconds
Keepalive Delay     : 1000 ms
Tx Interval         : 300 ms
Multiplier          : 3
Uplink Track Intf   : □□ ethernet 0/4,0/6
                   port-channel 54,64
No. of Peers        : 1
No. of Interfaces   : 17
Peer Info:
=====
  Peer              State
  -----
  dut3              UP
Interfaces Info:
=====

```

Id	Interface	Peer	Local/Remote State	Exceptions
1010	ethernet 0/13:1	dut3	UP / UP	
4294967295	ethernet 0/17:1	dut3	UP / UP	
1103	port-channel 103	dut3	UP / UP	
2000	port-channel 104	dut3□	UP / UP	
1181	port-channel 181	dut3	UP / UP	
1182	port-channel 182	dut3□	UP / UP	
183183	port-channel 183	dut3	UP / UP	
184184	port-channel 184	dut3	UP / UP	
1185	port-channel 185	dut3	UP / UP	
186186	port-channel 186	dut3	UP / UP	
1187	port-channel 187	dut3	UP / UP	
1188	port-channel 188	dut3□	UP / UP	
189	port-channel 189	dut3	UP / UP	
1190	port-channel 190	dut3	UP / UP	

```

191919      port-channel 191      dut3      UP / UP
1192      port-channel 192      dut3      UP / UP
1193      port-channel 193      dut3      UP / UP

device(config-mlag-peer-keepalive-keepalive01)# do show mlag peer

Peer dut3
=====
Peer State           : UP
MCP State            : UP
Role                 : BACKUP
Elected MAC         : 02:00:22:33:44:55
Extend Bridge Count  : 72
Peer Exception       : None
  Keepalive kp-primary
  -----
  Destination Address : 11.1.2.1
  State               : UP
  Type                : PRIMARY
  Address Type        : MANUAL
  source Interface    : port-channel 20 ( 11.1.2.2 )
  First Rx            : 2025-05-20 19:09:49
  Last Rx             : 2025-05-20 19:42:03
  Keepalive kp-secondary
  -----
  Destination Address : 10.x.x.x
  State               : UP
  Type                : SECONDARY
  Address Type        : AUTO
  Source Interface    : management 0 ( 10.3x.x.x )
  First Rx            : 2025-05-20 19:09:51
  Last Rx             : 2025-05-20 20:13:28

```

22. Display the currently running MLAG configuration on a device.

```

device(config-mlag-peer-keepalive-keepalive01)# do show running-config mlag

mlag
  role backup
  bringup-delay 100
  uplink-track interface ethernet 0/4,0/6,0/11:2
  uplink-track interface port-channel 12,54,64
  interface port-channel 183 id 183183
  interface ethernet 0/17:1 id 4294967295
  interface port-channel 190 id auto
  interface port-channel 182 id auto
  interface port-channel 193 id auto
  interface port-channel 191 id 191919
  interface ethernet 0/13:1 id 1010
  interface port-channel 188 id auto
  interface port-channel 181 id auto
  interface port-channel 184 id 184184
  interface port-channel 189 id 189
  interface port-channel 104 id 2000
  interface port-channel 187 id auto
  interface port-channel 192 id auto
  interface port-channel 103 id auto
  interface port-channel 185 id auto
  interface port-channel 186 id 186186
  peer dut3
    extend-bd except 1001,1301
    keepalive kp-primary
      type primary
      address-type manual
      destination 11.1.2.1

```

```

source-interface port-channel 20
!
keepalive kp-secondary
type secondary
!
!
!
!

```

Monitor MLAG Events and Notifications

Multichassis Link Aggregation Group (MLAG) event monitoring and notification involves tracking the operational status, health, and specific events of an MLAG deployment to ensure network redundancy and stability. Extreme OS ONE Switching and Routing offers several ways to monitor these events and provide notifications:

- Command line interface (CLI): You can use specific commands (such as **show counters mlag peer**, **show mlag**, **show mac-address-table bridge-domain**, or **show mlag interfaces**) to check the current status and statistics on the devices directly.
- Event notifications and alarms:
 - SNMP traps: You can configure events to send SNMP notifications (traps) to a central event monitor.
 - Reliability, Availability and Serviceability Log (RASLog) logging: MLAG typically logs events in the system log file and can forward them to a remote logging server.

Supported Notifications

The device generates the following traps when the peer is up or down:

Table 6: Extreme ONE MLAG MIB

Trap Name and OID	Varbinds	Description
extremeMlagPeerDownTrap 1.3.6.1.4.1.1916.1.63.0.1	extremeMlagPeerAddrType extremeMlagPeerAddr	Generated when MLAG peer goes down extremeMlagPeerAddrType: peer IP address type extremeMlagPeerAddr: peer IP address
extremeMlagPeerUpTrap 1.3.6.1.4.1.1916.1.63.0.2	extremeMlagPeerAddrType extremeMlagPeerAddr	Generated when MLAG peer comes up

MLAG MIB Definitions

- MIB Module: Defines objects and notifications for MLAG
- Objects:
 - extremeMlagPeerAddrType: Specifies peer IP address type
 - extremeMlagPeerAddr: Specifies peer IP address

- Notifications:
 - extremeMlagPeerDownTrap: MLAG peer is down
 - extremeMlagPeerUpTrap: MLAG peer is up

The following example shows an Extreme OS ONE Switching and Routing MLAG MIB.

```
EXTREME-ONE-MLAG-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
        Integer32, Unsigned32, Counter32, Counter64
        FROM SNMPv2-SMI
        -- RFC 2578

    MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
        FROM SNMPv2-CONF
        -- RFC 2580

    InetAddress, InetAddressType
        FROM INET-ADDRESS-MIB

    sysName
        FROM SNMPv2-MIB

    extremeAgent
        FROM EXTREME-BASE-MIB;

extremeOneMlagMIB MODULE-IDENTITY
    LAST-UPDATED "202503110000Z" -- 11 March 2025 00:00:00 GMT
    ORGANIZATION "Extreme Networks, Inc."
    CONTACT-INFO
        "Postal: Extreme Networks, Inc.
         6480 Via Del Oro
         San Jose, CA 95119 USA
         Phone: +1 408 579-2800
         E-mail: support@extremenetworks.com
         WWW: http://www.extremenetworks.com"
    DESCRIPTION
        "initial version"
    ::= { extremeAgent 63 }

-- Top-level components of this MIB module.

extremeMlagNotifications OBJECT IDENTIFIER ::= { extremeOneMlagMIB 0 }

extremeMlagPeerAddrType OBJECT-TYPE
    SYNTAX InetAddressType
    MAX-ACCESS accessible-for-notify
    STATUS current
    DESCRIPTION
        "This object specifies the IP address type of the peer."
    ::= { extremeOneMlagMIB 1 }

extremeMlagPeerAddr OBJECT-TYPE
    SYNTAX InetAddress
    MAX-ACCESS accessible-for-notify
    STATUS current
    DESCRIPTION
        "This object specifies the peer IP address."
    ::= { extremeOneMlagMIB 2 }

extremeMlagPeerDownTrap NOTIFICATION-TYPE
    OBJECTS {
        extremeMlagPeerAddr, -- peer address
        extremeMlagPeerAddrType, -- peer address type
```



```

        sysName -- The local host name assigned for this switch
    }
    STATUS current
    DESCRIPTION
        "MLAG peer is down"
    ::= { extremeMlagNotifications 1 }

extremeMlagPeerUpTrap NOTIFICATION-TYPE
    OBJECTS {
        extremeMlagPeerAddrType, -- peer address type
        extremeMlagPeerAddr, -- peer address
        sysName -- The local host name assigned for this switch
    }
    STATUS current
    DESCRIPTION
        "MLAG peer is up"
    ::= { extremeMlagNotifications 2 }

END

```

YANG Module for MLAG Status Retrieval

Peer Status

- Retrieved from the YANG leaf object: /mlag/peers/peer/state/peer-state
- Module: extreme-mlag
- Peer state values:
 - UP: Peer is up
 - DOWN: Peer is down
 - UNKNOWN: Peer state is unknown

```

module: extreme-mlag
+--rw mlag
+--rw peers
|   +--rw peer* [name]
|       +--rw name          -> ../config/name
|       +--ro state
|           | +--ro name?          string
|           | +--ro peer-state?    mlag-entity-state

```

Keepalive Status

- Retrieved from the YANG leaf object: /mlag/peer/keepalives/keepalive/state/keepalive-state
- Module: extreme-mlag
- Keepalive state values:
 - UP: Keepalive is up
 - DOWN: Keepalive is down
 - UNKNOWN: Keepalive state is unknown

```

module: extreme-mlag
+--rw mlag
+--rw peers
|   +--rw peer* [name]
|       +--rw keepalives
|           +--rw keepalive* [name]
|               +--rw name          -> ../config/name

```

```

|          +--ro state
|          +--ro name?          string
|          +--ro keepalive-state? mlag-entity-state

typedef mlag-entity-state {
  type enumeration {
    enum UP {
      description
        "The entity is perceived to be up by the system.";
    }
    enum DOWN {
      description
        "The entity is perceived to be down by the system.";
    }
    enum UNKNOWN {
      description
        "The current state of the entity is not known to the system.";
    }
  }
  description
    "Type for state of MLAG entities";
}

```

Displaying MLAG Addresses in the MAC Address Table for a Bridge Domain

You can use the **show mac-address-table bridge-domain *ID* mlag** command to display the MAC addresses that the system learns from Multichassis Link Aggregation Group (MLAG) clients. This command differentiates the addresses that it learns locally and remotely from the MLAG peers.

The following example displays the MLAG addresses in the MAC address table for bridge domain 100 on the device.

```

device# show mac-address-table bridge-domain 100 mlag

Total number of Mac Entries: 2
Hardware Status Codes - #:Failed
Mac-Address          Type          Interface
-----
00:10:94:10:01:01    MLAG          ethernet 0/1:1.10
device#

```

MLAG Event Log Messages

Peer Up or Down Events

Log messages indicate MLAG peer operational state changes (UP or DOWN). For example, MLAG peer 7.7.7.2 is operationally UP or operationally DOWN.

```

{"Level":"info","Service":"mlag","LogID":17011,"Time":"2025-01-13 06:25:09.969 UTC
+0000","Msg":"MLAG Peer 7.7.7.2 is operationally UP"}
{"Level":"info","Service":"mlag","LogID":17011,"Time":"2025-01-13 06:28:49.735 UTC
+0000","Msg":"MLAG Peer 7.7.7.2 is operationally DOWN"}

```

Keepalive Up or Down Events

Log messages indicate the MLAG keepalive state changes (UP or DOWN) for primary and secondary keepalives. For example, MLAG keepalives with peer 7.7.7.2 primary:DOWN secondary:UP.

