CONFIGURATION GUIDE

# Extreme SLX-OS Network Packet Broker Configuration Guide, 18r.2.00

## Supporting the ExtremeRouting SLX 9850 Router

*9035763*
*December 2018*

# Legal Notice

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

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

# Trademarks

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

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

For additional information on Extreme Networks trademarks, please see: www.extremenetworks.com/company/legal/trademarks

# Open Source Declarations

Some software files have been licensed under certain open source or third-party licenses. End-user license agreements and open source declarations can be found at: www.extremenetworks.com/support/policies/software-licensing

# Contents

# Preface



This section discusses the conventions used in this guide, ways to provide feedback, additional help, and other Extreme Networks® publications.

# Conventions

This section discusses the conventions used in this guide.

## 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. |

| Format | Description |
|---|---|
| `Courier font` | Identifies CLI output.<br>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.<br>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. |

# Documentation and Training

To find Extreme Networks product guides, visit our documentation pages at:

| | |
|---|---|
| Current Product Documentation | www.extremenetworks.com/documentation/ |
| Archived Documentation (for earlier versions and legacy products) | www.extremenetworks.com/support/documentation-archives/ |
| Release Notes | www.extremenetworks.com/support/release-notes |
| Hardware/Software Compatibility Matrices | https://www.extremenetworks.com/support/compatibility-matrices/ |
| White papers, data sheets, case studies, and other product resources | https://www.extremenetworks.com/resources/ |

## Training

Extreme Networks offers product training courses, both online and in person, as well as specialized certifications. For more information, visit www.extremenetworks.com/education/.

# Getting Help

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

| | |
|---|---|
| Extreme Portal | Search the GTAC (Global Technical Assistance Center) knowledge base, manage support cases and service contracts, download software, and obtain product licensing, training, and certifications. |
| The Hub | A forum for Extreme Networks customers to connect with one another, answer questions, and share ideas and feedback. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC. |
| Call GTAC | For immediate support: 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 |

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

## Subscribing to Service Notifications

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

1. Go to www.extremenetworks.com/support/service-notification-form.
2. Complete the form with your information (all fields are required).
3. Select the products for which you would like to receive notifications.

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

4. Click **Submit**.

# Providing Feedback to Us

Quality is our first concern at Extreme Networks, and we have made every effort to ensure the accuracy and completeness of this document. We are always striving to improve our documentation and help you work better, so we want to hear from you! We welcome all feedback but especially want to know about:

- Content errors or confusing or conflicting information.
- Ideas for improvements to our documentation so you can find the information you need faster.
- Broken links or usability issues.

If you would like to provide feedback to the Extreme Networks Information Development team, you can do so in two ways:

- Use our short online feedback form at https://www.extremenetworks.com/documentation-feedback/.
- Email us at documentation@extremenetworks.com.

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

# About This Document



# What's new in this document

This topic includes descriptions of changes in functionality for the current release.

**TABLE 1** Changes in Network Packet Broker (NPB) functionality for the current release

| Feature | Description | Described in |
|---|---|---|
| Route maps with UDAs | Enables precise redirection of L2 and L3 packets to a port or port-channel. | Policy-Based Routing under NPB on page 23 |
| Transparent Loopback | Enables a transmission-only link from a network device into an NPB device. | Transparent Loopback on page 33 |

# Supported hardware and software

In those instances in which procedures or parts of procedures documented here apply to some devices but not to others, this guide identifies exactly which devices are supported and which are not.

Although many different software and hardware configurations are tested and supported by Extreme Networks for the current release, documenting all possible configurations and scenarios is beyond the scope of this document.

The following hardware platforms are supported by this release:

- ExtremeRouting SLX 9850-4
- ExtremeRouting SLX 9850-8

To obtain information about other releases, refer to the documentation specific to that release.

