

Extreme NetIron Network Packet Broker Configuration Guide, 06.1.00

Supporting NetIron OS 06.1.00

© 2018, Extreme Networks, Inc. All Rights Reserved.

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 are the property of their respective owners. For additional information on Extreme Networks Trademarks please see www.extremenetworks.com/company/legal/trademarks. Specifications and product availability are subject to change without notice.

Contents

Preface	5
Document conventions.....	5
Notes, cautions, and warnings.....	5
Text formatting conventions.....	5
Command syntax conventions.....	6
Extreme resources.....	6
Document feedback.....	6
Contacting Extreme Technical Support.....	7
About This Document	9
Supported hardware and software.....	9
What's new in this document.....	9
Network Packet Broker	11
Network Packet Broker overview.....	11
Network Packet Broker enhancements.....	11
Limitations.....	12
Header processing.....	12
Configuration considerations.....	13
802.1BR header stripping.....	13
802.1BR header bypass.....	14
VN-tag header stripping.....	15
VN-tag header bypass.....	16
Show packet encap processing commands.....	17
VXLAN header stripping.....	25
NVGRE header stripping.....	26
GTP de-encapsulation configuration.....	28
GTP de-encapsulation.....	28
Configuring GTP de-encapsulation.....	29
show gtp-de-encapsulation slot.....	30
show gtp-de-encapsulation.....	31
show gtp-de-encapsulation interface.....	32
VN-tag and 802.1BR preserve enhancement.....	33
802.1BR/VN-tag preserve use case example.....	34
IP payload length based filtering using ACL.....	36
show ip match-payload-len.....	39
show ip match-payload-len interface ethernet.....	40
show ipv6 match-payload-len.....	41
show ipv6 match-payload-len interface ethernet.....	42
Source port labeling.....	43
SCTP port based traffic filtering using ACL.....	44
Packet timestamping.....	45

Preface

- Document conventions..... 5
- Extreme resources..... 6
- Document feedback..... 6
- Contacting Extreme Technical Support..... 7

Document conventions

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

Notes, cautions, and warnings

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

NOTE

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

ATTENTION

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



CAUTION

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



DANGER

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

Text formatting conventions

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

Format	Description
bold text	Identifies command names. Identifies keywords and operands. Identifies the names of GUI elements.
<i>italic text</i>	Identifies text to enter in the GUI. Identifies emphasis. Identifies variables.
Courier font	Identifies document titles. Identifies CLI output.

Format	Description
	Identifies command syntax examples.

Command syntax conventions

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

Convention	Description
bold text	Identifies command names, keywords, and command options.
<i>italic text</i>	Identifies a variable.
[]	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, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, <i>member[member...]</i> .
\	Indicates a "soft" line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Extreme resources

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

White papers, data sheets, and the most recent versions of Extreme software and hardware manuals are available at www.extremenetworks.com. Product documentation for all supported releases is available to registered users at www.extremenetworks.com/support/documentation.

Document feedback

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

You can provide feedback in two ways:

- Use our short online feedback form at <http://www.extremenetworks.com/documentation-feedback-pdf/>
- Email us at internalinfodev@extremenetworks.com

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

Contacting Extreme Technical Support

As an Extreme customer, you can contact Extreme Technical Support using one of the following methods: 24x7 online or by telephone. OEM customers should contact their OEM/solution provider.

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

- [GTAC \(Global Technical Assistance Center\)](#) for immediate support
 - Phone: 1-800-998-2408 (toll-free in U.S. and Canada) or +1 408-579-2826. For the support phone number in your country, visit: www.extremenetworks.com/support/contact.
 - Email: support@extremenetworks.com. To expedite your message, enter the product name or model number in the subject line.
- [GTAC Knowledge](#) - Get on-demand and tested resolutions from the GTAC Knowledgebase, or create a help case if you need more guidance.
- [The Hub](#) - A forum for Extreme customers to connect with one another, get questions answered, share ideas and feedback, and get problems solved. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC.
- [Support Portal](#) - Manage cases, downloads, service contracts, product licensing, and training and certifications.

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

- Your Extreme Networks service contract number and/or serial numbers for all involved Extreme Networks products
- A description of the failure
- A description of any action(s) 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

About This Document

- Supported hardware and software..... 9
- What's new in this document..... 9

Supported hardware and software

The hardware platforms in the following table are currently supported for NPB.

TABLE 1 Supported devices

ExtremeRouting XMR Series	ExtremeRouting MLX Series
ExtremeRouting XMR 4000	ExtremeRouting MLXe-4
ExtremeRouting XMR 8000	ExtremeRouting MLXe-8
ExtremeRouting XMR 16000	ExtremeRouting MLXe-16
ExtremeRouting XMR 32000	ExtremeRouting MLXe-32

What's new in this document

This is a rebranding release.

On October 30, 2017, Extreme Networks, Inc. acquired the data center networking business from Brocade Communications Systems, Inc. This document has been updated to remove or replace references to Brocade Communications, Inc. with Extreme Networks, Inc., as appropriate.

Network Packet Broker

- Network Packet Broker overview..... 11
- Header processing..... 12
- VXLAN header stripping..... 25
- NVGRE header stripping..... 26
- GTP de-encapsulation configuration..... 28
- VN-tag and 802.1BR preserve enhancement..... 33
- IP payload length based filtering using ACL..... 36
- Source port labeling..... 43
- SCTP port based traffic filtering using ACL..... 44
- Packet timestamping..... 45

Network Packet Broker overview

Beginning with NetIron 6.0.00a two FPGA bundles will be available for download.

- Installing the Network Packet Broker (NPB) FPGA bundle will place the Extreme MLXe device chassis into Packet Broker mode.
- Installing the MAIN (default) FPGA bundle will place the Extreme MLXe device chassis into the default mode.

The global setting across the chassis can be either Network Packet Broker (NPB) mode or MAIN (default).

- The Main (default) global setting requires the MAIN FPGA manifest to be installed.
- The NPB global setting requires the NPB FPGA manifest to be installed.

Network Packet Broker enhancements

Starting in the R06.0.00a release, some Network Packet Broker (NPB) features are enabled only on the NPB FPGA. If you are using any of the following features in NPB deployments on the following line cards, please ensure that you are using the correct NetIron 6.0.00a NPB FPGA files. All the other NPB features are enabled on all line cards and on both the Main and NPB FGAs.