```
{"Level":"info","Service":"mlog","LogID":17009,"Time":"2025-01-13 06:28:49.224 UTC
+0000","Msg":"MLAG Keepalives with Peer 7.7.7.2 Primary:DOWN Secondary:UP"}
{"Level":"info","Service":"mlog","LogID":17009,"Time":"2025-01-13 06:28:49.735 UTC
+0000","Msg":"MLAG Keepalives with Peer 7.7.7.2 Primary:DOWN Secondary:DOWN"}
```

Peer Health Events

Log messages indicate MLAG peer health state changes (healthy or unhealthy). For example, MLAG peer 10.2.5.5 is unhealthy or 10.2.5.5 is healthy.

```
LogID:17014 Msg:MLAG Peer 10.2.5.5 is Unhealthy
LogID:17014 Msg:MLAG Peer 10.2.5.5 is Healthy
```

Maintenance Mode Events

Log messages indicate the MLAG peer maintenance mode state changes (enabled or disabled). For example, the system enables or disables maintenance mode on MLAG peer 10.38.59.158.

```
LogID:17013 Msg:Maintenance Mode is Enabled on MLAG Peer 10.38.59.158
LogID:17013 Msg:Maintenance Mode is Disabled on MLAG Peer 10.38.59.158
```

RASlogs

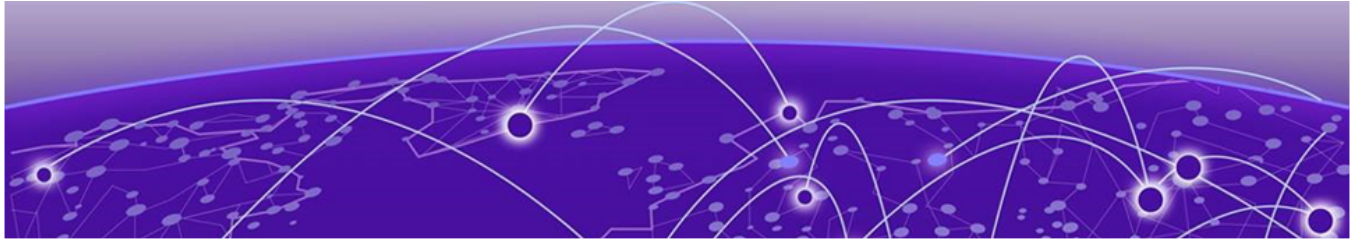
The following table describes the Reliability, Availability and Serviceability Log (RASLog) IDs.

RASlog ID	Description	Sample log	Trigger
5411	PSU minimum violation	Required minimum PSUs are not Up	The number of online PSUs drops below the device minimum
5412	PSU minimum restored	Required minimum PSUs are Up	The operator restores PSUs to meet or exceed the device minimum
17009	Mlag Keepalive UP event	MLAG Keepalives with Peer 1.1.1.1 Primary: UP Secondary: UP	MLAG starts using primary and secondary keepalives.
17009	Mlag Keepalive DOWN event	MLAG Keepalives with Peer 1.1.1.1 Primary: DOWN Secondary: DOWN	The Inter-Switch Link (ISL) interface shuts down.
17011	Mlag Peer UP event	MLAG Peer 1.1.1.1 is operationally UP	One of the keepalives comes up.
17011	Mlag Peer DOWN event	MLAG Peer 1.1.1.1 is operationally DOWN	Both keepalives go down.
17012	Mlag MLAG Control Plane Protocol (MCP) UP event	MLAG Peer 10.2.5.5 MCP State operationally UP	MCP state depends on the primary keepalive. So the trigger is the same as the primary keepalive.

RASlog ID	Description	Sample log	Trigger
17012	Mlag MCP DOWN event	MLAG Peer 10.2.5.5 MCP State operationally DOWN	MCP state depends on the primary keepalive. So the trigger is the same as the primary keepalive.
17013	Maintenance Mode Enable event	Maintenance Mode is Enabled on MLAG Peer 10.38.59.158	Enable maintenance mode on a peer MLAG device.
17013	Maintenance Mode Disable event	Maintenance Mode is Disabled on MLAG Peer 10.38.59.158	Disable maintenance mode on a peer MLAG device.
17014	Unhealthy event	MLAG Peer 10.2.5.5 is Unhealthy	Trigger a fault in the peer.
17013	Healthy event	MLAG Peer 10.2.5.5 is Healthy	Clear the fault in the peer.
17016	BringUp Delay timer start	Timer PSU_FAIL BringUp Delay Started for 90 Seconds	MLAG starts the PSU_FAIL BringUp Delay timer after the PSU fault clears
17017	BringUp Delay timer expiry	Timer PSU_FAIL BringUp Delay Expired	The PSU_FAIL BringUp Delay timer expires

**Note**

The MCP state depends on the primary keepalive state. The MCP state is UP if the primary keepalive is UP and DOWN if the primary keepalive is DOWN.



Bridge Domains

[Bridge Domain Overview](#) on page 69

[Bridge Domain Configuration](#) on page 69

Bridge Domain Overview

A bridge domain is an infrastructure that supports the implementation of different switching technologies.

A bridge domain is a generic broadcast domain that is not tied to a specific transport technology. Bridge domains support a wide range of service endpoints including regular Layer 2 endpoints and Layer 2 endpoints over Layer 3 technologies.

A bridge domain determines the flooding domain, which it shares among all of its members. This means that the bridge domain propagates the traffic that is flooded within the bridge domain to all associated members.

Bridge Domain Limitations

Extreme 8730 Platform Limitations

When double tagged traffic enters a single tagged logical interface (LIF) and exits an untagged strict LIF, the inner VLAN remains unchanged, and only the outer VLAN is removed.

Bridge Domain Configuration

Use this topic to learn about configuring a bridge domain.

By default, the **bridge-domain** command uses VLAN mode when creating a bridge domain with an ID of 1 through 4094 (mapping each one to a VLAN with the same ID) and uses default mode when creating a bridge domain with an ID of 4095 through 8192. The system displays one of the following messages to indicate this implicit behavior:

```
%Info: bridge domain IDs IDs created in vlan mode with VID VID.
```

```
%Info: bridge domain IDs IDs created in default mode.
```

To override this default (implicit) behavior and force the type of mode to use, use the **mode** keyword.

For information about commands and supported parameters to configure bridge domains, see the *Extreme OS ONE Switching and Routing Command Reference Guide*.

**Note**

When you configure a bridge domain member by using the **member** command in the CLI, it establishes the association only between the subinterface and the bridge domain. You must create the subinterfaces explicitly under the interfaces. Using the **no** form of the **member** command in the bridge domain removes only the association of the subinterface with the bridge domain. The subinterfaces themselves remain intact under the interface.

Configure a Subinterface

You can configure subinterfaces under Ethernet and port channel interfaces to define specific packet matching criteria. To manage packet flooding effectively, you must link these subinterfaces to bridge domains to enable proper packet forwarding and processing.

Use the following procedure to configure a subinterface.

1. Configure a tagged subinterface.

A tagged subinterface matches single tagged packets with a specific VLAN ID and sends them with the designated VLAN tag.

```
device# configure terminal
device(config)# interface ethernet 0/1
device(config-eth-0/1) # subinterface vlan 100
```

2. Configure an untagged strict subinterface.

**Note**

The Extreme 8730 platform does not support untagged strict subinterfaces.

An untagged strict subinterface matches only untagged packets that it receives on the port and sends packets without a VLAN tag.

```
device# configure terminal
device(config)# interface ethernet 0/2
device(config-eth-0/1) # subinterface untagged
```

3. Configure an untagged subinterface.

**Note**

The system supports this only with VLAN-mode bridge domains (not with default-mode bridge domains).

An untagged subinterface matches untagged packets, associates them with a specified VLAN, and sends them without a VLAN tag. This configuration also enables packets tagged with the VLAN ID associated with the VLAN-mode bridge domains.

```
device# configure terminal
device(config)# interface ethernet 0/3
device(config-eth-0/3) # subinterface untagged vlan 400
```

Configure a Bridge Domain in VLAN Mode

VLAN mode bridge domain: A bridge domain in VLAN mode determines VLAN switching behavior. You associate it with a specific VLAN ID (1 to 4094). This mode supports the following key features:

- Tagged and untagged configurations.
- Member ports that you add to the bridge domain create a subinterface linked to that bridge domain.

Untagged member behavior: An untagged member in a VLAN mode bridge domain accepts the following packets:

- Untagged packets
- Tagged packets with the configured VLAN ID

Use the following procedure to configure the bridge domain in VLAN mode.

1. Access global configuration mode.

```
device# configure terminal
```

2. (Implicit mode) Specify one or more bridge domains without specifying the mode explicitly. Eligible IDs are 1 to 4094. Use one or more comma separated integers or a range of integers. For example, 1 or 1,2,3 or 1-10.

```
device(config)# bridge-domain 100-101
%Info: bridge domain IDs 100-101 created in vlan mode with VID 100-101
device(config-bd-100-101)# do show running-config bridge-domain
bridge-domain 100 mode vlan
vlan-id 100
bridge-domain 101 mode vlan
vlan-id 101
!
```

3. (Explicit mode) Specify one or more bridge domains while specifying the mode explicitly. Eligible IDs are 1 to 8192. For example, 4095 or 4095,4096,4097 or 4095-5004.

```
device(config)# bridge-domain 100-101 mode vlan
device(config-bd-100-101)# do show running-config bridge-domain
bridge-domain 100 mode vlan
bridge-domain 101 mode vlan
```

4. Configure the tagged members.

You can add the physical ports or port channel as tagged member of the VLAN mode bridge domain.

```
device(config)# bridge-domain 200 mode vlan
device(config-vlan-bd-200)# vlan-id 200
device(config-vlan-bd-200)# member ethernet 0/1

device(config)# bridge-domain 100-101
%Info: bridge domain IDs 100-101 created in vlan mode with VID 100-101
device(config-vlan-bd-100-101)# member ethernet 0/1
```

5. Configure the untagged members.

You can add a physical port or port channel as an untagged member. It accepts packets with no VLAN tags and tagged traffic of the VLAN of this bridge domain. You cannot use untagged configuration in the bridge domain range mode.

```
device(config)# bridge-domain 200 mode vlan
device(config-bd-200)# vlan-id 200
device(config-bd-200)# member ethernet 0/1 untagged
device(config-bd-100)# member ethernet 0/1 untagged
```

6. Configure the untagged strict members.

You can add a physical port or port channel as an untagged strict member. It accepts packets with no VLAN tags. You must create the untagged strict subinterface and then add it to the bridge domain to get the untagged strict behavior.

```
device(config)# interface ethernet 0/2
device(config-eth-0/1)# subinterface untagged

device(config)# bridge-domain 100 mode vlan
device(config-bd-100)# vlan-id 100
device(config-bd-100)# member ethernet 0/2 untagged
```

7. Configure Virtual eXtensible LAN (VxLAN) tunnel members.

You include VxLAN tunnels in a bridge domain for extended network connectivity.