## Interface module capabilities

The following table lists the supported capabilities for the following SLX 9850 interface modules:

- BR-SLX9850-10Gx72S-M
- BR-SLX9850-100Gx36CQ-M
- BR-SLX9850-10Gx72S-D
- BR-SLX9850-100Gx36CQ-D
- BR-SLX9850-100Gx12CQ-M

**TABLE 2** SLX 9850 interface modules capabilities

| Capability | Modular interface module |
|---|---|
| MPLS | Yes |
| Packet Buffer memory per interface module | 12GB (BR-SLX9850-10Gx72S-M)<br>36GB (BR-SLX9850-100Gx36CQ-M) |

**TABLE 2** SLX 9850 interface modules capabilities (continued)

| Capability | Modular interface module |
|---|---|
| | 8GB (BR-SLX9850-10Gx72S-D)<br>24GB (BR-SLX9850-100Gx36CQ-D)<br>8GB (BR-SLX9850-100Gx12CQ-M) |

# Basics of Network Packet Broker

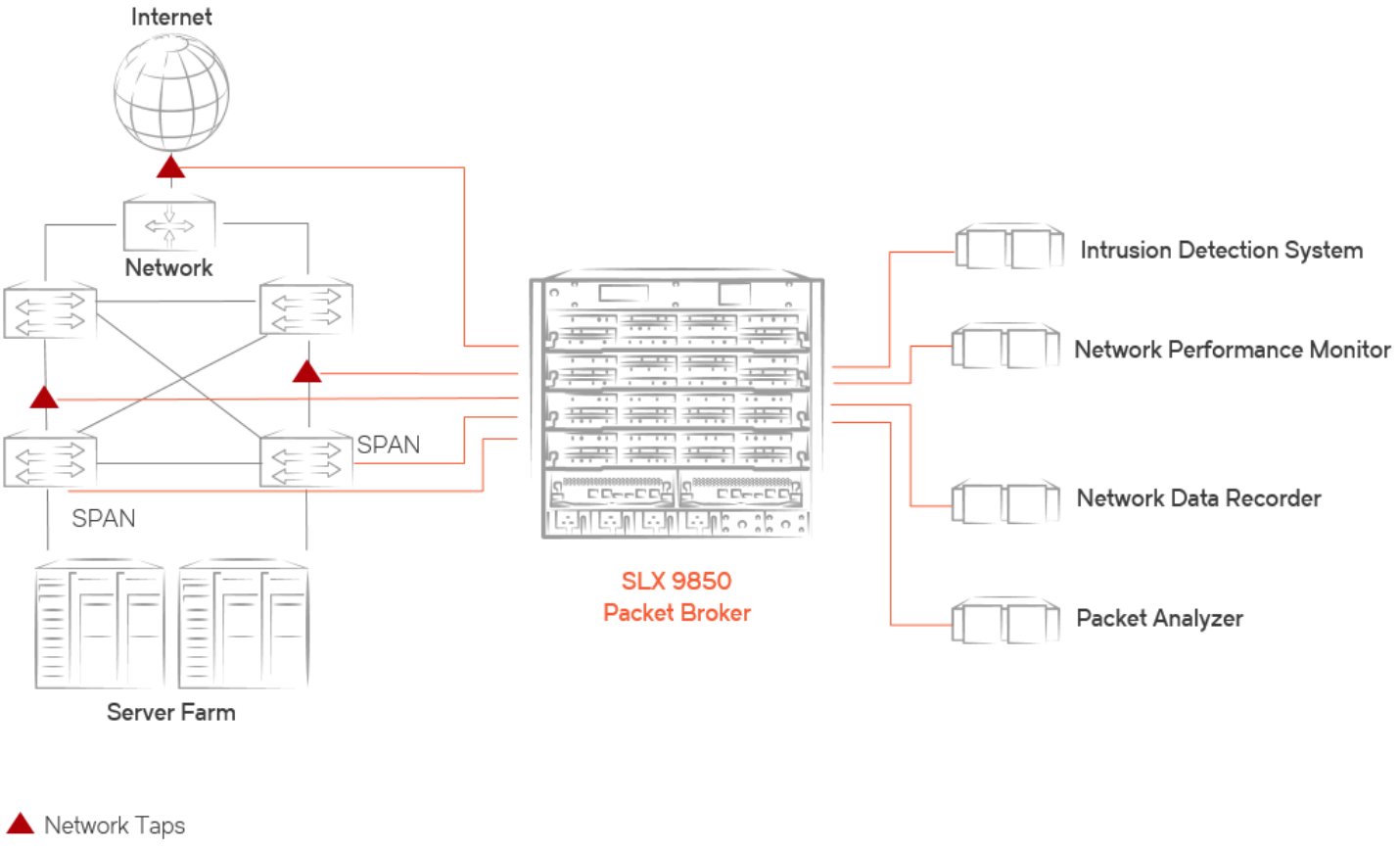


## Network Packet Broker overview

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

An NPB is the part of a network visibility infrastructure responsible for aggregating network traffic and directing it to visibility applications. The following diagram represents a typical NPB context in data centers and enterprises for compliance, intrusion detection, network-performance monitoring, data recording, packet analysis, flow manipulation, forensics, and so forth.

**FIGURE 1** A Network Packet Broker in context




Two technologies are supported for management of NPB ingress and egress traffic:

- OpenFlow, as described in *Extreme SLX-OS SDN Configuration Guide*. For a sample implementation flow, refer to Header-stripping implementation flow on page 30.
- Route-maps, as described in Policy-Based Routing under NPB on page 23.

## NPB features

SLX 9850 as an NPB supports the following features:

- Aggregation—the ability to aggregate traffic arriving from multiple TAPs or SPAN ports from upstream devices and direct the aggregated traffic to a single port or port-channel ("many to one").
- Replication—the ability to replicate network traffic to multiple ports and port-channels ("one to many").
- Load balancing—the ability to distribute network traffic among ports in a port-channel.
- ACL filtering:
  - Layer 2 and Layer 4 filtering—the ability to selectively direct network traffic based on fields of the Layer 2 and Layer 4 protocol headers.