TABLE 2 Network Packet Broker compatibility

MLXe module	NPB FPGA	Main FPGA
BR-MLX-10Gx20	<ul style="list-style-type: none"> • Packet Timestamping • NVGRE stripping • Source port labeling • VXLAN stripping • GTP de-encapsulation 	The following NPB features are not present: <ul style="list-style-type: none"> • Packet Timestamping • NVGRE stripping • Source port labeling • VXLAN stripping • GTP de-encapsulation
BR-MLX-40Gx4	Not applicable.	There is only one FPGA available for the 40G card. All NPB features are supported in the Main FPGA for this card.
BR-MLX-100Gx2	<ul style="list-style-type: none"> • Packet Timestamping • NVGRE stripping • Source port labeling • VXLAN stripping • GTP de-encapsulation 	The following NPB features are not present: <ul style="list-style-type: none"> • Packet Timestamping • NVGRE stripping • Source port labeling • VXLAN stripping • GTP de-encapsulation

The following Network Packet Broker features are available:

- **802.1BR and VN-Tag stripping:** This feature strips 802.1br header (ether-type=0x893f) and VN-tag header (ether-type=0x8926) from ingress traffic before sending it for further processing/forwarding. This is useful in cases where the analytics tools do not understand these headers.
- **Packet Timestamping:** This feature allows inserting an 8-byte timestamp into ingress packets. The timestamp can be NTP time or local clock time.
- **SCTP traffic filtering:** This feature enables the user to filter SCTP traffic based on source and destination TCP/UDP ports.
- **Source port labeling:** Users can enable this feature to insert a 4-byte label to identify the ingress port. This source port label will hold the SNMP IfIndex value from IFMIB for the interface. Source port is used for downstream filtering.
- **NVGRE stripping:** The NVGRE header-stripping feature enables the user to strip the outer Ethernet, Outer IPv4, and the NVGRE header from incoming IPv4 NVGRE packets. This is useful in cases where the analytics tools do not understand these headers, or if the tool is only interested in the tunneled information.
- **Packet Length filtering:** This feature allows users to filter ingress IPv4 and IPv6 traffic based on IP Payload Length of packets. For IPv4, payload length excludes IP header length. For IPv6, there is already a Payload Length field present in the header.
- **802.1BR bypass:** This feature enables the system to bypass the 802.1BR header from ingress traffic to comply with analytic tool requirements.
- **VN-Tag bypass:** This feature enables the system to bypass the VN-Tag header from ingress traffic to comply with analytic tool requirements.
- **GTP de-encapsulation:** This feature enables the user to de-encapsulate GTP packets.
- **VXLAN stripping:** This feature allows the user to strip the Virtual Extensible LAN (VXLAN) header from the incoming VXLAN packets.

Limitations

Network Packet Broker features have the following limitations.

802.1BR and VN-tag header processing have the following limitations.

- If the ingress port is on a 24x10 module, it is recommended to use a catch all Layer 2 Policy Based Routing (L2 PBR) to forward that traffic to a service port for VNTAG and 802.1BR header removal, followed by L2 and L3 PBR on the service port.
- Other ingress modules (8X10G etc) can separate the 802.1BR and VNTAG traffic to the service port using L2 PBR, and conduct L2/L3 PBR matching on the remaining traffic.
- 802.1BR header stripping and VN-tag header stripping features are supported in BR-MLX-40Gx4, BR-MLX-10Gx20, and BR-MLX-100Gx2-CFP2 modules.

Header processing

Header processing can be managed using the features discussed in this section. Once the headers are processed, the traffic can be forwarded to be processed by other analytic tools.

Header processing includes:

- 802.1BR header processing
- VN-tag header processing
- GTP header processing

NOTE

Before using the header stripping functionality, switch to **config-pkt-encap-proc** command line interface (CLI) mode.

The 802.1BR stripping feature performs the following on 802.1BR traffic.

- Identify 802.1BR traffic.
- Strip 802.1BR tags.
- Forward stripped packets to the next processing port for further filtering and forwarding.

The 802.1BR header bypass feature performs the following on 802.1BR traffic.

- Identify 802.1BR traffic.
- Bypass 802.1BR header and perform inner header lookup.
- Forward stripped packets to the next processing port for further filtering and forwarding.

The VN-tag header stripping feature performs the following on VN-tag traffic.

- Identify VN-tag traffic.
- Strip VN-tags.
- Forward stripped packets to the next processing port for further filtering and forwarding.

The VN-tag header bypass feature performs the following on VN-tag traffic.

- Identify VN-tag traffic.
- Bypass VN header and perform inner header lookup.
- Forward stripped packets to the next processing port for further filtering and forwarding.

The GTP header stripping feature performs the following on GTP traffic.

- Identify GTP traffic.
- GTP de-encapsulation occurs: the outer IP, the outer UDP header, and the GTP header are removed from GTP-U packets.
- Forward GTP de-encapsulated packets to the next processing port for further filtering and forwarding.

Configuration considerations

- Before using the header strip or bypass functionality, switch to **config-pkt-encap-proc** command line interface (CLI) mode.
- A PPCR in service card will only serve either the Bypass function or the Strip function for both 802.1BR/VN-tag. It cannot serve Bypass for 802.1BR and Strip for VN-tag or vice-versa.

802.1BR header stripping

The feature enables the system to strip the 802.1BR header from ingress traffic to comply with analytic tools that do not understand the 802.1BR header.

As part of this feature, the system will identify packets with an 802.1BR header, strip the header, and send the packet to next processing port.

NOTE

The 802.1BR header stripping feature is available only on the following line-cards.

- BR-MLX-40Gx4
- BR-MLX-10Gx20
- BR-MLX-100Gx2-CFP2

The syntax of the **strip-802-1br** command is discussed in the following examples.

Syntax: **[no] strip-802-1br all | slot slot-num | slot slot-num device device-id**

NOTE

To use the 802.1BR header stripping functionality, switch to **config-pkt-encap-proc** command line interface (CLI) mode.

Configuring 802.1BR header stripping on all modules

The **strip-802-1br all** command enables the 802.1BR header stripping feature on all the cards that support this feature.

```
device(sw)# config terminal
device(config)# packet-encap-processing
device(config-pkt-encap-proc)# strip-802-1br all
```

Syntax: **[no] strip-802-1br all**

Configuring 802.1BR header stripping on a specific module

The **strip-802-1br slot** command enables the 802.1BR header stripping feature on a specific card that supports this feature.

NOTE

Slot-num represents the module ID.

```
device(sw)# config terminal
device(config)# packet-encap-processing
device(config-pkt-encap-proc)# strip-802-1br slot 3
```

Syntax: **[no] strip-802-1br slot slot-num**

The following example enables the 802.1BR header stripping feature on a specific device (identified using a device-ID) of a card, that supports this feature.

NOTE

Slot-num represents the module ID and device-id represents the np-id.

```
device(sw)# config terminal
device(config)# packet-encap-processing
device(config-pkt-encap-proc)# strip-802-1br slot 2 device-id 1
```

Syntax: **[no] strip-802-1br slot slot-num device device-id**

802.1BR header bypass

The feature enables the system to bypass the 802.1BR header from ingress traffic to comply with analytic tool requirements.

As part of this feature, the system will identify packets with an 802.1BR header, bypass the header, and send the packet to next processing port.

NOTE

The 802.1BR header bypass feature is available only on the following line-cards.

- BR-MLX-40Gx4
- BR-MLX-10Gx20
- BR-MLX-100Gx2-CFP2

The syntax of the **bypass-802-1br** command is discussed in the following examples.

Syntax: `[no] bypass-802-1br all | slot slot-num | slot slot-num device device-id`

NOTE

To use the 802.1BR header bypass functionality, switch to **config-pkt-encap-proc** command line interface (CLI) mode.

Configuring 802.1BR header bypass on all modules

The **bypass-802-1br all** command enables the 802.1BR header bypass feature on all the cards that support this feature.

```
device(sw)# config terminal
device(config)# packet-encap-processing
device(config-pkt-encap-proc)# bypass-802-1br all
```

Syntax: `[no] bypass-802-1br all`

Configuring 802.1BR header bypass on a specific module

The **bypass-802-1br slot** command enables the 802.1BR header bypass feature on a specific card that supports this feature.