```
device(config)# bridge-domain 100 mode vlan
device(config-bd-100)# member tunnel vxlan-tunnel-1

device(config)# bridge-domain 200-210
device(config-vlan-bd-200-210)# member tunnel vxlan-tunnel-1
```

The following example displays all bridge domain configurations that are running currently on the device. This example configures bridge domain 200 explicitly in VLAN mode:

```
device# show running-config bridge-domain

bridge-domain 200 mode vlan
  description This is bridge domain 200 in VLAN mode.
  vlan-id 11
  member port-channel 190
  member port-channel 191
  member port-channel 192
  member port-channel 193
  member port-channel 194 untagged
  member port-channel 196 untagged
  member port-channel 197 untagged
  member port-channel 200 untagged
  member tunnel vxlan-tunnel-1
  arp-proxy
    suppress-arp
    arp-snooping
  !
  nd-proxy
    suppress-nd
    nd-snooping
  !
!
device#
```


Configure a Bridge Domain in Default Mode

Bridge domain default mode: In default mode, a bridge domain enables extended bridging functionality to support various subinterface configurations:

- Single-tagged
- Double-tagged
- Untagged strict

Untagged traffic handling: An untagged member in the default mode bridge domain accepts untagged traffic exclusively via an untagged strict subinterface.

Use the following procedure to configure a bridge domain in default mode.

1. Enter global configuration mode.

```
device# configure terminal
```

2. (Implicit mode) Specify one or more bridge domains without specifying the mode explicitly. Eligible IDs are 4095 to 8192. Use one or more comma-separated integers or a range of integers. For example, 4095 or 4095,4096,4097 or 4095-5004.

```
device(config)# bridge-domain 5000-5001
%Info: bridge domain IDs 5000-5001 created in default mode
device(config-def-bd-5000-5001)# do show running-config bridge-domain
bridge-domain 5000 mode default
bridge-domain 5001 mode default
```

3. (Explicit mode) Specify one or more bridge domains while specifying the mode explicitly. Eligible IDs are 1 to 8192. Use one or more comma-separated integers or a range of integers. For example, 1 or 1,2,3 or 1-10.

```
device(config)# bridge-domain 100-101 mode default
device(config-def-bd-100-101)# do show running-config bridge-domain
bridge-domain 100 mode default
bridge-domain 101 mode default
```

4. Configure bridge domain members.

```
device(config)# bridge-domain 100
device(config-bd-100)# member ethernet 0/1 vlan 100
device(config-bd-100)# member ethernet 0/1 vlan 200 inner-vlan 300
device(config-bd-100)# member ethernet 0/1 untagged
device(config-bd-100)# member tunnel vxlan-tunnel-1
```

The following example displays all bridge domain configurations that are running currently on the device. This example configures bridge domain 100 explicitly in default mode:

```
device# show running-config bridge-domain

bridge-domain 100 mode default
description This is bridge domain 100 in default mode.
member port-channel 201 vlan 1-102
member port-channel 202 vlan 1-102
member port-channel 203 vlan 1-102
member port-channel 204 vlan 1-102
member port-channel 205 untagged
member port-channel 206 untagged
member port-channel 208 untagged
member port-channel 209 untagged
member tunnel vxlan-tunnel-1
arp-proxy
```

```

suppress-arp
arp-snooping
!
nd-proxy
suppress-nd
nd-snooping
!
!
device#

```

Configure a Static MAC Address

Perform the following steps to configure static MAC address entries in a bridge domain on the device.

1. Enter the following commands to configure static MAC address entries.



Note

The **bridge-domain** command uses VLAN mode when creating a bridge domain with an ID of 1 through 4094 (mapping each one to a VLAN with the same ID) and uses default mode when creating a bridge domain with an ID of 4095 through 8192. To override this default (implicit) behavior and force the type of mode, use **bridge-domain bridge-domain-range [mode { vlan | default }]**.

```

device# configure terminal
device(config)# bridge-domain 1
%Info: bridge domain IDs 1 created in vlan mode with VID 1
device(config-bd-1)# static-mac-address 40:88:2f:f9:c0:03 ethernet 0/1
device(config-bd-1)# static-mac-address 02:e0:52:11:11:11 ethernet 0/2 untagged
device(config-bd-1)# static-mac-address f0:64:26:f5:c8:05 port-channel 2
device(config-bd-1)# static-mac-address f0:64:26:f5:c8:03 port-channel 2 untagged

```

You can configure only one static MAC address per port.

2. (Optional) Enter the following command to verify the bridge domain configuration on the device.

```

device# show running-config bridge-domain

bridge-domain 1 mode vlan
vlan-id 1
static-mac-address 02:e0:52:11:11:11 ethernet 0/2 untagged
static-mac-address 40:88:2f:f9:c0:03 ethernet 0/1
static-mac-address f0:64:26:f5:c8:03 port-channel 2 untagged
static-mac-address f0:64:26:f5:c8:04 ethernet 0/5 untagged
static-mac-address f0:64:26:f5:c8:05 port-channel 2
!

```

3. (Optional) Enter the following command to show the MAC address table for all bridge domains on a device.

```

device# show mac-address-table bridge-domain all

Bridge-Domain:1
-----
Total number of Mac Entries: 1
Hardware Status Codes - #:Failed
Mac-Address           Type           Interface
-----

```

```

00:10:00:00:00:01      Static      ethernet 0/3

Bridge-Domain:10
-----
Total number of Mac Entries: 1
Hardware Status Codes - #:Failed
Mac-Address            Type            Interface
-----
00:10:00:00:00:02      Static      ethernet 0/4.10

vm1# show mac-address-table bridge-domain 10 static
Total number of Mac Entries: 1
Hardware Status Codes - #:Failed
Mac-Address            Type            Interface
-----
00:10:00:00:00:02      Static      ethernet 0/4.10

vm1# show mac-address-table bridge-domain 10 00:10:00:00:00:02
'*' denotes best route-source
Mac-Address            Type            Interface            Last
Change                Seq No      Hardware Status
-----
*00:10:00:00:00:02      Static      ethernet 0/4.10
48s                    0

```

Configure MAC Learning

The device always enables MAC learning on the bridge domains.

1. Enter the following command to view the MAC learning information.

```

device# show mac-address-table bridge-domain all

Bridge-Domain:100
-----
Total number of Mac Entries: 6
Hardware Status Codes - #:Failed
Mac-Address            Type            Interface
-----
00:10:00:00:00:00      Dynamic      ethernet 0/1:1.100
00:10:00:00:00:06      Dynamic      ethernet 0/1:1.102
00:10:94:00:00:02      Dynamic      ethernet 0/1:1.100

```

2. Enter the following commands to clear the MAC address table.

```

device# clear mac-address-table bridge-domain ID static
device# clear mac-address-table bridge-domain ID dynamic
device# clear mac-address-table bridge-domain ID mlag
device# clear mac-address-table bridge-domain ID evpn
device# clear mac-address-table bridge-domain ID X:X:X:X:X

```

The following example shows how to clear the MAC address table.

```

device# clear mac-address-table bridge-domain 100 static
device# clear mac-address-table bridge-domain 100 dynamic
device# clear mac-address-table bridge-domain 100 mlag
device# clear mac-address-table bridge-domain 100 evpn
device# clear mac-address-table bridge-domain 100 01:01:01:01:01:01

```

3. Enter one of the following commands to display bridge domain information.

- To display information for all bridge domains on the device:


```

device# show bridge-domain all

*untag-s --accepts untagged only *untag --accepts untagged+tagged

```

```

Bridge Domain 1      Mode L2VSI_VLAN Vlan 100
Total number of member ports 2
If Name  Vlan      Inner Vlan Admin Status Oper Status
=====  =====  =====  =====  =====
Eth 0/1:1 --      --          UP          UP
Po 100   untag-s  --          UP          UP

Bridge Domain 100    Mode L2VSI_P2MP
Vni Domain base Vni 64000
Total number of member ports 1
Member ports:
If Name  Vlan      Inner Vlan Admin Status Oper Status
=====  =====  =====  =====  =====
Eth 0/7   100          UP          DOWN

```

- To display the MAC address table for a specific bridge domain:

```

device# show mac-address-table bridge-domain 100

Total number of Mac Entries: 3
Hardware Status Codes - #:Failed
Mac-Address          Type          Interface
-----
00:10:94:00:00:02    Dynamic       ethernet 0/13.100
00:10:94:00:00:10    Static        ethernet 0/13.100
00:16:3e:1e:b6:03    Local         Local

```

- To display the MAC address table for a specific bridge domain and a specific source:

```

device# show mac-address-table bridge-domain 100 dynamic

Total number of Mac Entries: 3
Hardware Status Codes - #:Failed
Mac-Address          Type          Interface
-----
00:10:94:00:00:02    Dynamic       ethernet 0/13.100

```

- To display details of an Ethernet interface or a range of interfaces (such as 1/1-2,2/1-2,3/2:1-4):

```

device# show interface ethernet 0/1 subinterface

Interface ethernet 0/1
Vlan      Inner Vlan Admin Status Oper Status If Mode
=====  =====  =====  =====  =====
untag(100) -          UP          DOWN      IF_MODE_L2
100       200       UP          DOWN      IF_MODE_L2

```

- To display details of a port channel:

```

device# show interface port-channel 103 subinterface

Vlan      Inner Vlan Admin Status Oper Status If Mode
=====  =====  =====  =====  =====
untag(1000) -          UP          UP          IF_MODE_L2
1         0         UP          UP          IF_MODE_L2
2         0         UP          UP          IF_MODE_L2

```

- To display global aging time for the MAC address table:

```

device# show system mac-address-table aging-time

Mac Aging Time: 1500

```

Configure MAC Address Aging

The device stores dynamically learned MAC addresses in the MAC address table. The MAC address aging feature flushes out the dynamic MAC addresses that are inactive for a specified period.

You can configure the aging time of dynamic MAC address entries by using the **mac-address-table aging-time** command. You can disable the MAC address aging by specifying the aging time as 0 (zero). The range is 60 to 38400 seconds. The default is 1800 seconds. The MAC aging time applies to all MAC addresses in the system.

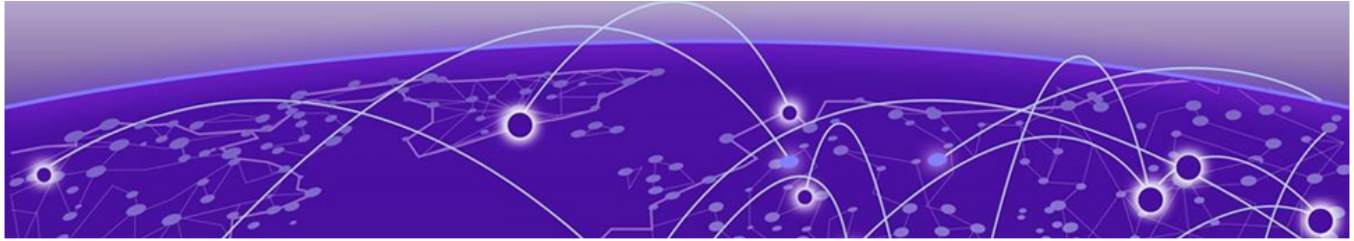


Note

The MAC address aging configuration per bridge domain is not supported.

Following is an example configuration of MAC address aging (MAC forwarding table aging time):

