CONFIGURATION GUIDE

Extreme NetIron Network Packet Broker Configuration Guide, 06.3.00

Supporting NetIron OS 06.3.00

9035612
31 August 2018

Contents

Preface

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.
	Identifies text to enter in the GUI.
italic text	Identifies emphasis.
	Identifies variables.
	Identifies document titles.
`Courier font`	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.
italic text	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, *member*[*member*...].
\	Indicates a “soft” line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

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 https://www.extremenetworks.com/documentation-feedback/.
- Email us at documentation@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

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

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

Supported software

For the complete list of supported features and the summary of enhancements and configuration notes for this release, refer to the *Extreme NetIron Release Notes*.

What's new in this document

This topic lists new and modified features for the current release.

The maximum TVF LAG FID group size (**system-max tvf-lag-lb-fid-group**) is increased to 32. For implementation details, refer to the VLAN > "Transparent VLAN flooding" section of the *Extreme NetIron Layer 2 Switching Configuration Guide*.

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 basics

Network Packet Broker overview

A Network Packet Broker (NPB) provides a collection of monitoring tools with access to traffic across the network.

Two FPGA bundles are available for download.

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

NPB support and features

Under NetIron, NPB is supported only on the MLXe series.

Some 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, make sure that you are using the correct NPB FPGA files. All the other NPB features are enabled on all line cards and on both the Main and NPB FPGAs.

TABLE 2 Network Packet Broker compatibility

MLXe module	NPB FPGA	Main FPGA
BR-MLX-10Gx20	• Packet Timestamping • NVGRE stripping • Source port labeling • VXLAN stripping • GTP de-encapsulation	• BR/VN tags • packet-len-filter • SCTP source ports
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	• Packet Timestamping • NVGRE stripping • Source port labeling • VXLAN stripping • GTP de-encapsulation	• BR/VN tags • packet-len-filter • SCTP source ports

The following NBR 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 modification

Header modification overview

Header modification 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 modification includes:

- 802.1BR header stripping
- 802.1BR header bypass
- VN-tag header stripping
- VN-tag header bypass
- NVGRE header stripping
- 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 NVGRE header-stripping feature performs the following on NVGRE traffic.

- Identify NVGRE traffic.
- Strip NVGRE headers.
- 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 card serves either the Bypass function or the Strip function for both 802.1BR and 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 identifies packets with an 802.1BR header, strip the header, and sends 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.

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

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

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# configure 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# configure 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# configure 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*

VN-tag and 802.1BR preservation

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 preservation 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 preservation feature is available only on the following line-cards:

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

Configuration considerations

- Before using the header preservation functionality, switch to route-map configuration mode.
- Egress ports need to be untagged members of VLAN-x where VLAN-x is a dummy VLAN used in replace-vlan command.
- It is mandatory to use **route-map** with the **replace-vlan** option for the preservation feature to function.
- A PPCR in service card serves 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 preservation 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 preservation 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 preservation functionality, switch to route-map configuration mode.

Configuring VN-tag and 802.1BR preservation

> NOTE
> Before using the header preservation functionality, switch to route-map configuration 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# configure terminal
device(config)# route-map ingress-rmap permit 1
device(config-routemap ingress-rmap)# match l2acl  mac_acl500
device(config-routemap ingress-rmap)# match ip address ipv4_ext_acl750
device(config-routemap ingress-rmap)# match ipv6 address ipv6_ext_acl750
device(config-routemap ingress-rmap)# set next-hop-flood-vlan 2749 replace-vlan 4020
device(config-routemap ingress-rmap)# set interface null0
```

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

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 identifies packets with a VXLAN header, strips the header, and sends the packet to the 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# configure 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# configure 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# configure 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

For ingress NPB traffic, NVGRE headers are not useful in forwarding packets; this feature strips the NVGRE header from incoming NVGRE packets.

As part of this feature, the system identifies packets with an NVGRE header, strips the header, and sends 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 is 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# configure 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#
```

GTP de-encapsulation

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

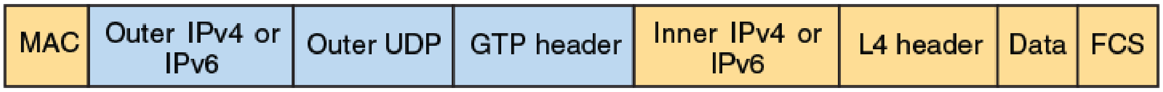
> NOTE
>
> For supported platforms, refer to Network Packet Broker overview on page 11.