NOTE

Slot-num represents the module ID.

```
device(sw)# config terminal
device(config)# packet-encap-processing
device(config-pkt-encap-proc)# bypass-802-1br slot 3
```

Syntax: `[no] bypass-802-1br slot slot-num`

The following example enables the 802.1BR header bypass feature on a specific device (identified using a device-ID) of a card, that supports this feature.

NOTE

Slot-num represents the module ID and device-id represents the np-id.

```
device(sw)# config terminal
device(config)# packet-encap-processing
device(config-pkt-encap-proc)# bypass-802-1br slot 2 device-id 1
```

Syntax: `[no] bypass-802-1br slot slot-num device device-id`

VN-tag header stripping

As part of VN-tag header stripping feature, the system strips the VN-tag header from the ingress traffic as some analytic tools do not understand the VN-tag header.

NOTE

The VN-tag header stripping feature is available only on the following line-cards.

- BR-MLX-40Gx4
- BR-MLX-10Gx20
- BR-MLX-100Gx2-CFP2

The syntax of the **strip-vn-tag** command is discussed in the following examples.

Syntax: `[no] strip-vn-tag all | slot slot-num | slot slot-num device device-id`

NOTE

To use the VN-tag header stripping functionality, switch to **config-pkt-encap-proc** command line interface (CLI) mode.

Configuring VN-tag header stripping on all modules

The **strip-vn-tag all** command enables the VN-tag header stripping feature on all the cards that support this feature.

```
device(sw)# config terminal
device(config)# packet-encap-processing
device(config-pkt-encap-proc)# strip-vn-tag all
```

Syntax: **[no] strip-vn-tag all**

Configuring VN-tag header stripping on a specific module

The **strip-vn-tag slot** command enables the VN-tag header stripping feature on a specific card that supports this feature.

NOTE

Slot-num represents the module ID.

```
device(sw)# config terminal
device(config)# packet-encap-processing
device(config-pkt-encap-proc)# strip-vn-tag slot 3
```

Syntax: **[no] strip-vn-tag slot slot-num**

The following example enables the VN-tag header stripping feature on a specific device (identified using a device-ID) of a card, that supports this feature.

NOTE

Slot-num represents the module ID and device-id represents the np-id.

```
device(sw)# config terminal
device(config)# packet-encap-processing
device(config-pkt-encap-proc)# strip-vn-tag slot 2 device-id 1
```

Syntax: **[no] strip-vn-tag slot slot-num device device-id**

VN-tag header bypass

As part of VN-tag header bypass feature, the system bypasses the VN-tag header.

NOTE

The VN-tag header bypass feature is available only on the following line-cards.

- BR-MLX-40Gx4
- BR-MLX-10Gx20
- BR-MLX-100Gx2-CFP2

The syntax of the **bypass-vn-tag** command is discussed in the following examples.

Syntax: **[no] bypass-vn-tag all | slot slot-num | slot slot-num device device-id**

NOTE

To use the VN-tag header bypass functionality, switch to **config-pkt-encap-proc** command line interface (CLI) mode.

Configuring VN-tag header bypass on all modules

The **bypass-vn-tag all** command enables the VN-tag header bypass feature on all the cards that support this feature.

```
device(sw)# config terminal
device(config)# packet-encap-processing
device(config-pkt-encap-proc)# bypass-vn-tag all
```

Syntax: [no] bypass-vn-tag all

Configuring VN-tag header bypass on a specific module

The **bypass-vn-tag slot** command enables the VN-tag header bypass feature on a specific card that supports this feature.

NOTE

Slot-num represents the module ID.

```
device(sw)# config terminal
device(config)# packet-encap-processing
device(config-pkt-encap-proc)# bypass-vn-tag slot 3
```

Syntax: [no] bypass-vn-tag slot *slot-num*

The following example enables the VN-tag header bypass feature on a specific device (identified using a device-ID) of a card, that supports this feature.

NOTE

Slot-num represents the module ID and device-id represents the np-id.

```
device(sw)# config terminal
device(config)# packet-encap-processing
device(config-pkt-encap-proc)# bypass-vn-tag slot 2 device-id 1
```

Syntax: [no] bypass-vn-tag slot *slot-num* device *device-id*

Show packet encap processing commands

The status and configuration of 802.1BR and VN-tag header processing are available with the show packet-encap-processing commands.

show packet-encap-processing

The **show packet-encap-processing** command displays the status of the 802.1BR and VN-tag header processing features.

Syntax

```
show packet-encap-processing
```

Modes

Exec mode.

Command Output

The **show packet-encap-processing** command displays the following information:

Output field	Description
Slot ID	Slot ID
Dev ID	Device ID
802.1BR Strip	On or Off
802.1BR Bypass	On or Off
VN-tag Strip	On or Off
VN-tag Bypass	On or Off
NVGRE Strip	On or Off
VXLAN Strip	On or Off

Examples

The following is an example of **show packet-encap-processing** command output.

```
device(config)# show packet-encap-processing
ON      : Feature is configured
-       : Feature is not configured
*       : Feature is not supported
<Blank> : Slot is Empty
```

Slot Id	Dev Id	802.1BR Strip	802.1BR Bypass	VN-Tag Strip	VN-Tag Bypass	NVGRE Strip	VXLAN Strip
S1	1	-	ON	-	ON	-	-
	2	ON	-	ON	-	-	-
S2	1	-	-	-	-	-	-
	2	-	-	-	-	-	-
S3	1	-	ON	-	ON	-	-
	2	ON	-	ON	-	-	-
S4	1	ON	-	ON	-	-	-
	2	-	ON	-	ON	-	-

History

Release version	Command history
6.0.00a	This command was introduced.
6.1.00	Output has been updated.

show packet-encap-processing strip-802-1BR

The **show packet-encap-processing strip-802-1BR** command displays the status of the 802.1BR header processing features.

Syntax

```
show packet-encap-processing strip-802-1BR
```

Modes

EXEC mode.

Command Output

The **show packet-encap-processing strip-802-1BR** command displays the following information:

Output field	Description
Slot	Slot number
Dev	Device number
VN-Tag Strip	On, -, or *

Examples

The following is an example of **show packet-encap-processing strip-802-1BR** command output.

```
device(config)# show packet-encap-processing strip-802-1BR
ON      : Feature is configured
-       : Feature is not configured
*       : Feature is not supported
<Blank> : Slot is Empty
```

```
-----
| Slot| Dev| 802.1BR Strip |
-----
| S1  | 1  | -              |
|     | 2  | -              |
-----
| S2  | 1  | *              |
|     | 2  | *              |
-----
| S3  | 1  | *              |
|     | 2  | *              |
-----
| S4  | 1  | -              |
|     | 2  | ON             |
-----
```

History

Release version	Command history
6.0.00a	This command was introduced.

show packet-encap-processing strip-vn-tag

The `show packet-encap-processing strip-vn-tag` command displays the status of the strip-vn-tag header processing features.

Syntax

```
show packet-encap-processing strip-vn-tag
```

Modes

EXEC mode.