```
device# configure terminal
device(config)# system
device(config-system)# global
device(config-system-global)# mac-address-table aging-time 1000
device(config-system-global)
```



MAC Movement Detection and Resolution

[MAC Movement Overview](#) on page 78

[MAC Movement Detection](#) on page 79

[MAC Movement Resolution](#) on page 79

[MAC Movement Detection and Resolution Commands](#) on page 81

MAC movement detection and resolution prevents loop detection in networks. This feature fixes network malfunctions by detecting frequent MAC learnings between different logical interfaces (also called detection of MAC mobility), then detecting the network loop that causes the MAC mobility, and then shutting down the necessary ports or logging them so that you can take action manually.

Repeated MAC movement detection and resolution works for switch ports.

Loop detection using MAC movement detection is mutually exclusive of Spanning Tree Protocol (STP) and Error Limiting Device (ELD) protocol operations.

The system does not consider MAC movement detection for a port where you have enabled port security and the device triggers a restrict violation action.

MAC Movement Overview

A MAC address is defined as moved when the same MAC address arrives on a different interface in the same VLAN. Multichassis Link Aggregation Group (MLAG) enables MAC movement on local and remote nodes. However, a high MAC move rate might indicate a loop in the network or an issue with the server side interface, which causes flapping. A high rate causes the control plane to process a very high rate of MAC learning events and might exhaust the control plane resources.

MAC move definitions

- **Rapid MAC movement:** The system tracks a MAC that moves across multiple logical interfaces (LIF), ports, and VLANs for each second it moves. If the number of moves crosses the defined threshold, then the system treats it a MAC move violation.
- **Slow MAC movement:** In the first second that the system detects MAC movement, it monitors the movement. In that first second, if the number of moves does not cross the threshold, then it tracks the MAC for a maximum of 10 seconds. If after 10 seconds the total number of moves is within the threshold, then no action occurs. If it exceeds the threshold limit, then the system triggers an action.

MAC Movement Detection

This feature uses Reliability, Availability and Serviceability (RAS) traces to log MAC movement rules violations.

After you enable and configure MAC movement detection, it tracks the movement of each MAC address between all interfaces that learned them. If the number of MAC moves in one second exceeds the user defined threshold limit, then the feature records 3 values (the old logical interface (LIF), the current LIF, and the number of MAC moves).

The system parses this list of recorded values automatically. If a port has many LIFs with MAC moves, then MAC movement detection records those LIFs in the RASlog by default or shuts them down (if configured).

For example, MAC A moves between port 1, VLAN 10 and port 2, VLAN 10. MAC B moves between port 1, VLAN 20 and port 3, VLAN 20. MAC C moves between port 4, VLAN 10 and port 5, VLAN 10. All MACs have crossed the user defined threshold. The following list shows these moves:

- MAC A: port 1,VLAN 10; and port 2, VLAN 10
- MAC B: port 1, VLAN 20; and port 3, VLAN 20
- MAC C: port 4, VLAN 10; and port 5, VLAN 10

MAC Movement Resolution

Based on the previous example, the system selects port 1 for the MAC movement action (**shutdown** | **raslog**) because it has more logical interfaces (LIFs) in the list. The system either shuts down LIFs port 1,VLAN 10 and port 1, VLAN 20 with the actions logged or logs the movement details (without a shutdown).

The default MAC movement action is **raslog**. The system saves all MAC movement information to the Reliability, Availability and Serviceability Log (RASlog). When the auto recovery time limit expires, the LIF becomes operational, and the tracking process restarts.



Note

The system ignores all LIFs that are part of a tunnel, an Inter-Switch link (ISL), or a pseudowire (PW) and takes no resolution action on them.

MAC Movement Resolution enables automatic recovery by default. The LIF remains shut down for 5 minutes by default. The range is 3 through 30 minutes. You can disable automatic recovery from the command line.

You can disable shutdown from the command line. You can also enable logging as the only action when the number of MAC moves exceeds the threshold. The MAC movement details appear in the RASlog as in the following example.

```
device# 2025-06-03 21:45:18.6698 uftm[7]: Level:warn LogID:0 Topic:4 Msg:MAC-MOVE  
DETECTION: MAC 00:11:22:33:44:55/48 exceeded the number of moves between ports!  
Interface: port-channel 101.4088 Bridge Domain: 1  
2025-06-03 21:45:20.6714 uftm[7]: Level:warn LogID:0 Topic:4 Msg:MAC-MOVE DETECTION: MAC  
00:11:22:33:44:55/48 exceeded the number of moves between ports! Interface: port-channel  
101.4088 Bridge Domain: 1
```