  - User-defined ACLs (UDAs)—the ability to filter packets that Layer 2 and Layer 3 ACLs cannot filter.
- Route-map forwarding—the ability to redirect Layer 2 and Layer 3 packets to the desired physical or port-channel interface. By means of match statements, the following types of ACLs are supported:
  - Layer 2
  - Layer 3
  - User-defined (UDAs)
- Encapsulation-header stripping—the ability to remove headers not supported by some visibility applications. Supported headers:
  - 802.1BR
  - VN-Tag
  - VXLAN

# NPB configuration guidelines

Follow these guidelines when implementing Network Packet Broker (NPB):

- A system reboot is required when moving to the required **profile tcam npb-optimised-1** TCAM profile.
- Most Layer 2 and Layer 3 configurations are not supported in NPB mode. Although implementing them does not generate error messages, do not configure them in NPB mode.
- Although dynamic port-channel is not supported, static port-channel is supported. For details, refer to *Extreme SLX-OS Layer 2 Switching Configuration Guide*
- You can also use OpenFlow Group concepts to manage sets of physical ports. For details, refer to *Extreme SLX-OS SDN Configuration Guide*.

# NPB TCAM-profiles

The only ternary content-addressable memory (TCAM) profile supported for Network Packet Broker (NPB) is **npb-optimised-1**.

Under **npb-optimised-1**, you can modify the default TCAM allocations for IPv4 and IPv6 flows.

TCAM for OpenFlow—12288 (12K) entries—is shared between IPv4 and IPv6 flows.

## Default and maximum NPB TCAM-profile settings

For the **npb-optimised-1** profile, the following table displays the default and configurable TCAM scaling limits per ASIC.

NOTE
For details of which ports share ASICs, refer to the "Association of ports to ASICs" tables in the *SLX 9850 Router Technical Specifications*.

**TABLE 3** Default and maximum OpenFlow TCAM entries per ASIC

| Configuration | IPv4 | IPv6 | Total |
|---|---|---|---|
| Default | 6144 (6K) | 4096 (4K) | 10240 (10K) |
| Modified | 0K–12K | 0K–12K | 12288(12K) |

For the **npb-optimised-1** profile, the following table displays total OpenFlow TCAM entries available on an SLX 9850.