Command Output

The `show packet-encap-processing strip-vn-tag` command displays the following information:

Output field	Description
Slot	Slot number
Dev	Device number
VN-Tag Strip	On, - , or *

Examples

The following is an example of `show packet-encap-processing strip-vn-tag` command output.

```
device(config)# show packet-encap-processing strip-vn-tag
ON      : Feature is configured
-      : Feature is not configured
*      : Feature is not supported
<Blank> : Slot is Empty
-----
| Slot| Dev| VN-Tag Strip |
-----
| S1  | 1  | -           |
|     | 2  | ON          |
-----
| S2  | 1  | *           |
|     | 2  | *           |
-----
| S3  | 1  | *           |
|     | 2  | *           |
-----
| S4  | 1  | -           |
|     | 2  | -           |
-----
```

History

Release version	Command history
6.0.00a	This command was introduced.

show packet-encap-processing slot

The **show packet-encap-processing slot** command displays the status of the 802.1BR and VN-tag header processing features on a elected slot.

Syntax

```
show packet-encap-processing { slot slot-num }
```

Parameters

slot
Identifies the slot to be displayed.

slot-num
The slot number.

Modes

EXEC mode

Command Output

The **show packet-encap-processing slot** command displays the following information:

Output field	Description
Slot ID	Slot ID
Dev ID	Device ID
802.1BR Strip	On or Off
802.1BR Bypass	On or Off
VN-tag Strip	On or Off
VN-tag Bypass	On or Off
NVGRE Strip	On or Off

NOTE

802.1BR Bypass and VN-Tag Bypass are not supported in NetIron R06.0.00a.

Examples

The **show packet-encap-processing slot** command displays the following information:

```
device(config)# show packet-encap-processing slot 1
ON      : Feature is configured
-      : Feature is not configured
*      : Feature is not supported
<Blank> : Slot is Empty

-----
| Slot| Dev| 802.1BR| 802.1BR| VN-Tag | VN-Tag | NVGRE |
| Id  | Id | Strip  | Bypass | Strip  | Bypass | Strip  |
-----
| S1  | 1  | -      | -      | -      | -      | -      |
|     | 2  | -      | -      | ON     | -      | -      |
-----
```

History

Release version	Command history
6.0.00a	This command was introduced.

show packet-encap-processing interface ethernet

The **show packet-encap-processing interface ethernet** command displays the status of the 802.1BR and VN-tag header processing features.

Syntax

```
show packet-encap-processing interface ethernet
```

Modes

EXEC mode.

Command Output

The **show packet-encap-processing interface ethernet** command displays the following information:

Output field	Description
Port State	Enabled or Disabled
802.1BR Stripping	On or Off
802.1BR Preservation	On or Off
VN-tag Stripping	On or Off
VN-tag Preservation	On or Off
NVGRE Stripping	On or Off

NOTE

802.1BR Preservation and VN-Tag Preservation are not supported in NetIron R06.0.00a.

Examples

The following is an example of **show packet-encap-processing interface ethernet** command output.

```
device(config)# show packet-encap-processing interface ethernet 1/1
-----
Port State :                               Disabled
-----
Feature-Name                               Status
802.1BR Stripping                          ON
802.1BR Preservation                        OFF
VN-tag Stripping                           OFF
VN-tag Preservation                         OFF
NVGRE Stripping                             OFF
-----
```

History

Release version	Command history
6.0.00a	This command was introduced.

show running-config

The show running-config output discussed in this section is unique to config-pkt-encap-proc mode.

Syntax

```
show running-config
```

Modes

config-pkt-encap-proc mode

Examples

The following is an example output when the functionality has been enabled on all ppcrs have been configured.

```
device(config-pkt-encap-proc)# strip-802-1br all
device(config-pkt-encap-proc)# show running-config
packet-encap-processing
    strip-802-1br all
```

The following is an example output when the functionality has been enabled on all ppcrs have been configured, and then disabled on a selective ppcr.

```
device(config-pkt-encap-proc)# strip-802-1br all
device(config-pkt-encap-proc)# no strip-802-1br slot 4 device-id 1
device(config-pkt-encap-proc)# show running-config
packet-encap-processing
    strip-802-1br slot 1
    strip-802-1br slot 2
    strip-802-1br slot 3
    strip-802-1br slot 4 device-id 2
```

History

Release version	Command history
6.0.00a	This command was introduced.

VXLAN header stripping

This feature allows the user to strip the Virtual Extensible LAN (VXLAN) header from the incoming VXLAN packets.

As part of this feature, the system will identify packets with a VXLAN header, strip the header, and send the packet to next processing port.

NOTE

The VXLAN header stripping feature is only supported on the NPB FPGA for BR-MLX-10Gx20 and BR-MLX-100Gx2-CFP2 modules.

NOTE

The VXLAN header stripping feature is supported on the MAIN FPGA for BR-MLX-40Gx4 module.

NOTE

This feature will not work unless the system is configured as a Network Packet Broker by enabling **fpga-mode-npb**.

The syntax of the **strip-vxlan** command is discussed in the following examples.

Syntax: **[no] strip-vxlan all | slot slot-num | slot slot-num device-id slot-number np-id**

NOTE

To use the VXLAN header stripping functionality, switch to **config-pkt-encap-proc** command line interface (CLI) mode.

Configuring VXLAN header stripping on all modules

The **strip-vxlan all** command enables the VXLAN header stripping feature on all cards that support this feature.

```
device(sw)# config terminal
device(config)# packet-encap-processing
device(config-pkt-encap-proc)# strip-vxlan all
```

Configuring VXLAN header stripping on a specific module

The **strip-vxlan slot** command enables the VXLAN header stripping feature on the specified card that supports this feature.

NOTE

Slot-num represents the module ID.

```
device(sw)# config terminal
device(config)# packet-encap-processing
device(config-pkt-encap-proc)# strip-vxlan slot 3
```

Syntax: **[no] strip-vxlan slot slot-num**

The following example enables the VXLAN header stripping feature on a specific device (identified using a device-ID) of a card, that supports this feature.

NOTE

Slot-num represents the module ID and device-id represents the np-id.

```
device(sw)# config terminal
device(config)# packet-encap-processing
device(config-pkt-encap-proc)# strip-vxlan slot 2 device-id 1
```

Syntax: **[no] strip-vxlan slot slot-num device device-id**

NVGRE header stripping

In a packet broker configuration, the ingress traffic to the MLXe will be processed and forwarded to the analytic tools. For such traffic, the NVGRE headers will not be useful in forwarding the packets. For such scenarios, this feature strips the NVGRE header from incoming NVGRE packets.

As part of this feature, the system will identify packets with an NVGRE header, strip the header, and send the packet to the next processing port.

NOTE

The NVGRE header stripping feature is only supported on the NPB FPGA for BR-MLX-10Gx20 and BR-MLX-100Gx2-CFP2 modules.

NOTE

The NVGRE header stripping feature is supported on the MAIN FPGA for BR-MLX-40Gx4 module.

NOTE

This feature will not work unless the system is configured as a Network Packet Broker by enabling **fpga-mode-npb**.

The **all** option applies the command to all the slots in the system. If the line card is not a supported card, the command will be ignored.

Specifying a **slot number** using the slot-number variable limits the command to an individual module.

Specifying a **slot number and a network processor ID** using the slot-number and ppcr-id variables limits the command to the ports supported by the specified network processor on the specified interface module.

Configuring NVGRE header stripping

The following is an example of configuring NVGRE header stripping.

```
device# config terminal
device(config)# packet-encap-processing
device(config-pkt-encap-proc)# strip-nvgre slot 1 device-id 1
```

Syntax: **[no] strip-nvgre all | slot slot-num | slot slot-num device device-id**

NVGRE header stripping show command

The following is an example of configuring NVGRE header stripping.