```
2025-06-03 21:45:35.6821 uftm[7]: Level:warn LogID:0 Topic:4 Msg:MAC-MOVE DETECTION: MAC
00:11:22:33:44:55/48 exceeded the number of moves between ports! Interface: port-channel
102.409610 Bridge Domain: 1
```

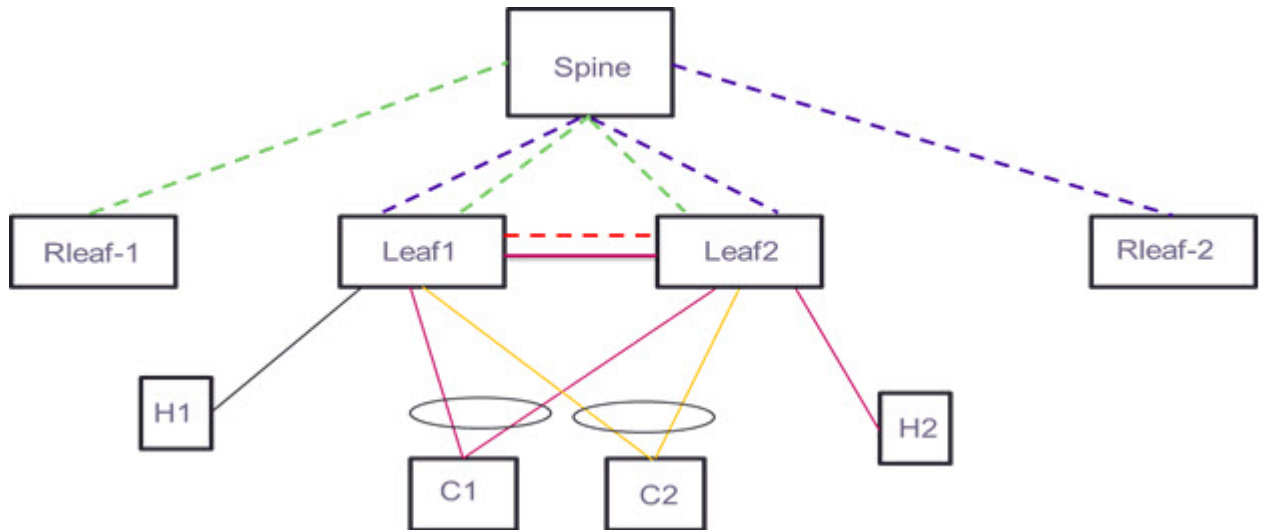


Figure 1: MAC movement detection scenario

Multichassis Link Aggregation Group (MLAG) Use Case: MAC loop between MLAG Interfaces

If MAC movement between C1 and C2 crosses the MAC move threshold, the cluster node with the higher IP address shuts down the LIF. For example: Leaf1 has the higher cluster IP configured. Leaf1 takes the decision to shut down the LIF connecting to C1 and communicates to Leaf2 to shut down the corresponding LIF connecting to C1.

MLAG Use Case: MAC loop between an MLAG Interface and a Non MLAG interface

If MAC movement between C1 and H1 crosses the threshold MAC move limit, the MLAG node with the higher IP address shuts down the LIF. For example, Leaf1 has the higher cluster IP configured. Leaf1 takes the decision to shut down the LIF on C1 and communicates to Leaf2 to shut down the corresponding LIF on C1. If the MAC movement occurs between the MLAG Interface and Non-MLAG interface, the system always takes the action on the MLAG Interface.

MLAG Use Case: MAC loop between Non MLAG interfaces

If MAC movement between H1 and H2 crosses the threshold MAC move limit, each leaf independently takes the action. Leaf1 and Leaf2 run MAC move detection separately.



Note

In an IP fabric configuration, you must enable this feature on all leaf nodes.



Note

This feature supports only bridge domains.

:

MAC Movement Detection and Resolution Commands

The following commands support MAC movement detection and resolution. You run these commands under Global configuration (config) mode.

Table 7: MAC movement detection and resolution commands

Command	Details
12 mac-move-detection enable	Enables MAC movement detection. MAC movement detection and resolution enables this feature by default. Use the no form of this command to disable the feature.
12 mac-move-detection threshold <i>moves</i>	Event count (maximum number of moves) before the system shuts down the port or triggers a Reliability, Availability and Serviceability log (RASlog). Use an integer from 3 to 500. The default is 20.
12 mac-move-detection action shutdown	Shuts down the port or interface when MAC movement exceeds the specified threshold. This is in addition to a RASlog entry (which always occurs). MAC movement detection and resolution enables shutdown by default when you use the 12 mac-move-detection enable command. Use the no form of this command to disable shutdown. Details about every shutdown appear in the RASlog.
12 mac-move-detection auto-recovery-time <i>minutes</i>	Time (in seconds) until MAC movement detection and resolution resets the port state to up. Use an integer from 0 to 1800. 0 disables auto-recovery. The default is 300 (5 minutes). Details about every recovery appear in the RASlog.
12 mac-move-detection interval <i>seconds</i>	Interval (in seconds) before the system deletes the MAC movement events. The default is 10 seconds. The range is 10 through 120 seconds.
clear 12 mac-move-detection shut-list [interface { ethernet <i>interface-name</i> port-channel <i>port</i> ve <i>interface-name</i> loopback <i>interface-name</i> }]	Clears entries from the shutdown list and sets the port state to up for all ports in a list or for a specified port.

Table 7: MAC movement detection and resolution commands (continued)

Command	Details
show bridge-domain all	Shows when the logical interface (LIF) is down because of MAC movement as in the following example: <pre> device# show bridge-domain 105 *untag-s --accepts untagged only *untag --accepts untagged+tagged Bridge Domain 105 Mode L2VSI_VLAN Vlan 105 Total_number of member ports 3 If Name Vlan Inner Vlan Admin Status Oper Status ===== ----- Tu ISL_10.20.20.7 - - UP UP Po 101 -- DOWN - UP (MAC MOVE_SHUT) Po 201 -- UP - UP UP device# show bridge-domain 115 *untag-s --accepts untagged only *untag --accepts untagged+tagged Bridge Domain 115 Mode L2VSI_VLAN Vlan 115 Total_number of member ports 3 If Name Vlan Inner Vlan Admin Status Oper Status ===== ----- Po 102 -- DOWN - UP (MAC MOVE_SHUT) Po 202 -- UP - UP UP Tu ISL_10.20.20.7 - - UP UP device# </pre>
show l2 mac-move-detection	Displays the MAC move settings as in the following example: <pre> device# show l2 mac-move-detection MAC-move detection is enabled and configured: Tracking interval: 17 seconds Threshold for triggering action: 10 moves Action taken when threshold is reached: shutdown Auto-recovery is done after 60 seconds. device# </pre>

Table 7: MAC movement detection and resolution commands (continued)

Command	Details
show l2 mac-move-detection shut-list	Displays the MAC move detection port shutdown list for a device as in the following example: <pre> device# show l2 mac-move-detection shutlist MAC-move detection shut port list Interface BD ID Time remaining (s) ----- ethernet 0/1:3.100 500 2 ethernet 0/1:2.100 500 34 device# </pre>
show interface ethernet <i>interface-name</i> subinterface	Displays when the LIF or port subinterface is down because of MAC movement.

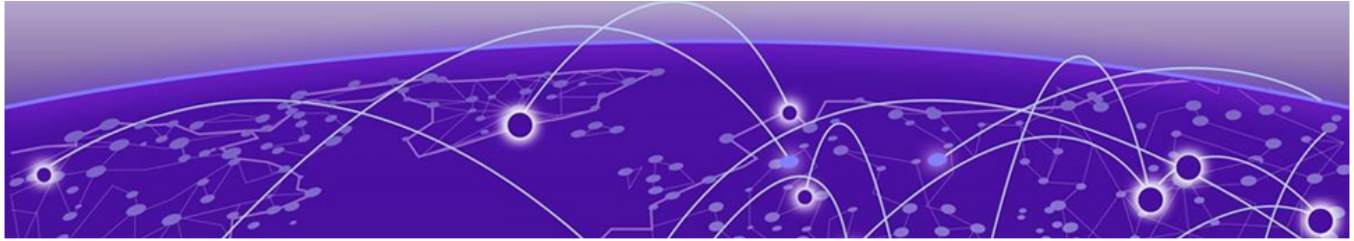
Table 7: MAC movement detection and resolution commands (continued)

Command	Details
system internal service uftm command mac-move-detection [args <i>arguments]</i>	Displays the internal state of MAC move detection. It lists the following details: • The MAC move globals • The current window index that captures MAC moves • The event map, which is the list of pairs between LIFs and the number of moves per second for each bridge domain-MAC pair • The detailed shut list, which comprises the output of the show 12 mac-move-detection shut-list command plus the the LIF information For example: <pre>device# system internal service uftm command mac-move-detection MAC-move detection debug information Dumping macMove globals: enabled = true interval = 60 threshold = 18 action = shutdown shutInterval = 65 Current macMoveData.windowIndex = 52 Dumping macMoveData.eventMap: BD: 1, MAC: 00:a9:09:11:0a:01/48, moves: 2 = {0,[]},{0,[]},{2, [0x20000036,0x3100002,]}, {0,[]},{0, []},{0,[]},{0,[]},{0,[]}, Dumping macMoveData.shutPortList: Interface LIF BD ID Time remaining (s) ----- ----- ethernet 0/1:1.30 0x200002c 1 63 device#</pre>
show system internal cdb path /system/global/12/mac- move-detection/config	Displays the internal state database (SDB) records or the timestamp, user, and IP address of the last update for the /system/global/12/mac-move-detection/config data path.

Table 7: MAC movement detection and resolution commands (continued)

Command	Details
show system internal sdb path /system/global/l2/mac- move-detection/state	Displays the internal configuration database (CDB) records or the timestamp, user, and IP address of the last update for the /system/global/l2/mac-move-detection/state data path.
show running-config system global	Displays whether the MAC movement detection feature is enabled.
system global l2 mac-move-detection enable l2 mac-move-detection action shutdown	Shuts down the port or interface when MAC movement exceeds the specified threshold. This is in addition to a RASlog entry (which always occurs). The system enables this by default when you use l2 mac-move-detection enable . Use the no form of this command to disable shutdown. Details about every shutdown appear in the RASlog.

For details about these commands, see the *Extreme OS ONE Switching and Routing Command Reference Guide*.



Multiple Spanning Tree Protocol

- [MSTP Limitations](#) on page 86
- [MSTP Terminology](#) on page 87
- [CIST Root Bridge Election](#) on page 88
- [MSTP Port Roles and Priority Vector](#) on page 89
- [MSTP Default Path Cost](#) on page 91
- [MSTP Port States](#) on page 92
- [MSTP Timers](#) on page 93
- [MSTP Configuration](#) on page 94
- [YANG Modules and gNMI Operational State Paths for MSTP](#) on page 96
- [Supported Show Commands](#) on page 97

Multiple Spanning Tree Protocol (MSTP) prevents Layer 2 loops by building one or more loop-free logical topologies over a common physical infrastructure. MSTP extends Rapid Spanning Tree Protocol (RSTP), allowing multiple spanning tree instances to run simultaneously, each governing a subset of VLANs. This enables load balancing across redundant links and limits the scope of topology reconvergence when a failure occurs.

Extreme OS ONE supports the Common and Internal Spanning Tree (CIST) domain of MSTP. The CIST runs as a single end-to-end spanning tree across all MST regions in the network, preventing loops across all VLANs on MSTP-enabled ports.

MSTP Limitations

Extreme OS ONE Multiple Spanning Tree Protocol (MSTP) supports the Common and Internal Spanning Tree (CIST) domain only. Extreme OS ONE does not support the following features and behaviors:

- MST region configuration and VLAN-to-Multiple Spanning Tree Instance (MSTI) mapping. The single CIST instance governs all VLANs on an MSTP-enabled port.
- MSTI domains. The switch supports only Instance 0 (CIST).
- Spanning Tree Protocol (STP) on Multi-Chassis Link Aggregation (MLAG) interfaces.
- Boundary port behavior details beyond CIST-only interaction with STP and Rapid STP (RSTP) peers.
- Bridge Protocol Data Unit (BPDU) Guard, Admin Edge Port (PortFast), Auto Edge Detection, and BPDU Filter.
- Root Guard, Loop Guard, and Edge Guard.
- Spanning Tree across virtual extensible LAN (VXLAN) or Ethernet VPN (EVPN) overlays.

- Per-VLAN STP (PVST) and Rapid PVST.
- Default-mode bridge domain support.
- MST Priority Vector cannot be configured or modified through this interface.
- MST timer parameters including Hello Time, Max Age, Forward Delay, and Max Hops cannot be configured or modified through this interface.

MSTP supports only the following physical port types: Ethernet (Eth) and Port-Channel (Po).

CLI Configuration

The current release supports only two CLI configurations:

- Enable the MSTP.
- Add an interface to MSTP.

Convergence

MSTP convergence time depends on the failure scenario:

- Normal convergence: Typically less than 1 second to a few seconds for direct link failures.
- Worst-case convergence: Typically around 6 seconds due to timer handling and synchronization delays.

Uncontrolled Restart

An uncontrolled restart of the MSTP process, such as an unexpected or unplanned process or system restart without proper shutdown procedures, can potentially cause a temporary loss of Spanning-Tree state, service disruption, or inconsistent forwarding until reconvergence.

MSTP Terminology

Bridge ID	A combination of bridge priority and MAC address that uniquely identifies a bridge in the MSTP domain. MSTP prefers lower values.
BPDU	Bridge Protocol Data Unit. The control frame that MSTP bridges exchange to elect the root bridge and compute port roles and states.
CIST	Common and Internal Spanning Tree. The spanning tree that connects all MST regions and legacy STP/RSTP domains into a single loop-free topology.
IST	Internal Spanning Tree. Instance 0 within an MST region; connects to the external STP/RSTP domain to form the CIST.

MST Configuration Identifier	A three-part identifier — Name, Revision, and Digest — that determines MST region membership.
MSTP	Multiple Spanning Tree Protocol. An IEEE 802.1Q standard that allows multiple spanning tree instances over a shared physical topology.
Port State	The logical forwarding state of a port: Discarding, Learning, or Forwarding.
Priority Vector	An ordered set of fields MSTP uses to compare path quality and elect port roles. MSTP prefers lower values.
Regional Root	The bridge within an MST region that provides the best path to the CIST Root.
Root Bridge	The bridge with the lowest CIST Bridge Identifier. All other bridges compute their port roles relative to the root bridge.
STG	Spanning Tree Group. The hardware forwarding abstraction used to implement spanning tree instances on supported ASICs.

CIST Root Bridge Election

Extreme OS ONE Multiple Spanning Tree Protocol (MSTP) elects a single Common and Internal Spanning Tree (CIST) root bridge for the entire spanning tree domain. The root bridge is the bridge with the lowest CIST Bridge Identifier, which is the combination of bridge priority and MAC address.

MSTP elects the root bridge using the following rules:

1. On startup, each bridge assumes it is the root bridge and advertises its own bridge identifier as the CIST Root Identifier in its BPDUs.
2. When a bridge receives a BPDU advertising a lower CIST Root Identifier or a lower total path cost to the same root, it accepts that BPDU as superior. The bridge updates its stored CIST root information and recalculates port roles and states.
3. If a bridge stops receiving superior BPDUs on all ports — because Max Age expires with no refresh — it re-elects itself as root based on its own bridge identifier.

To influence which bridge becomes the root, configure a lower bridge priority on the intended root. Bridge priority values must be multiples of 4096. The default bridge priority is 32768.



Note

A bridge with a lower priority value wins the root bridge election. Configure a priority of 61440 on a non-preferred bridge to prevent it from becoming root unintentionally.

MSTP Port Roles and Priority Vector

Understanding how Extreme OS ONE Multiple Spanning Tree Protocol (MSTP) selects the active topology requires two related concepts: the priority vector, which carries path and bridge information in each BPDU, and the port role, which MSTP assigns to each port based on that information.

MSTP Port Roles

MSTP assigns a role to each port based on received BPDU information and calculated priority vectors. The port role determines whether a port forwards data traffic and participates in the active spanning tree topology.

MSTP assigns one of the following roles to each bridge port based on the priority vector comparison for the CIST.

MSTP Port Roles and Forwarding Behavior

Role	Description	Port state
Root	Provides the best path to the CIST Root. Forwards data traffic and BPDUs toward the root. Exists as exactly one per non-root bridge.	Forwarding
Designated	Serves as the best port for a given LAN segment. Forwards BPDUs and data traffic toward the segment. Exists as exactly one per segment.	Forwarding
Alternate	Offers a backup path toward the root. Remains in Discarding under normal conditions. Transitions to Root if the active root path fails.	Discarding
Backup	Provides a redundant path to the same LAN segment on the same bridge. Remains in Discarding. Activates only when the designated port on that segment fails.	Discarding

Role	Description	Port state
Master	Carries inter-region CIST traffic toward the external CIST Root on region boundary bridges. Functions as the CIST Root Port for the region. Applies only to MSTP boundary bridges.	Forwarding
Disabled	Indicates the port is administratively down or not participating in spanning tree. Blocks all BPDUs and user data. Remains in Discarding permanently.	Discarding

Priority Vector

MSTP determines port roles by comparing priority vectors — ordered tuples that encode the quality of a path to the CIST Root. The algorithm compares each field in sequence and selects the vector whose first differing field contains the lower value.

Each BPDU carries a priority vector composed of the following fields, listed from most significant to least significant in the comparison:

MSTP Priority Vector Fields and Comparison Order

Field	Description
Root Bridge ID	Identifies the current CIST Root bridge (priority + MAC). Determines which bridge assumes the root role — the lowest value wins.
External Root Path Cost	Represents the cumulative cost to reach the CIST Root across region boundaries.
Regional Root ID	Identifies the best bridge within the local MST region for forwarding CIST traffic toward the root.
Internal Root Path Cost	Represents the cumulative cost from this bridge to the Regional Root within the region.
Designated Bridge ID	Identifies the bridge advertising this priority vector on the LAN segment.

Field	Description
Designated Port ID	Identifies the specific port on the designated bridge transmitting this information.
Receiving Port ID	Identifies the local port that received this BPDU. Serves as the final tie-breaker when all other fields are equal.

MSTP Default Path Cost

Extreme OS ONE Multiple Spanning Tree Protocol (MSTP) uses per-link path costs to express link quality. Faster links always have a lower cost, so the tree naturally prefers higher-bandwidth paths when all other priority vector fields are equal.

The Root Path Cost, in the priority vector, is the sum of link costs along the path from the bridge to the Common and Internal Spanning Tree (CIST) Root. Longer or slower paths produce higher total cost and lose to shorter or faster ones.

The following table provides the path cost values.

Table 8: Default path costs by link speed

Link speed	Default path cost
10 Mbps	2,000,000
100 Mbps	200,00
1 Gbps	20,000
10 Gbps	2,000
25 Gbps	800
40 Gbps	500
100 Gbps	200

MSTP Port States

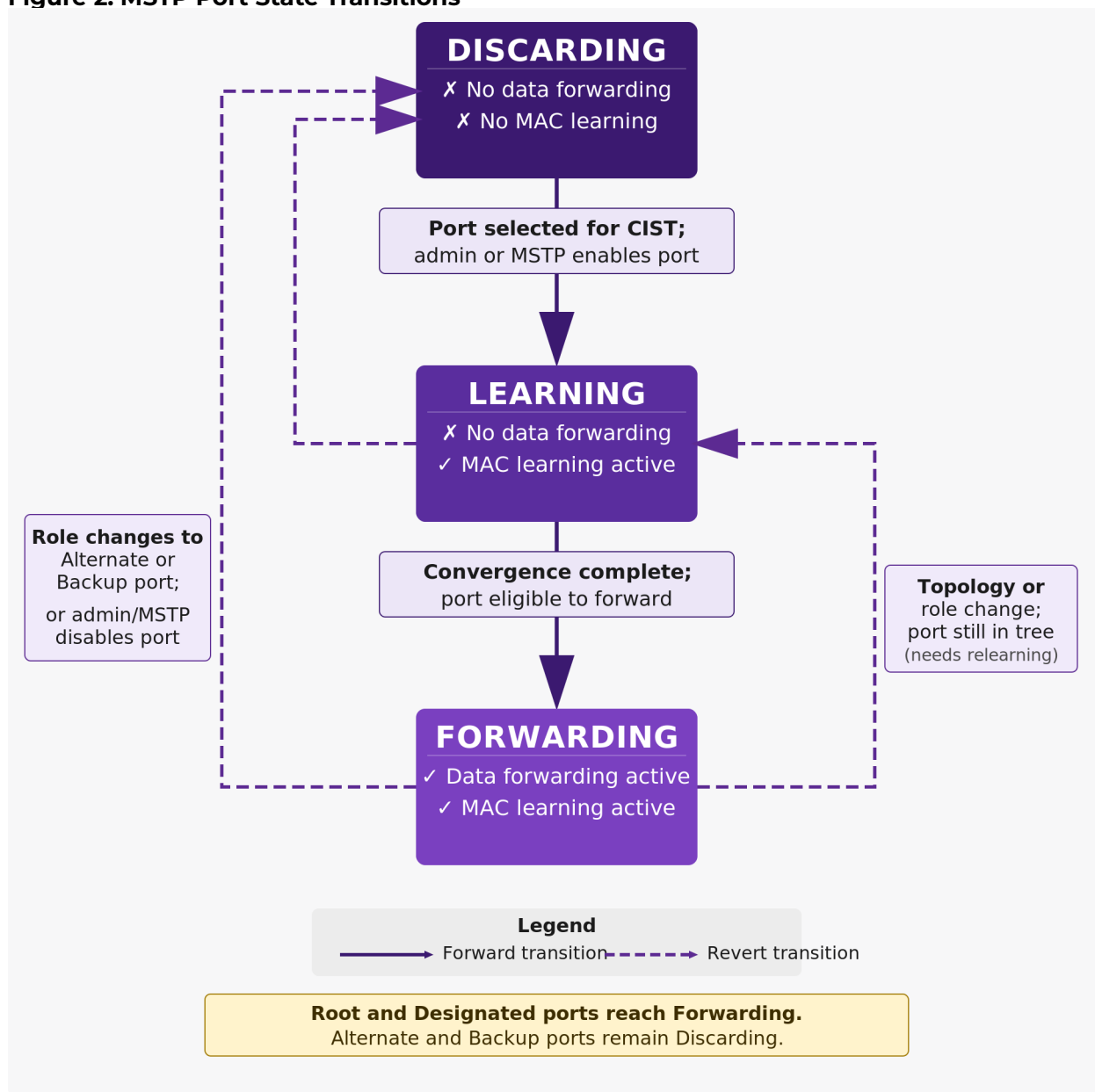
Each Extreme OS ONE Multiple Spanning Tree Protocol (MSTP) port operates in one of three states that control whether the port forwards user data and learns MAC addresses. The following table describes each state and its behavior.

Table 9: MSTP Port States — Data Forwarding and MAC Learning Behavior

State	Data forwarding	MAC Address Learning	Description
Discarding	No	No	Blocks all user data frames and suppresses MAC address learning. Processes BPDUs only. Serves as the initial state for all ports.
Learning	No	Yes	Learns source MAC addresses from received frames but blocks user data forwarding. Prepares the forwarding database to reduce flooding when the port transitions to Forwarding.
Forwarding	Yes	Yes	Forwards user data frames and continues learning MAC addresses. Operates as the fully active state for Root ports and Designated ports.

MSTP transitions ports between states as the topology converges or changes. The following diagram shows the allowed transitions and the events that trigger each one.

Figure 2: MSTP Port State Transitions



Note

Only Root Ports and Designated Ports transition to the Forwarding state. Alternate and Backup ports remain in Discarding under normal operation.

MSTP Timers

Extreme OS ONE Multiple Spanning Tree Protocol (MSTP) uses a set of protocol timers to control convergence speed, BPDU aging, and state transition delays. The CIST Root bridge advertises these timer values in its BPDUs, and all bridges in the network

adopt them to ensure consistent behavior. The following table describes each timer, its default value, and its role in the protocol.

Table 10: MSTP Protocol Timers — Defaults and Convergence Behavior

Timer	Default	Description
Hello Timer	2 s	Specifies the interval between periodic CIST BPDU transmissions on Designated ports.
Max Age	20 s	Defines how long a bridge retains received CIST information before discarding it. Expires when no fresh BPDU arrives and Message Age exceeds this value.
Forward Delay	15 s	Configures the legacy STP delay per state transition. Serves as a fallback in MSTP/RSTP, where the proposal/agreement handshake largely replaces this timer. Remains carried in BPDUs for interoperability.
Max Hops	20 s	Limits BPDU propagation within an MST region by acting as a hop-count TTL. Decrements by one at each bridge and discards the BPDU upon reaching zero.
Tx Hold Count	6 s	Restricts the number of BPDUs a port transmits per second. Prevents BPDU bursts during rapid topology changes.

MSTP Configuration

Use these topics to configure Multiple Spanning Tree Protocol (MSTP).

For information about commands and supported parameters to configure MSTP, see the *Extreme OS ONE Switching and Routing Command Reference Guide*.

Enable MSTP

Use the following procedure to enable Multiple Spanning Tree Protocol (MSTP) on the device.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter STP configuration mode.