General packet radio service (GPRS) Tunneling Protocol (GTP) is a group of IP-based communications protocols used to transmit 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 a MLX Series device 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.

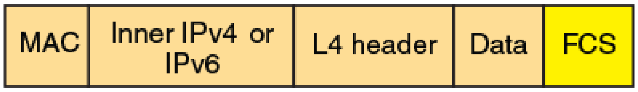
In an L4 UDP header, the port for GTP control (GTP-C) packets is 2123 and GTP data (GTP-U) packets port is 2152. De-encapsulation of GTP-C packets is not required because the GTP-C packets do not have inner IPv4 or IPv6 headers. GTP-C packets remain unchanged if the GTP de-encapsulation feature is enabled.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

Header modification show commands

There are show commands that display details of the header processing features, as listed in the following table.

TABLE 3 Header modification show commands in the *Command Reference*

Command	Description
show gtp-de-encapsulation	Displays all interfaces with the GPRS Tunneling Protocol (GTP) de-encapsulation feature enabled.
show gtp-de-encapsulation interface	Displays whether or not the GPRS Tunneling Protocol (GTP) de-encapsulation is configured on an interface.
show gtp-de-encapsulation slot	Displays whether or not the GPRS Tunneling Protocol (GTP) de-encapsulation is configured on all interfaces in a slot.
show packet-encap-processing	Displays the configured packet encapsulation processing.
show packet-encap-processing bypass-802-1br	Displays the status of the 802.1BR header processing bypass feature.
show packet-encap-processing bypass-vn-tag	Displays the VN-tag bypassing information.
show packet-encap-processing interface ethernet	Displays the packet encapsulation processing configuration on the specified Ethernet interface.
show packet-encap-processing slot	Displays the status of the 802.1BR and VN-tag header processing features on a specified slot.
show packet-encap-processing strip-802-1br	Displays 802.1BR header stripping information.
show packet-encap-processing strip-vn-tag	Displays VN-tag stripping information.

Appending source ports and timestamps

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 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# configure 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# configure 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# configure 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# configure 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# configure 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# configure 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
```

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# configure 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# 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*

ACLs under NPB

IP payload-length filtering using ACLs

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.
- If the IP payload length range is not configured, filters with **match-payload-len** are ignored and packets are not matched.

Payload-length configuration

You can update the IP payload length range on the global, slot, and PPCR level.

The steps common to all payload-length configurations are as follows:

1. Enable and configure the IP payload length for PPCR using the **ip match-payload-len slot** command.
2. In an ACL, enable the IP payload length check attribute.
3. Apply the ACL to an 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 configuring 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
```

Configuring 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
```

Configuring 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 configuring 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
```

Configuring 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
```

Configuring 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
```

IP payload-length-filtering show commands

There are show commands that display IP payload-length ranges, as listed in the following table.

TABLE 4 IP payload-length range show commands in the *Command Reference*

Command	Description
show ip match-payload-len	Displays the configuration for all PPCRs on which IPv4 payload length range is configured.
show ip match-payload-len interface ethernet	Displays the IPv4 payload length configuration on a specified Ethernet interface.
show ipv6 match-payload-len	Displays the configuration for all PPCRs on which IPv6 payload length range is configured.
show ipv6 match-payload-len interface ethernet	Displays the IPv6 payload length configuration on a specified Ethernet interface.

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.

Traffic flow is identified by port range; appropriate action is performed according to the 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 is not supported when ACL rules are provisioned with protocol numbers instead of protocol names.

Configuring IPv4 Extended access control list

```
device# configure 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# 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
```

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# 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
```