NOTE

The **show packet-encap-processing** command can also be used to show the NVGRE configuration.

```
device# show packet-encap-processing strip-nvgre

-----
Displaying Information of slot 1 device 1:
  Feature-Name      Status
  NVGRE Stripping   ON

Displaying Information of slot 1 device 2:
  Feature-Name      Status
  NVGRE Stripping   ON

-----
Displaying Information of slot 2 device 1:
  Feature-Name      Status
  NVGRE Stripping   ON

Displaying Information of slot 2 device 2:
  Feature-Name      Status
  NVGRE Stripping   ON

-----
Displaying Information of slot 3 device 1:
  Feature-Name      Status
  NVGRE Stripping   ON

Displaying Information of slot 3 device 2:
  Feature-Name      Status
  NVGRE Stripping   ON

-----
Displaying Information of slot 4 device 1:
  Feature-Name      Status
  NVGRE Stripping   ON

Displaying Information of slot 4 device 2:
  Feature-Name      Status
  NVGRE Stripping   ON

-----
MLXe#
```

Syntax: show packet-encap-processing strip-nvgre

GTP de-encapsulation configuration

GTP de-encapsulation

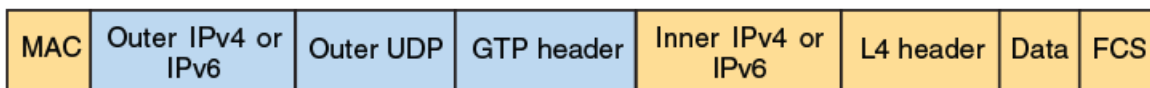
NetIron BR-MLX-1GX20-U10G-M, BR-MLX-1000GX2-CFP2-M, and BR-MLX-40Gx4-M modules support the general packet radio service (GPRS) Tunneling Protocol (GTP) de-encapsulation feature.

GTP is a group of IP-based communications protocols used to transmit general packet radio service (GPRS) within global system for mobile communication (GSM), universal mobile telecommunications service (UMTS) and long-term evolution (LTE) networks in a 3rd Generation Partnership Project (3GPP) architecture.

In a packet-broker configuration, the ingress traffic to an MLXe switch is tapped and forwarded to the analytic tools. In GTP encapsulation, the packet content inside the Layer 2 header is encapsulated inside the new Layer 2 header, Layer 3 and Layer 4 headers. These new headers represent the two main network nodes that the GTP tunnels have been established. The nodes can be serving GPRS support node (SGSN) and gateway GPRS support node (GGSN) in a 3G network, or between an E-UTRAN Node B (eNodeB) and serving gateway (SGW), or SGW and packet data network gateway (PGW) in an LTE network. For the tapped traffic the tunneling headers, such as GTP, are not needed by analytic tools. In such cases, the GTP de-encapsulation feature allows the user to de-encapsulate GTP packets.

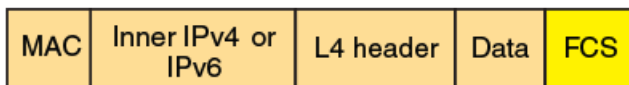
In an L4 UDP header, the port for GTP control (GTP-C) packets is 2123 and GTP data (GTP-U) packets port is 2152. De-encapsulation of GTP-C packets is not required because the GTP-C packets do not have inner IPv4 or IPv6 headers. GTP-C packets remain unchanged if the GTP de-encapsulation feature is enabled. The GTP de-encapsulation feature is useful only for GTP-U packets.

The following figure is an example of a GTP-U packet.



In GTP de-encapsulation, the outer IP, the outer UDP header and the GTP header are removed from GTP-U packets. The cyclic redundancy check (CRC) of the frame is recalculated and the frame check sequence (FCS) field is updated. The MAC header and the inner L3 and L4 headers are retained.

The following figure depicts the above packet after stripping the GTP header.



You can configure a port to de-encapsulate GTP packets after enabling the GTP de-encapsulation feature on that port. The GTP de-encapsulation feature is supported for both IPv4 and IPv6.

Configuring GTP de-encapsulation

Use the following instructions to enable the GTP de-encapsulation feature on a NetIron BR-MLX-1GX20-U10G-M or BR-MLX-1000GX2-CFP2-M, or BR-MLX-40Gx4-M.

1. Enter global configuration mode.

```
device# configuration terminal
```

2. Enter interface configuration mode.

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

3. Use the **gtp-de-encapsulation** command to enable GTP de-encapsulation feature on the interface.

```
(config-if-e40000-1/1)# gtp-de-encapsulation
```

show gtp-de-encapsulation slot

Displays whether or not the GPRS Tunneling Protocol (GTP) de-encapsulation is configured on all interfaces in a slot.

Syntax

```
show gtp-de-encapsulation slot slot-number
```

Parameters

slot-number

Specifies the slot number.

Modes

User EXEC mode

Usage Guidelines

This command is supported only on the BR-MLX-10Gx20 and BR-MLX-100Gx2 modules.

Command Output

The **show gtp-de-encapsulation** command displays the following information:

Output field	Description
Port	Displays the port number.
GTP De-encapsulation	Displays the GTP De-encapsulation feature.
State	Displays the state of the port.

Examples

The following example displays whether or not the GTP de-encapsulation is configured on all interfaces in a slot.

```
device# show gtp-de-encapsulation slot 1
-----
| Port          | GTP De-encapsulation | State          |
-----
| 1/1          | Configured           | Up            |
-----
| 1/2          | Not Configured       | Disabled      |
-----
| 1/3          | Not Configured       | Disabled      |
-----
| 1/4          | Not Configured       | Disabled      |
-----
```

History

Release version	Command history
6.0.00a	This command was introduced.

show gtp-de-encapsulation

Displays all interfaces with the GPRS Tunneling Protocol (GTP) de-encapsulation feature enabled.

Syntax

```
show gtp-de-encapsulation
```

Modes

User EXEC mode

Usage Guidelines

This command is supported only on the BR-MLX-10Gx20 and BR-MLX-100Gx2 modules.

Command Output

The **show gtp-de-encapsulation** command displays the following information:

Output field	Description
Port	Displays the port number.
State	Displays the state of the port.

Examples

The following example displays all interfaces with the GTP de-encapsulation feature enabled.

```
device# show gtp-de-encapsulation
GTP de-encapsulation is configured on the following ports:
```

```
-----
| Port      | State      |
-----
| 1/1      | Up        |
-----
| 4/1      | Up        |
-----
```

History

Release version	Command history
6.0.00a	This command was introduced.

show gtp-de-encapsulation interface

Displays whether or not the GPRS Tunneling Protocol (GTP) de-encapsulation is configured on an interface.

Syntax

```
show gtp-de-encapsulation interface interface-number
```

Parameters

interface-number

Specifies the interface number.

Modes

User EXEC mode

Usage Guidelines

This command is supported only on the BR-MLX-10Gx20 and BR-MLX-100Gx2 modules.

Command Output

The **show gtp-de-encapsulation** command displays the following information:

Output field	Description
Port	Displays the port number.
GTP De-encapsulation	Displays the GTP De-encapsulation feature.
State	Displays the state of the port.