```
device(config)# stp
device(config-stp)#
```

3. Enable MSTP.

```
device(config-stp)#enabled-protocol mstp
```

4. Verify the MSTP configuration.

```
device# show stp mstp
```

The following example enables MSTP.

```
device# configure terminal
device(config)# stp
device(config-stp)#enabled-protocol mstp
```

The following example displays the MSTP configuration.

Add an Interface to MSTP

Use the following procedure to add a physical Ethernet interface or Port-Channel interface to Multiple Spanning Tree Protocol (MSTP). You must enable MSTP globally before you add interfaces.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter STP configuration mode.

```
device(config)# stp
device(config-stp)#
```

3. Add the interface to MSTP.

```
device(config-stp)#interface {ethernet <name> | port-channel <port-channel-range>}
```

4. Verify the MSTP interface configuration.

```
device# show stp mstp interface {all | ethernet <name> | port-channel <port-channel-range>}
```

The following example adds Ethernet 0/1 to the MSTP configuration:

```
device# configure terminal
device(config)# stp
device(config-stp)#interface ethernet 0/1
```

The following example displays the MSTP configuration for Ethernet interface 0/1.

YANG Modules and gNMI Operational State Paths for MSTP

YANG Modules

MSTP configuration and operational data are accessible using the augmented `app_tree` YANG tree under the `/stp/mstp` path. The following tree shows the available CIST state nodes. All leaves are read-only.