**TABLE 4** Total maximum OpenFlow TCAM entries

| Type | ASIC | Line card | F4 chassis | F8 chassis |
|---|---|---|---|---|
| IPv4 and IPv6 flows | 12288 (12K) | 24576 (24K) | 100352 (98K) | 200704 (196K) |

# Configuring NPB hardware profiles

Use this task to select the counter and TCAM profiles required for Network Packet Broker (NPB), with an option to modify TCAM profile scaling limits.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter **hardware** to access hardware configuration mode.

```
device(config)# hardware
```

3. Enter **profile counters counter-profile-4**.

```
device(config-hardware)# profile counters counter-profile-4
```

4. Enter **profile tcam npb-optimised-1**.

```
device(config-hardware)# profile tcam npb-optimised-1
%INFO: For network packet broker (NPB) stats functionality select appropriate counter profile.
%Warning: To activate the new profile config, please run 'copy running-config startup-config'
followed by 'reload system'.
```

NOTE
By default, the IPv4 scaling limit is 6K and the IPv6 scaling limit is 4K.

5. To modify the default scaling limits, enter the **profile tcam limit** command, specifying the limits you require.

```
device(config-hardware)# profile tcam limit l2l3v4-app 10240 v6-app 2048
```

NOTE
To modify the default user-defined ACL (UDA) scaling limits, you can also include the **flex-acl** option.

6. Enter **end** to return to privileged EXEC mode.

```
device(config-hardware)# end
```

7. Enter **copy running-config startup-config**.

```
device# copy running-config startup-config
```

8. Enter **reload system** to reboot the system, effecting the TCAM profile change.

```
device# reload system
Warning: This operation will cause the chassis to reboot and requires all existing telnet,
secure telnet and SSH sessions to be restarted.
Unsaved configuration will be lost.
Please run `copy running-config startup-config` to save the current configuration if not done
already.
Are you sure you want to reboot the chassis [y/n]?
```

9. Press **y** and then **Enter**.

# Modifying NPB TCAM-profile scaling limits

Use this task to modify the IPv4, IPv6, and flex-ACL (UDA) scaling limits of the TCAM profile required for Network Packet Broker (NPB).

NOTE
This feature is available is available only if you first set the TCAM profile to **npb-optimised-1**, as described in Configuring NPB hardware profiles on page 13.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter **hardware** to access hardware configuration mode.

```
device(config)# hardware
```

3. Enter the **profile tcam limit l2l3v4-app** command, specifying the limits you require.

```
device(config-hardware)# profile tcam limit l2l3v4-app 10240 v6-app 2048
```

4. Enter **end** to return to privileged EXEC mode.

```
device(config-hardware)# end
```

5. Enter **copy running-config startup-config**.

```
device# copy running-config startup-config
```

6. Enter **reload system** to reboot the system, effecting the NPB TCAM-profile scaling change.

```
device# reload system
Warning: This operation will cause the chassis to reboot and requires all existing telnet,
secure telnet and SSH sessions to be restarted.
Unsaved configuration will be lost.
Please run `copy running-config startup-config` to save the current configuration if not done
already.
Are you sure you want to reboot the chassis [y/n]?
```

7. Press **y** and then **Enter**.

# ACLs and UDAs under NPB



## Stanza and ACL permit and deny keywords

Both route-map stanzas and access-control lists (ACLs) have **permit** and **deny** keywords.

> NOTE
> In this context, "ACL" includes standard, extended, and user-defined ACLs (UDAs).

In NPB mode, ACLs within route-maps exclude certain traffic flows from set statements.

Route-maps contain **match { ip | ipv4 } address acl** statements. (NPB route-maps can also contain **match uda** statements.) However, permit and deny rules in ACLs applied to route maps function differently than rules in the security ACLs discussed in the *Extreme SLX-OS Security Configuration Guide* :

- In security ACLs, permit rules allow packets and deny rules drop packets.
- In ACLs applied to route-maps, permit and deny rules specify criteria for route-map decisions.