Examples

The following example displays whether or not the GTP de-encapsulation is configured on an interface.

```
device# show gtp-de-encapsulation interface ethernet 4/1
-----
| Port          | GTP De-encapsulation | State          |
-----
| 4/1          | Configured           | Up            |
-----
```

History

Release version	Command history
6.0.00a	This command was introduced.

VN-tag and 802.1BR preserve enhancement

This feature enables the system to preserve the VN-tag and 802.1BR header from ingress traffic to comply with analytical tools that understand the VN-tag and 802.1BR header.

As part of the VN-tag and 802.1BR preserve function, in the initial part of the processing, the VN-tag/802.1BR is kept aside in a buffer and the VLAN (if present after the VN-tag and 802.1BR header) is learned by the system. This VLAN info can be changed using the configuration command.

NOTE

VN-tag and 802.1BR header preserve feature is available only on the following line-cards.

- BR-MLX-100Gx2-CFP2
- BR-MLX-40Gx4- X
- BR-MLX-10Gx20
-

Limitations

- The BR-MLX-10Gx24 card supports only up to 480 TVF LAG load balancing egress streams.

Configuration considerations

- Before using the header preserve enhancement functionality, switch to **config-routemap pbr_tvf_vlan** command line interface (CLI) mode.
- Egress ports need to be untagged members of VLAN-x, where VLAN-x is a dummy VLAN used in **replace-vlan** command.
- A PPCR in service card will only serve either Bypass function for both 802.1BR/VN-tag or Strip function for both. It cannot serve Bypass for 802.1BR and Strip for VN-tag or vice-versa.
- The **replace-vlan** command should only be applied for 802.1BR/VN-tag Strip/Bypass flows.
- The **replace-vlan** command should only be applied to ports which are marked to handle 802.1BR/VN-tag Strip/Bypass traffic.
- Route-map with the **replace-vlan** option is mandatory for the preserve feature to work; whereas, it is only required for **strip** when 10G/100G cards are used as a service-card for stripping these tags. Its not required for **strip** with 40G service-card.
- The port where the **replace-vlan** command is applied for 802.1BR/VN-tag preserve can neither serve 802.1BR/VN-tag stripped nor Non 802.1BR/VN-tag traffic

The syntax of the commands are discussed in the following examples.

Syntax: `[no] set next-hop-tvf-domain tvf-domain-id replace-vlan vlan-X`

Syntax: `[no] set next-hop-flood-vlan vlan-id replace-vlan vlan-X`

NOTE

To use the VN-tag and 802.1BR preserve functionality, switch to **config-routemap pbr_tvf_vlan** command line interface (CLI) mode.

Configuring the VN-tag and 802.1BR preserve feature

NOTE

Before using the header preserve enhancement functionality, switch to **config-routemap pbr_tvf_vlan** command line interface (CLI) mode.

In a PBR, this route-map should only be configured on 802.1BR/VN-tag Bypass or 802.1BR/VN-tag Strip PPCR. For example, if the incoming VLAN is 10, flooding is done on VLAN 20, and if the user wants to change the outgoing VLAN to 30 then user should use **replace-vlan 30**.

There are two commands that can be used to accomplish this:

Syntax: [no] set next-hop-tvf-domain tvf-domain-id replace-vlan vlan-X

Syntax: [no] set next-hop-flood-vlan vlan-id replace-vlan vlan-X

Vlan-X is the dummy VLAN for which the egress port should be an untagged member.

The following is an example of the configuration:

```
device(sw)# config terminal
device(config)# config-routemap pbr_tvf_vlan
device(config-routemap pbr_tvf_vlan)# set next-hop-flood-vlan vlan-20 replace-vlan vlan-30
```

Syntax: [no] set next-hop-flood-vlan replace-vlan

802.1BR/VN-tag preserve use case example

The following is an example configuration for the 802.1BR/VN-tag preserve use case. 6/18 is the Ingress port, 7/18 the Service port, and eth 1/1, eth 2/1, eth 5/1, and eth 6/1 are the Egress ports.

```
!
lag "test" static id 2
ports ethernet 1/1 ethernet 2/1
primary-port 1/1
deploy
!
vlan 1 name DEFAULT-VLAN
no untagged eth 1/1 eth 2/1 eth 5/1 eth 6/1
no spanning-tree
!
vlan 100
tagged ethe 7/18
transparent-hw-flooding
!
vlan 200
tagged eth 1/1 eth 2/1 ethe 5/1 ethe 6/1
transparent-hw-flooding
!
vlan 222
untagged eth 1/1 eth 2/1 eth 5/1 eth 6/1
transparent-hw-flooding
!
tvf-domain 1
port eth 7/18
!
tvf-domain 2
port eth 1/1 eth 2/1 eth 5/1 eth 6/1
!
system-max tvf-lag-lb-fid-pool 4096
system-max tvf-lag-lb-fid-group 4
!
no route-only
!
interface ethernet 1/1
```

```

enable
!
interface ethernet 2/1
enable
! interface ethernet 5/1
enable
!
interface ethernet 6/1
enable
!
interface ethernet 6/18
enable
ip policy route-map mall
ipv6 policy route-map mallv6
allow-all-vlan pbr
!
!
interface ethernet 7/18
enable
loopback system
ip policy route-map mall-service
ipv6 policy route-map mallv6-service
allow-all-vlan pbr
!
!
ip access-list extended MME_1_ACL
permit vlan 114 udp any eq 1066 any
!
ip access-list extended SGW_1_ACL
permit vlan 2405 ip host 10.33.44.55 any
!
ipv6 access-list MME_1_ACL_ipv6
permit vlan 114 udp any eq 1066 any
!
!
route-map mall permit 100
rule-name MME_1
match ip address MME_1_ACL
set next-hop-tvf-domain 1
!
route-map mall-service permit 100
rule-name MME_1
match ip address MME_1_ACL
set next-hop-tvf-domain 2 replace-vlan 222
!
route-map mall permit 110
rule-name SGW_1
match ip address SGW_1_ACL
set next-hop-flood-vlan 100
!
route-map mallv6 permit 100
match ipv6 address MME_1_ACL_ipv6
set next-hop-tvf-domain 1
!
route-map mallv6-service permit 100
match ipv6 address MME_1_ACL_ipv6
set next-hop-tvf-domain 2 replace-vlan 222
!
packet-encap-processing
bypass-802-1br slot 7
bypass-vn-tag slot 7

```

IP payload length based filtering using ACL

Using this feature a range of IP payload length can be configured to be used for filtering of traffic with ACL.

The IP payload length is the size of the data portion of the IP datagram. The IP payload length range can be configured for port per packet processor (PPCR) filtering. This range of IP payload length then can be used as a filter parameter with the Access control List (ACL). This feature is supported for both IP and IPv6 traffic. The IP payload length based filtering using ACL feature allows a user to filter ingress IP/IPv6 traffic based on IP payload length of packets.

- IP payload length is the size of data carried in IP packets.
- In IPv4 packets payload length is the total length excluding the IP header length.
- In IPv6 packets payload length is present in the IPv6 header.
- The range of IP payload length can be configured for both versions.
- IPv4 extended and IPv6 ACL filters can be configured with the match-payload-len clause.
- IP packets having a payload length inside the configured range will be filtered with using the ACL.
- IPv4 packets with option header and IPoMPLS transit traffic are not supported with feature.