```

/stp/mstp/cist
├── state (ro)                                     ← CIST bridge-level state
│   ├── mst-id?                                   uint16
│   ├── vlan*                                     union
│   ├── bridge-priority?                         (8192/4096/...)
│   ├── bridge-address?                         mac-address
│   ├── designated-root-priority?
│   ├── designated-root-address?
│   ├── root-port?                               uint16
│   ├── root-cost?                               uint32
│   ├── hold-time?                               uint8
│   ├── topology-changes?                       counter64
│   └── last-topology-change? timeticks64
└── interfaces (ro)                               ← CIST per-port
    ├── interface* [name]
    │   └── state
    │       ├── cost?                             uint32
    │       ├── port-priority?
    │       ├── port-num?                         uint16
    │       ├── role?                             identityref
    │       ├── port-state?                       identityref
    │       ├── designated-root-priority/address
    │       ├── designated-bridge-priority/address
    │       ├── designated-port-priority/num
    │       ├── forward-transitions? counter64
    │       └── counters
    │           ├── bpdu-sent?                     counter64
    │           └── bpdu-received?                 counter64

```

gNMI Operational State Paths

The following table provides the gRPC Network Management Interface (gNMI) state paths that you can use to query MSTP port role and port state programmatically.

Table 11: gNMI state paths

gNMI path	Description	Type
<code>/stp/mstp/ports/port-id/state/role</code>	Current port role: Root, Designated, Alternate, Backup, Master, or Disabled.	State
<code>/stp/mstp/ports/port-id/state/state</code>	Current port state: Discarding, Learning, or Forwarding.	State

Supported Show Commands

- **show stp mstp**: Displays MSTP CIST bridge-level information and per-interface spanning tree state.

```
device# show stp mstp

Spanning Tree: Enabled
Protocol: MSTP

CIST Bridge Parameters:
  CIST Root ID       : 32768 (8000) / d8:c4:97:80:f0:ac
  CIST Regional Root ID: 32768 (8000) / d8:c4:97:80:f0:ac
  CIST Bridge ID     : 36864 (9000) / a4:bb:6d:01:23:45
  Root Port         : ethernet 0/1
  Root Path Cost    : 2000
  Hello Time        : 2 s
  Max Age           : 20 s
  Forward Delay     : 15 s
  Max Hops          : 20
  Hold Time         : 1 s
  Topology Changes  : 1
  Last Topology Change : Wed Jul 1 10:05:33 2026

Interface          Role      State      Cost      Prio  Designated Bridge
BPDUs Tx    BPDUs Rx
-----
ethernet 0/1      Root      Forwarding 2000      128    8000.d8c49780f0ac
620             1540
ethernet 0/2      Designated Forwarding 2000      128    9000.a4bb6d012345
1538            618
ethernet 0/3      Alternate Discarding 2000      128    8000.d8c49780f0ac      0
```

- **show stp mstp interfaces**: Displays detailed per-interface MSTP CIST spanning tree state, including port role, port state, path cost, priority, designated bridge information, and BPDU counters:

```
device# show stp mstp interfaces

Interface          Role      State      Cost      Prio  Port Num
-----
ethernet 0/1      Root      Forwarding 2000      128    1
ethernet 0/2      Designated Forwarding 2000      128    2
ethernet 0/3      Alternate Discarding 2000      128    3
port-channel 1    Designated Forwarding 2000      128    65

Interface          Designated Root          Designated Bridge
Designated Port
-----
ethernet 0/1      8000.d8c49780f0ac          8000.d8c49780f0ac      128.1
ethernet 0/2      8000.d8c49780f0ac          9000.a4bb6d012345      128.2
ethernet 0/3      8000.d8c49780f0ac          8000.d8c49780f0ac      128.5
port-channel 1    8000.d8c49780f0ac          9000.a4bb6d012345      128.65

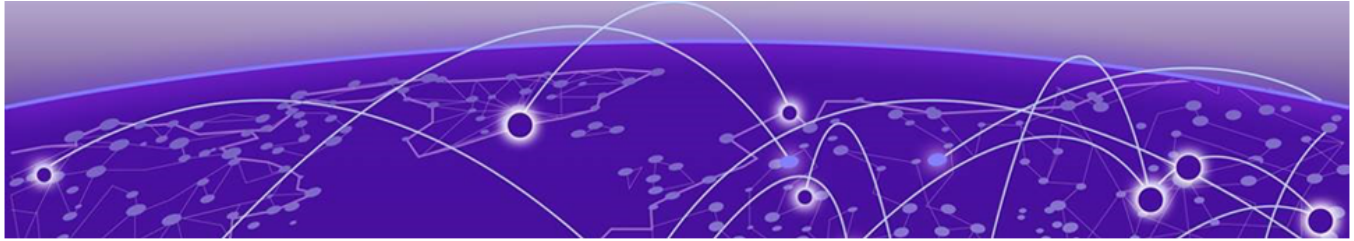
Interface          BPDUs Sent  BPDUs Received  Forward Transitions
-----
ethernet 0/1      620         1540            2
ethernet 0/2      1538        618             1
ethernet 0/3      0           1535            0
port-channel 1    1538        765             1
```

- **show stp mstp interface all:** Display per-port CIST info (cost, priority, role, state, designated bridge/port, BPDU counters)