The following table describes the interactions between route-map permit and deny stanzas; and permit and deny rules in ACLs applied to those stanzas by **match { ip | ipv6 } address acl** or **match uda** statements.

**TABLE 5** Route-map stanza and ACL permit and deny interactions

| Stanza | ACL rule | Resulting TCAM action |
|---|---|---|
| Permit | Permit | The **set** statement or statements are applied. |
| Permit | Deny | Packets that match a **deny** keyword are denied from using the stanza **set** statement. The packet is routed as normal. |
| Deny | Permit | No action is taken; the packet is routed as normal. |
| Deny | Deny | No action is taken; the packet is routed as normal. |

## Creating ACLs for NPB

For NPB traffic aggregation and traffic replication, an access-control list (ACL) or user-defined ACL (UDA)—included in a route-map—is required.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter the **{ ip | ipv6 } access-list** command to create an ACL.

```
device(config)# ip access-list standard aclNPB_01
```

3. Create one or more permit or deny rules.

```
device(conf-ipacl-std)# permit host 192.1.1.1 count
```

NOTE
A deny rule specifies that a matching packet is denied from using the **set** statement.

# UDAs

User-defined ACLs (UDAs) examine packet fields at specified offsets, applying permit and deny rules.

## Basics of user-defined ACLs (UDAs)

UDAs offer greater flexibility than other ACLs in defining deny and permit rules. This flexibility is required for certain Network Packet Broker (NPB) scenarios.

UDAs are supported for NPB:

- Applied to physical and port-channel interfaces with the **uda access-group** command, as described in this section.
- For policy-based routing (PBR), as described in "Policy-Based Routing under NPB."

### *UDA configuration guidelines*

The following general guidelines apply to UDA configuration:

- UDAs are supported only if the npb-optimised-1 profile is enabled.
- You can apply UDAs only on physical and port-channel interfaces, but not on virtual interfaces.
- UDAs are not supported as receive ACLs (rACLS).
- The processing priority of UDAs is lower than other ACLs: OpenFlow > rACLs > PBR > Layer 3 ACLs > Layer 2 ACLs > UDAs.
- You cannot apply more than one UDA profile to a group of interfaces that share an ASIC. For details, refer to the "Association of ports to ASICs" tables in the *SLX 9850 Router Technical Specifications*.
- TCAM sharing is not supported for UDAs.

### *UDA-profile guidelines*

For the **uda-key profile** > **uda-offsets** > **packet-start** option, the offset begins with the Ethernet header.

The following **uda-key profile** guidelines apply to **uda-offsets**> **first-header**, **second-header**, **third-header**, or **fourth-header**:

- The Ethernet header (including VLAN header) is not considered.
- The profile works equally for tagged and untagged packets. For example, for both of the following packets, **uda-offsets** considers <IPv6> as the first header and <UDP> as the second header:
  - (Untagged) <MAC><IPV6><UDP>
  - (Tagged) <MAC><VLAN><IPV6><UDP>
- To match fields inside a tunnel header (for example, GTP, VXLAN, or GRE), define that header as the offset base.

### *UDA hardware support and scaling*

UDAs are supported on all interface-module line cards available for the SLX 9850 device.

For UDA and UDA-rule scaling limits, refer to the "ACL and rule limits" topic in the *Extreme SLX-OS Security Configuration Guide*.

## UDA-profile design

Before you create a UDA profile, you need to analyze header composition.