NOTE

IPv4 packets with option header and IPoMPLS transit traffic are not supported with feature.

NOTE

In a case where the IP payload length range is not configured all the filters with match-payload-len will be ignored and packets are not matched.

Configuration steps

The IP payload length range can be updated on either the global, slot or PPCR level.

- Enable and configure the IP payload length for PPCR using the `ip match-payload-len` command.
- Enable the IP payload length check attribute in ACL filter.
- Apply ACL on the interface.

Enabling and configuring the IP payload length for PPCR

The ACL filter creation command supports the match payload length attribute.

```
device(config)# access-list 111 permit ip any any match-payload-len
```

Globally configure the IPv4 payload length

This command sets the IP payload length range [700, 1000] in each PPCR of all slots.

```
device(config)#ip match-payload-len slot all range 700 1000
```

Configure the IPv4 payload length on all PPCR on a given slot

This command will set the IP payload length equal to 800 for all PPCR in slot 2.

```
device(config)#ip match-payload-len slot 2 range 800 800
```

Configure the IPv4 payload length on a selected PPCR

This command will set the IP payload length less than or equal to 1000 for PPCR 1 of slot 1.

```
device(config)#ip match-payload-len slot 1 ppcr 1 range 0 1000
```

Removing IPv4 payload length configuration from a selected PPCR

This command will remove the IP payload length configuration from PPCR 1 of slot 2. When using the remove command the range attribute and values are not required.

```
device(config)#no ip match-payload-len slot 2 ppcr 1
```

Enabling and configuring the IPv6 payload length for PPCR

The ACL filter creation command supports the match payload length attribute.

```
device(config-ipv6-access-list payload)#permit ipv6 any any match-payload-len
```

Globally configure the IPv6 payload length

This command sets the IPv6 payload length range [700, 1000] in each PPCR of all slots.

```
device(config)#ipv6 match-payload-len slot all range 700 1000
```

Configure the IPv6 payload length on all PPCR on a given slot

This command will set the IPv6 payload length equal to 800 for all PPCR in slot 2.

```
device(config)#ipv6 match-payload-len slot 2 range 800 800
```

Configure the IPv6 payload length on a selected PPCR

This command will set the IPv6 payload length less than or equal to 1000 for PPCR 1 of slot 1.

```
device(config)#ipv6 match-payload-len slot 1 ppcr 1 range 0 1000
```

Removing IPv6 payload length configuration from a selected PPCR

This command will remove the IPv6 payload length configuration from PPCR 1 of slot 2. When using the remove command the range attribute and values are not required.

```
device(config)#no ipv6 match-payload-len slot 2 ppcr 1
```

show ip match-payload-len

This show command displays the configuration for all PPCRs on which IP payload length range is configured.

Syntax

```
show ip match-payload-len interface ethernet slot / port
```

Modes

EXEC mode.

Command Output

The `show ip match-payload-len` command displays the following information:

Output field	Description
Slot	Slot number
PPCR	PPCR number
Min-Payload-length	Minimum configured payload length
Max-Payload-length	Maximum configured payload length

Examples

The following is an example of the show command for the IP payload length system configuration.

```
device(config)#show ip match-payload-len
IP Match Payload Length Configuration
Slot      PPCR      Min-Payload-length      Max-Payload-length
1         1         0                       1000
1         2         700                     1000
2         2         800                     800
3         1         700                     1000
3         2         700                     1000
```

The following is an example of the show command for the IP payload length configuration on a specific interface.

```
device(config)#show ip match-payload-len interface ethernet 1/5
IP Match Payload Length Configuration
Slot      PPCR      Min-Payload-length      Max-Payload-length
1         2         0                       1000
```

History

Release version	Command history
6.0.00a	This command was introduced.

show ip match-payload-len interface ethernet

This show command displays the IP payload length configuration on a specific interface.

Syntax

```
show ip match-payload-len { interface ethernet slot | port }
```

Parameters

interface ethernet

Indicates a specific interface output to be displayed.

slot

The specific slot of the interface.

port

The specific port of the interface.

Modes

EXEC mode

Command Output

The `show ip match-payload-len interface ethernet` command displays the following information:

Output field	Description
Slot	Slot number
PPCR	PPCR number
Min-Payload-length	Minimum configured payload length
Max-Payload-length	Maximum configured payload length

Examples

The following is an example of the show command for the IP payload length configuration on a specific interface.

```
device(config)#show ip match-payload-len interface ethernet 1/5
IP Match Payload Length Configuration
Slot      PPCR      Min-Payload-length      Max-Payload-length
1         2         0                       1000
```

History

Release version	Command history
6.0.00a	This command was introduced.

show ipv6 match-payload-len

This show command displays the configuration for all PPCRs on which IPv6 payload length range is configured.

Syntax

```
show ipv6 match-payload-len
```

Modes

EXEC mode

Command Output

The `show ipv6 match-payload-len` command displays the following information:

Output field	Description
Slot	Slot number
PPCR	PPCR number
Min-Payload-length	Minimum configured payload length
Max-Payload-length	Maximum configured payload length

Examples

The following is an example of the show command for the IP payload length system configuration.

```
device(config)#show ipv6 match-payload-len
IPv6 Match Payload Length Configuration
Slot      PPCR      Min-Payload-length      Max-Payload-length
1         1         0                       1000
1         2         700                     1000
2         2         800                     800
3         1         700                     1000
3         2         700                     1000
```

History

Release version	Command history
6.0.00a	This command was introduced.

show ipv6 match-payload-len interface ethernet

This show command displays the IPv6 payload length configuration on a specific interface.

Syntax

```
show ipv6 match-payload-len { interface ethernet slot | port }
```

Parameters

interface ethernet

Indicates a specific interface output to be displayed.

slot

The specific slot of the interface.

port

The specific port of the interface.

Modes

EXEC mode

Command Output

The `show ipv6 match-payload-len interface ethernet` command displays the following information:

Output field	Description
Slot	Slot number
PPCR	PPCR number
Min-Payload-length	Minimum configured payload length
Max-Payload-length	Maximum configured payload length

Examples

The following is an example of the show command for the IPv6 payload length configuration on a specific interface.

```
device(config)#show ipv6 match-payload-len interface ethernet 1/5
IPv6 Match Payload Length Configuration
Slot      PPCR      Min-Payload-length      Max-Payload-length
1         2         0                        1000
```

History

Release version	Command history
6.0.00a	This command was introduced.

Source port labeling

Source port labeling will append a source port label to the incoming packets on a MLXe when this feature is enabled. This feature is used in Network Packet Broker deployments.

This feature is supported on BR-MLX-10Gx20 and BR-MLX-100Gx2-CFP2 cards only in the NPB-FPGA.

This feature is supported on BR-MLX-40Gx4 in the main-FPGA.

If both **packet timestamping** and **source port labeling** features are enabled, then the timestamp bytes will precede the port label bytes followed by the recalculated CRC.