```

device(config)# do show stp mstp interface all
Interface           : Ethernet 0/1:2
MST Instance        : 0
Port role           : Designated
Port state          : forwarding
Port priority       : 128
Port cost           : 2000000
Port type           : -
Admin edge port     : no
Oper edge port      : no
BPDU guard          : disabled
BPDU filter         : disabled
Root guard          : disabled
Loop guard          : disabled
PortFast/PVST       : disabled
Root ID             : 50:0a:9c:97:38:46
Root cost           : 2000000
Bridge ID           : 50:0a:9c:97:38:46
Designated ID       : 50:0a:9c:97:38:46
Interface           : Ethernet 0/9:1
MST Instance        : 0
Port role           : Designated
Port state          : forwarding
Port priority       : 128
Port cost           : 2000000
Port type           : -
Admin edge port     : no
Oper edge port      : no
BPDU guard          : disabled
BPDU filter         : disabled
Root guard          : disabled
Loop guard          : disabled
PortFast/PVST       : disabled
Root ID             : 50:0a:9c:97:38:25
Root cost           : 2000000
Bridge ID           : 50:0a:9c:97:38:25
Designated ID       : 50:0a:9c:97:38:25
Interface           : Ethernet 0/13:1
MST Instance        : 0
Port role           : Designated
Port state          : forwarding
Port priority       : 128
Port cost           : 2000000
Port type           : -
Admin edge port     : no
Oper edge port      : no
BPDU guard          : disabled
BPDU filter         : disabled
Root guard          : disabled
Loop guard          : disabled
PortFast/PVST       : disabled
Root ID             : 50:0a:9c:97:38:25
Root cost           : 2000000
Bridge ID           : 50:0a:9c:97:38:25
Designated ID       : 50:0a:9c:97:38:25

```



DHCP Relay

[DHCP Relay Agent Overview](#) on page 99
[DHCP Relay Features and Capabilities](#) on page 100
[DHCP Relay in MLAG Deployments](#) on page 101
[DHCP Relay gNMI Operations](#) on page 103
[DHCP Relay Limitations and Scalability](#) on page 106

DHCP Relay Agent Overview

The DHCP Relay Agent enables forwarding of DHCP messages between clients and servers across different IP subnets.

Purpose of DHCP Relay

DHCP clients send broadcast messages requesting IP address configuration. These broadcasts are limited to the local subnet and cannot propagate beyond router boundaries. The DHCP Relay Agent solves this limitation by intercepting client broadcast requests on local subnets and forwarding them as unicast packets to remote DHCP servers.

DHCP Relay Protocol Basics

The DHCP Relay Agent operates transparently in the DHCP message flow:

- **Client-to-Server Direction:** The relay receives DHCP broadcast requests (DISCOVER, REQUEST), modifies the giaddr field with the relay interface IP address, and forwards as unicast to configured servers
- **Server-to-Client Direction:** The relay receives unicast replies (OFFER, ACK, NAK), processes agent information if configured, and forwards to the originating client subnet
- The relay increments the DHCP hops field for each relayed request. Requests exceeding the maximum supported hop count are discarded.

Key Concepts

giaddr (Gateway IP Address)

The IP address of the relay interface through which the request is being forwarded. The server uses this address to select the appropriate address pool and routes its reply to this address.

Helper Address (Server Address)

The IP address of the DHCP server to which relay requests are forwarded.

Option 82 (Agent Information Option)

An optional DHCP extension that allows the relay to insert circuit identification and remote identification information.

DHCP Relay Features and Capabilities

Extreme OS ONE DHCP Relay forwards DHCP messages between clients and DHCP servers located on different IP subnets.

Feature Highlights

- **Global and Interface-Level Configuration:** Configure DHCP Relay at the global and interface levels.
- **Option 82 Support:** Agent Information Option insertion and removal compliant with RFC 3046, Option 82 supports Circuit ID and Remote ID.
- **Per-Interface Enablement:** Enable DHCP relay independently on each Layer 3 interface.
- **Multiple Helper Addresses:** Configure up to 4 helper addresses per interface.

Interface and Routing Support

- DHCP Relay is enabled on Layer 3 router interfaces.

Helper Address Support

- Configuration of multiple DHCP helper addresses (servers) per Layer 3 router interface
- Relay forwarding of client packets to all configured helper addresses
- Maximum of 4 helper addresses supported per interface

Gateway Address (giaddr) Handling

- Automatic population of the giaddr field with the IP address of the ingress relay-enabled Layer 3 interface
- Subnet identification by DHCP servers using relay gateway information
- Support for correct address pool selection on DHCP servers based on giaddr value

MLAG Redundancy Support

- Peer-local-physical-giaddr pattern for deterministic server reply routing
- MLAG-aware binding management for split DHCP transactions
- Suppression of relay on peer-link packets to prevent duplicate discovery at servers
- Single-peer failure tolerance during DHCP transactions
- Support for high-availability virtual IP (VIP) in Option 3 (Router Option)

DHCP Relay in MLAG Deployments

This section describes DHCP Relay operation in Multi-Chassis Link Aggregation (MLAG) deployments.

MLAG Topology Context

In an MLAG deployment, a DHCP client is dual-homed to two peer switches (MLAG-A and MLAG-B) via a Link Aggregation Group (LAG). The client's network interface card (NIC) uses a hash algorithm to select which LAG member carries each frame; only one of the two MLAG peers receives the broadcast at any given time.

The DHCP server returns its unicast reply to the IP address specified in the giaddr field. Therefore, the choice of what to place in giaddr is the single most important design decision for MLAG DHCP relay deployment, because it determines:

- Which MLAG peer the server reply is delivered to
- Whether relay bindings need to be synchronized between peers
- Whether the client's default gateway is highly available

Peer-Local-Physical-giaddr + Option-3-VIP Pattern

Extreme OS ONE uses a peer-local-physical-giaddr + Option-3-VIP pattern.:

Field	Purpose	Consumer
giaddr	Selects the server's address pool and carries the destination address for the server's reply	DHCP Server (pool selection + reply routing)
Option 3 (Router Option)	Tells the client which IP to install as its default route	DHCP Client (post-bind default gateway)

The design has the following characteristics:

- The relay sets giaddr to its own physical SVI IP on the client VLAN
- The DHCP server is configured to return Option 3 with the MLAG cluster's high-availability virtual IP (VIP)
- Deterministic server reply routing to the originating peer
- No cross-peer state synchronization required
- DHCP service continuity during a single MLAG peer failure

Relay Behavior in MLAG Context

The DHCP Relay Agent implements the following behavior in MLAG environments:

- **Local giaddr:** Set giaddr to the local peer's physical SVI IP on the client VLAN. This makes server replies route back deterministically to the originating peer.
- **Local Socket Binding:** Bind the server-side listening socket to the peer's physical SVI IP. The existing socket-creation path is per-giaddr, so this works naturally.
- **Peer-Link Suppression:** Suppress relay of packets received on the peer-link (CCEP / ICL). A client broadcast that traverses MLAG-A → CCEP → MLAG-B must not be re-relayed by MLAG-B, preventing duplicate DISCOVER messages at the server.
- **Single-Peer Failure Tolerance:** Tolerate single-peer failure mid-DORA. If MLAG-A fails after relaying a DISCOVER, the client's re-transmit will hash to MLAG-B. MLAG-B relays with its own giaddr and reply lands on it. RFC-compliant DHCP servers accept the re-transmit as either the same transaction (same XID) or a fresh one.

Expected DORA Event Flow in MLAG

The hash that a dual-homed client's NIC uses to select between its LAG members is independent of the relay direction. DISCOVER and REQUEST often land on different MLAG peers within the same DORA exchange — this is normal and supported. Because giaddr is peer-local, each server reply deterministically returns to the peer that originated the corresponding request. Bindings on each peer are local and self-consistent for the half of the transaction that peer handled.

Example DORA Sequence:

1. DISCOVER (broadcast) hashes to MLAG-A
2. MLAG-A relays it with giaddr = 10.1.1.2 (MLAG-A's physical IP), creating a local binding (INFLIGHT)
3. Server replies with OFFER unicast to 10.1.1.2 — lands on MLAG-A
4. MLAG-A looks up its binding (HIT), strips Option 82, forwards OFFER to client. OFFER carries Option 3 = 10.1.1.1 (VIP)
5. REQUEST (broadcast) hashes to MLAG-B (different LAG-member hash)
6. MLAG-B relays it with giaddr = 10.1.1.3 (MLAG-B's physical IP), creating its own local binding (INFLIGHT)
7. Server replies with ACK unicast to 10.1.1.3 — lands on MLAG-B
8. MLAG-B looks up its binding (HIT), strips Option 82, forwards ACK to client. ACK still carries Option 3 = 10.1.1.1 (VIP)
9. Client installs IP = 10.1.1.50, default gateway = 10.1.1.1 (VIP). DORA complete across two peers

There is no cross-peer state requirement. Each reply always lands on the peer that has the matching binding.

MLAG Failure Scenarios

Failure Scenario	Expected Behavior
MLAG-A down after DISCOVER, before OFFER	OFFER unicast to 10.1.1.2 is undeliverable (peer down). Client retransmits DISCOVER; LACP hashes to MLAG-B. MLAG-B relays fresh DISCOVER with giaddr = 10.1.1.3; DORA completes via MLAG-B. Client sees a few seconds of retransmit delay.
MLAG-A down mid-DORA, between OFFER and REQUEST	Client already has OFFER. Sends REQUEST; LACP hashes to MLAG-B. MLAG-B relays REQUEST with its own giaddr; server ACKs to MLAG-B. New transaction at the server but same XID—RFC-compliant servers accept.
CCEP / ICL link down (split brain)	Each peer continues to relay independently. Server-side may see DORAs from both peer IPs for the same client; standard server behavior is to honor whichever completes first. Slight log noise; no client-side impact.
CCP down and keep-alive down (peer fully isolated)	MLAG convention takes one peer's Layer 3 interfaces down. Surviving peer carries 100% of relay traffic.
Peer reload	Bindings on the reloading peer are lost. No service interruption—client retransmits hit the surviving peer, which builds its own binding.
DHCP server unreachable from one peer but reachable from the other	LACP hashing determines which peer's server-reachability matters per transaction. Effectively, the dual-homed client gets relay coverage as long as either peer has a working server path.

Binding Consistency and Synchronization

A key advantage of the peer-local-physical-giaddr pattern is that no cross-peer binding synchronization is required:

- Each peer maintains its own local binding table for transactions it handles
- Server replies always route to the originating peer based on giaddr
- Bindings are self-contained and inherently consistent within each peer
- If a peer fails mid-transaction, the client's retransmit naturally routes to the surviving peer via LACP hashing
- The surviving peer builds a new binding for the retransmitted request

DHCP Relay gNMI Operations

Reference for gNMI (gRPC Network Management Interface) operations used to configure and retrieve DHCP Relay state. These operations follow the OpenConfig relay-agent YANG model.

gNMI Overview for DHCP Relay

gNMI operations for DHCP Relay use YANG paths based on the OpenConfig relay-agent model.

SET Operations (Configuration)

Use SET operations to configure DHCP Relay parameters.

Operation	Description	Example
Enable Global Relay	Enable DHCP relay agent at the system level	<pre>gnmic -a 10.32.100.188:443 \ -u admin -p password --skip- verify \ set --update \ "/relay-agent/ dhcp/config/\ enable-relay- agent:::bool:::true"</pre>
Enable Global Option 82	Enable Agent Information Option globally	<pre>gnmic -a 10.32.100.188:443 \ -u admin -p password --skip- verify \ set --update \ "/relay-agent/ dhcp/agent- information-\ option/config/ enable:::bool:::true"</pre>
Enable Interface Relay	Enable DHCP relay on a specific interface	<pre>gnmic -a 10.32.100.188:443 \ -u admin -p password --skip- verify \ set --update \ "/relay-agent/ dhcp/interfaces/\ interface[id=etherne t 0/8]/config/\ enable:::bool:::true"</pre>
Enable Interface Option 82	Enable Option 82 on a specific interface	<pre>gnmic -a 10.32.100.188:443 \ -u admin -p password --skip- verify \ set --update \ "/relay-agent/ dhcp/interfaces/\ interface[id=etherne t 0/8]/agent-\ information-option/ config/\ enable:::bool:::true"</pre>

Operation	Description	Example
Configure Circuit ID	Set the circuit-id sub-option for Option 82	<pre>gnmic -a 10.32.100.188:443 \ -u admin -p password --skip-verify \ set --update \ "/relay-agent/dhcp/interfaces/\ interface[id=ethernet 0/8]/agent-\ information-option/config/\ circuit-id::string::eth0/8"</pre>
Configure Remote ID	Set the remote-id sub-option for Option 82	<pre>gnmic -a 10.32.100.188:443 \ -u admin -p password --skip-verify \ set --update \ "/relay-agent/dhcp/interfaces/\ interface[id=ethernet 0/8]/agent-\ information-option/config/\ remote-id::string::vtos"</pre>

GET Operations (Retrieval)

Use GET operations to retrieve configuration and operational state information.

Operation	Example Command
Get All Relay-Agent Configuration	<pre>gnmic -a 10.32.100.188:443 -u admin \ -p password --skip-verify get \ --path "/relay-agent/"</pre>

YANG Path Reference

Key YANG paths used in DHCP Relay gNMI operations:

/relay-agent/dhcp/config/enable-relay-agent

Global relay-agent enable flag (boolean)

/relay-agent/dhcp/agent-information-option/config/enable

Global Option 82 enable flag (boolean)

/relay-agent/dhcp/interfaces/interface[id=<name>]/config/enable

Per-interface relay-agent enable flag (boolean)

```
/relay-agent/dhcp/interfaces/interface[id=<name>]/agent-information-option/config/enable
```

Per-interface Option 82 enable flag (boolean)

```
/relay-agent/dhcp/interfaces/interface[id=<name>]/agent-information-option/config/circuit-id
```

Circuit ID string for Option 82 sub-option (string)

```
/relay-agent/dhcp/interfaces/interface[id=<name>]/agent-information-option/config/remote-id
```

Remote ID string for Option 82 sub-option (string)

Related Links

DHCP Relay Limitations and Scalability

This reference describes known limitations, unsupported features, and scalability characteristics of DHCP Relay functionality in Extreme OS ONE.

Unsupported Features

The following DHCP Relay functionality is not supported.

- DHCPv6 Relay
- Vendor-specific Option 82 sub-options
- DHCP server functionality

Operational Limitations

- **Binding State on Restart:** DHCP relay binding tables are lost upon device restart. Clients must re-transmit DHCP requests following a reboot.
- **In-Flight Transaction Handling:** DHCP transactions in progress at restart time are interrupted. Clients follow standard DHCP back-off timers and re-transmit automatically.
- **Active Lease Unaffected:** Clients already in BOUND state with active leases are not affected by relay restart — the relay agent is not consulted during normal lease use.
- **No Cross-Peer Binding Sync:** In MLAG deployments, binding state is not synchronized between peers. Each peer maintains local binding tables only.
- **MLAG Binding Loss on Peer Failure:** If a peer fails during a DHCP transaction, bindings on that peer are lost. The client's re-transmit is handled by the surviving peer, which creates a new binding.

Scalability Limits

DHCP Relay supports the limits shown in the following table.

Table 12: DHCP Relay Scalability Limits

Feature	Limit
Helper addresses per interface	4
Relay bindings	1024
VLAN interfaces with DHCP Relay enabled	16

Supported Platforms and Interfaces

Supported Platforms: DHCP Relay is supported on all Extreme OS ONE Layer 3 router interfaces, including physical interfaces, VLAN interfaces, and sub-interfaces configured with IPv4 addresses.

Option 82 Limitations

- **Standard Format Only:** Only RFC 3046 standard Option 82 format is supported.
- **Sub-Options Supported:** Circuit-ID and Remote-ID sub-options are supported.
- **Removal on Server Reply:** Option 82 is automatically removed from server replies before forwarding to clients (standard relay behavior).

MLAG-Specific Limitations

- **No Binding Replication:** Bindings are not replicated between MLAG peers. Each peer maintains its own binding table.
- **Split-Brain Behavior:** In case of ICL (Inter-Chassis Link) failure creating split-brain, each peer continues relaying independently. DHCP server may see duplicate requests.
- **Peer Failure Recovery:** Clients experiencing a peer failure mid-DORA may experience a few seconds of delay while re transmitting and re-establishing on the surviving peer.

Known Issues

Refer to the Release Notes for current known issues, limitations, and workarounds.