This analysis is especially important for tunnel-protocol elements like GTP, VXLAN, and GRE, which are not parsed. Before you begin, examine the following *Extreme SLX-OS Command Reference* topics:

- **uda-key profile**
- **uda-offsets**

**TABLE 6** UDA-profile example

| Outer header | Inner header | Field | Field-value | Base and offset |
|---|---|---|---|---|
| Ethernet<br>VLAN | | | | |
| IPv6 (1st) | | | | |
| UDP (2nd) | | Outer UDP dest. port | 0x0868 | `second-header 2` |
| GTP (3rd) | IPv6 | Inner IPv6 | 0x11 | `third-header 14` |
| | UDP | Inner UDP dest. port | 0x13c4 | `third-header 50` |
| | | Data | 0x4D455353 | `third-header 56` |

The following topics implement this design on a physical interface:


- Configuring a UDA profile on page 17
- Applying a UDA profile on a physical interface on page 18
- Creating a UDA on page 18
- Applying a UDA on a physical interface on page 19


## Configuring a UDA profile

Use this task to create a UDA profile and define its offsets.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter the **uda-key profile** command to create a UDA profile.

```
device(config)# uda-key profile uda_gtp
```

3. Enter the **uda-offsets** command to configure the profile.

```
device(config-uda-key)# uda-offsets second-header 2 third-header 14 third-header 50 third-header 56
```

The following example uses the second header (UDP) as the offset base and specifies 49 as the offset.

```
device# configure terminal
device(config)# uda-key profile uda-prof_02
device(config-uda-key)# uda-offsets second-header 49 ignore ignore ignore
```

# Applying a UDA profile on a physical interface

Use this task to apply a user-defined ACL (UDA)-profile on an Ethernet interface.

1. Enter **configure terminal** to access global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter the **interface ethernet** command—specifying the slot and port—to access interface configuration mode.

   ```
   device(config)# interface ethernet 1/2
   ```

3. Enter the **uda-profile-apply** command—specifying the UDA profile—to apply the UDA profile to the interface.

   ```
   device(conf-if-eth-1/2)# uda-profile-apply uda_gtp
   ```

# Applying a UDA profile on a port-channel interface

Use this task to apply a user-defined ACL (UDA)-profile on a port-channel interface.

1. Enter the **configure terminal** command to access global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter the **interface port-channel** command, specifying the port-channel number.

   ```
   device(config)# interface port-channel 10
   ```

3. Enter the **uda-profile-apply** command, specifying the UDA profile.

   ```
   device(config-Port-channel-10)# uda-profile-apply uda-prof_02
   ```

# Creating a UDA

Use this task to create a user-defined ACL (UDA) and define permit and deny rules within.

1. Enter **configure terminal** to access global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter the **uda access-list** command to create the access list.

   ```
   device(config)# uda access-list extended uda_gtp
   ```

3. Enter rules, specifying the needed parameters.

   ```
   device(conf-udaacl-ext)# permit 0x08680000 0xffff0000 0x11000000 0xff000000 0x13c40000
           0xffff0000 0x4d455353 0xffffffff count
   ```

The following example creates a UDA and defines a permit rule, with statistics enabled for the rule.

```
device# configure terminal
device(config)# uda access-list extended uda_01
device(conf-udaacl-ext)# permit 0x00001111 0x0000ffff 0x00002222 0x0000ffff 0x00003333 0x0000ffff
0x00004444 0x0000ffff count
```

# Applying a UDA on a physical interface

Use this task to apply a user-defined ACL (UDA) on a physical interface.

1. Enter **configure terminal** to access global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter the **interface ethernet** command—specifying the slot and port—to access interface configuration mode.

   ```
   device(config)# interface ethernet 5/2
   ```

3. Enter the **uda access-group** command—specifying the UDA and ingress direction—to apply the UDA on the interface.

   ```
   device(conf-if-eth-5/2)# uda access-group uda_gtp in
   ```

# Applying a UDA on a port-channel interface

Use this task to apply a user-defined ACL (UDA) on a port-channel interface.

1. Enter the **configure terminal** command to access global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter the **interface port-channel** command, specifying the port-channel number.

   ```
   device(config)# interface port-channel 10
   ```

3. Enter the **uda access-group** command, specifying the specifying the UDA and the ingress direction.

   ```
   device(config-Port-channel-10)# uda access-group uda_02 in
   ```

# UDA-on-interface example

This UDA-on-interface use case supports an SLX 9850 Network Packet Broker (NPB) for a wireless-communication enterprise.

> **NOTE**
> For UDA matching in route-maps, refer to "Configuring route-maps with UDAs."