Source port labeling configuration

The **source-port-label all** command enables the source-port-label feature on all the cards that support this feature.

```
device# config terminal
device(config)# source-port-label all
```

Syntax: [no] **source-port-label all**

Configuring source port labeling on a slot

The **source-port-label slot** command enables the source-port-label feature on the indicated slot.

```
device# config terminal
device(config)# source-port-label slot 2
```

Syntax: [no] **source-port-label slot** *slot number*

Configuring source port labeling using a device ID

The **source-port-label slot** command enables the source-port-label feature on the device-id for a particular slot.

```
device# config terminal
device(config)# source-port-label slot 2 device-id 1
```

Syntax: [no] **source-port-label slot** *slot number device-id device-id number*

Disabling source port labeling

The **no source-port-label all** command disables the source-port-label feature on all ports.

```
device# config terminal
device(config)# no source-port-label all
```

The **no source-port-label slot** command disables the source-port-label feature on the indicated slot.

```
device# config terminal
device(config)# no source-port-label slot 1
```

The **no source-port-label slot device-id** command disables the source-port-label feature on the indicated device-id on the slot.

```
device# config terminal
device(config)# no source-port-label slot 2 device-id 1
```

Source port label show command

The **show source-port-label** command displays the source-port-label feature on all ports.

```
device# show source-port-label
Slot-Num  Device-Id  Src-Port-Label  Card-State
1          All       Configured       CARD_STATE_UP
2          All       Configured       CARD_STATE_NOT_PRESENT
3          All       Configured       CARD_STATE_UP
```

Syntax: **show source-port-label slot** *slot number* **device-id** *device-id number*

The **show source-port-label interface ethernet** command displays the source-port-label feature on the indicated interface.

```
device# show source-port-label interface ethernet 1/1
Source Port Label : Configured
Port State       : Up
```

SCTP port based traffic filtering using ACL

The Stream Control Transmission Protocol (SCTP) port based traffic filtering using ACL feature allows you to filter SCTP traffic based on the L4 source and destination ports of packets.

The Stream Control Transmission Protocol (SCTP) port based traffic filtering using ACL feature enhances the access control list (ACL) to support SCTP packet filtering based on source and destination L4 ports. Traffic flow will be identified with help of port range and appropriate action will be performed according to ACL filters.

Different SCTP operators such as **greater than** (gt), **less than** (lt), **equal to** (eq), **not equal to** (neq), **range** can be used to give the range of source and destination L4 ports for SCTP traffic flow.

NOTE

SCTP protocol L4 source and destination port based filtering is only supported when ACL rules are provisioned with protocol names. SCTP protocol L4 source and destination port based filtering will not be supported when ACL rules are provisioned with protocol numbers instead of protocol names.

Configuring IPv4 Extended access control list

```
device# config terminal
device(config)#access-list 111 permit sctp 11.110.110.1 0.0.0.255 neq 129 20.20.20.1 0.0.0.255 gt 100
device(config)#access-list 111 permit sctp 11.110.110.1 0.0.0.255 neq 129 20.20.20.1 0.0.0.255 gt 100
device(config)#access-list 111 permit sctp any neq 129 any gt 100
device(config)#access-list 111 permit sctp 111.110.110.1 0.0.0.255 lt 129 20.20.20.1 0.0.0.255 gt 100
device(config)#access-list 111 deny sctp any range 10 100 any eq 10
```

Syntax: [**no**] **access-list** *access-list num* **deny** | **permit** [**sctp** *source-ip* | *hostnamewildcard* | **any**] [**sctp-operator** (gt, lt, eq, or neq) *dest-l4-port*] [*min-dest-l4-port* *max-dest-l4-port*]

Show configuration of an IPv4 extended access list with SCTP Port filter

```
device(config)#show acc 111

Extended IP access list 111 : 5 entries
10: permit sctp 11.110.110.0 0.0.0.255 neq pwdgen 20.20.20.0 0.0.0.255 gt newacct
20: permit sctp 11.110.110.0 0.0.0.255 neq pwdgen 20.20.20.0 0.0.0.255 gt newacct
30: permit sctp any neq pwdgen any gt newacct
40: permit sctp 111.110.110.0 0.0.0.255 lt pwdgen 20.20.20.0 0.0.0.255 gt newacct
50: deny sctp any range 10 newacct any eq 10
```

Syntax: show access-list

Configuring IPv6 access control list with SCTP filters

```
device# configure terminal
device(config)#ipv6 access-list new_sctp_filter
device(config-ipv6-access-list new_sctp_filter)#permit sctp 201:1::1:1/64 eq 28 any range 10 100
device(config-ipv6-access-list new_sctp_filter)#permit sctp 201:1::1:1/64 lt 28 301:1::1:1/64 gt 100
device(config-ipv6-access-list new_sctp_filter)#deny sctp any lt 28 any gt 100
device(config-ipv6-access-list new_sctp_filter)#exit
```

Show configuration of given IPv6 access control list with SCTP Port filter

```
device(config)#show ipv6 access-list new_sctp_filter
ipv6 access-list new_sctp_filter: 3 entries
10: permit sctp 201:1::/64 eq 28 any range 10 newacct
20: permit sctp 201:1::/64 lt 28 301:1::/64 gt newacct
30: deny sctp any lt 28 any gt newacct
```

Packet timestamping

The packet timestamping feature appends eight bytes of timestamp (followed by four bytes of FCS) to the incoming packets on an MLXe.

NOTE

The packet timestamping feature is available only on the following line-cards.

- BR-MLX-40Gx4 - only in the main-FPGA.
- BR-MLX-10Gx20 - only in the NPB-FPGA
- BR-MLX-100Gx2-CFP2 - only in the NPB-FPGA

The packet timestamp feature will append a timestamp to the incoming packets on an MLXe when this feature is enabled. The timestamping feature will be configurable per ppcr. When this feature is enabled, the time stamp of eight bytes will be added after the payload, followed by four bytes of CRC. The CRC will be recalculated after inserting the timestamp in the packet.

Configuration options:

- The all option applies the command to all the slots in the system. Cards that do not support the feature will ignore the command.
- Specifying a slot number using the slot-number variable limits the command to an individual module.
- Specifying a slot number and a network processor ID using the slot-number and np-id variables limits the command to the ports supported by the specified network processor on the specified interface module.

Configuring packet timestamping

The following example shows the configuration process for the timestamping feature.

```
device# config terminal
device(config)# packet-timestamp slot all
```

Syntax: [no] packet-timestamp slot all | slot *slot-num* | slot *slot-num* device *device-id*

Packet timestamping show command

The following example shows the output of the show packet-timestamp command.

```
device(config)# show packet-timestamp

Slot-Num  Device-Id  Pkt-Timestamp  Card-State
1          All       Configured     CARD_STATE_UP
2          All       Configured     CARD_STATE_UP
4          All       Configured     CARD_STATE_UP
5          All       Configured     CARD_STATE_UP
6          All       Configured     CARD_STATE_UP
8          All       Configured     CARD_STATE_UP
9          All       Configured     CARD_STATE_UP
10         All       Configured     CARD_STATE_UP
11         All       Configured     CARD_STATE_UP
12         All       Configured     CARD_STATE_UP
13         All       Configured     CARD_STATE_UP
14         All       Configured     CARD_STATE_UP
15         All       Configured     CARD_STATE_UP
16         All       Configured     CARD_STATE_UP
```

```
Time source is Set Clock: 20:24:28.700 GMT+00 Tue Aug 23 2016
```

Syntax: show packet-timestamp

Packet timestamping interface show command

The following example shows the output of the show packet-timestamp interface command.

```
device# show packet-timestamp interface ethernet 2/1

Packet Timestamp      : Enabled
```

Syntax: show packet-timestamp interface ethernet *slot-num* / *port-num*