The goal of this use case is to enable ingress traffic with UDP header = ESP (Encapsulating Security Protocol) to be forwarded to monitoring and analytics tools.

Packet details

- Ethernet Type: 0x8100 (802.1Q)
- 802.1Q Type: 0x0800 (IPv4)
- IP Protocol: 17 (UDP)

- UDP: D. Port 2152

UDP payload

- GPRS Message Type: 0xff (T-PDU)
- IP Protocol: 47 (GRE)
- GRE Type: 0x0800 (IP)
- IP Protocol: 0x32 (ESP)

Match: ESP Packets (encapsulating sec. pay)

- IP Protocol Type 50
- 1 byte
- 0x32 (ESP)
- Byte 87 (from start of packet)

## *Task 1: Configure a UDA profile*

For this UDA profile, the second-header location (UDP) is specified as the offset base and the offset value is 49.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter the **uda-key profile** command to create a UDA profile.

```
device(config)# uda-key profile uda-prof_03
```

3. Enter the **uda-offsets** command to configure the profile.

```
device(config-uda-key)# uda-offsets second-header 49 ignore ignore ignore
```

NOTE
UDP is the second header. The offset value is 49 from the start of the UDP header.

## *Task 2: Create a UDA ACL*

Create a UDA ACL with the needed permit rule.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter the **uda access-list** command to create the access list.

```
device(config)# uda access-list extended uda_03
```

3. Enter a permit rule to validate 0x32 (Encapsulating Security Protocol [ESP]) at the 49th position from the UDP header.

```
permit 0x32 0xff 0x0 0x0 0x0 0x0 0x0 0x0
```

### *Task 3: Apply the UDA profile and the UDA on an interface*

Apply the user-defined ACL (UDA)-profile and the UDA on each relevant interface.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter the **interface ethernet** command—specifying the slot and port—to access interface configuration mode.

```
device(config)# interface ethernet 1/2
```

3. Enter the **uda-profile-apply** command—specifying the UDA profile—to apply the UDA profile to the interface.

```
device(conf-if-eth-1/2)# uda-profile-apply uda-prof_03
```

4. Enter the **uda access-group** command—specifying the UDA and ingress direction—to apply the UDA on the interface.

```
device(conf-if-eth-5/2)# uda access-group uda_03 in
```

## UDA show and clear commands

There is a full range of UDA show and clear commands, listed here with descriptions.

**TABLE 7** UDA show commands in the *Command Reference*

| Command | Description |
|---|---|
| **show access-list** | For an ACL type and inbound/outbound direction, displays ACL information. You can show information for a specified ACL or only for that ACL on a specific interface. You can also display information for all ACLs bound to a specific interface. |
| **show running-config uda access-list** | Displays a list of user-defined ACLs (UDAs) defined on the device—or a specific UDA—including the rules they contain. |
| **show running-config uda-key profile** | Displays a list of user-defined ACL (UDA)-profiles defined on the device—or a specific UDA profile. |
| **show statistics access-list** | For an ACL type and inbound/outbound direction, displays statistical information—for ACL rules that include the **count** keyword. You can show statistics for a specific ACL or only for that ACL on a specific interface. You can also display statistical information for all ACLs bound to a specific interface. |

**TABLE 8** UDA clear commands in the *Command Reference*

| Command | Description |
|---|---|
| **clear counters access-list** | For an ACL type and inbound/outbound direction, clears ACL statistical information. You can clear all statistics for a specific ACL or only for that ACL on a specific interface. You can also clear statistical information for all ACLs bound to a specific interface. |

# Policy-Based Routing under NPB



## Route-map overview

A route map is a container for one or more numbered permit or deny stanzas; each stanza contains a sequence of statements.

Evaluation of route maps consists of a list scan, from the lowest-numbered stanza to the highest-numbered stanza. Within each stanza, statements are evaluated in order. The following technologies are some that use route maps:

- BGP and OSPF protocols. For details, refer to the *Extreme SLX-OS Layer 3 Routing Configuration Guide*.
- Policy-based routing (PBR) under Network Packet Broker (NPB), as described in the current guide.
- Non-NPB PBR. For details, refer to the *Extreme SLX-OS Security Configuration Guide*.

Route-map syntax and evaluation vary with the technology. However, all route-map implementations include a "match" statement. Upon a match, the actions specified in the match statement and in the remaining statements of that stanza are implemented. The list scan ends without examining higher-numbered stanzas.

Route-map "set" statements are also supported in permit stanzas for the mentioned technologies. Upon a match, set statements perform an action on the matched traffic (generally, forwarding the traffic to a specified interface). The exact action varies with the technology and the set command. Under NPB, multiple set statements are handled as follows:

- If the specified target interface is UP, traffic is forwarded to that interface.
- If that interface is not UP, succeeding set statements are evaluated.
- If a down interface specified in a preceding set statement later goes UP, traffic is instead routed to that interface.
- If none of the interfaces specified in the set statements are UP, the traffic is forwarded according to the routing table.

The following table compares NPB and non-NPB route-map implementations.

**TABLE 9** Route-map comparison

| Feature | PBR under NPB (matching ACLs) | PBR under NPB (matching UDAs) | PBR (non-NPB) |
|---|---|---|---|
| Match statements | One **match { ip \| ipv6 } address acl** statement | One **match uda** statement | One **match { ip \| ipv6 } address acl** statement |
| Set statements (supported only in permit stanzas) | 0, 1, or multiple **set interface**, **set { ip \| ipv6 } next-hop**, or **set { ip \| ipv6 } interface null0** statements | 0, 1, or multiple **set interface** or **set uda interface null0** statements | 0, 1, or multiple **set interface**, **set { ip \| ipv6 } next-hop**, or **set { ip \| ipv6 } interface null0** statements |
| Apply with | **{ ip \| ipv6 } policy route-map** | **uda policy route-map** | **{ ip \| ipv6 } policy route-map** |

# PBR-under-NPB configuration guidelines

The following guidelines apply to configuration of policy-based routing (PBR) under NPB:

- Match statements are supported for both Layer 3 ACLs and user-defined ACLs (UDAs).
- You can apply route-maps to physical interfaces.
- For physical interfaces included in port-channels (LAGs), you can apply route-maps only at port-channel level.
- Egress ports specified in set statements can be either physical interfaces or port-channels.
- Route-maps support up to 1024 stanzas.
- Each stanza supports up to 128 **set interface** statements.
- Each stanza supports only one **match** statement.
- There is no need to reapply a route-map to an interface following changes:
  - To the route-map.
  - To the ACL used as a match in the route-map.
- There is no requirement for consistency— specifying IPv4 ACLs, IPv6 ACLs or UDAs—among the match statements in multiple stanzas of a route-map. However:
  - If you apply the route-map to an interface with the **uda policy route-map** command, the only stanzas enabled are those that contain a **match uda** statement.
  - If you apply the route-map to an interface with the **ip policy route-map** command, the only stanzas enabled are those that contain a **match ip address acl** statement.
  - If you apply the route-map to an interface with the **ipv6 policy route-map** command, the only stanzas enabled are those that contain a **match ipv6 address acl** statement.
- On Layer 3 ports, route-maps applied with either **uda policy route-map** or with **{ ip | ipv6 } policy route-map** are supported.
- On Layer 2 ports, only route-maps applied with **uda policy route-map** are supported.
- On Layer 2 ports, to handle all kinds of tagged and untagged packets, make sure to include the **switchport trunk allowed vlan all** command.
- CAM sharing is not supported for UDAs.

# Configuring route-maps with ACLs

For relevant use-cases, you can configure and apply a route-map that references a standard or extended ACL.

NOTE
The example used in this task is for VXLAN header-stripping.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Configure the egress interface with the appropriate IP address.

```
device(config)# interface Ethernet 8/16
device(conf-if-eth-8/16)# ip address 20.20.20.15/24
device(conf-if-eth-8/16)# no shut
device(conf-if-eth-8/16)# exit
```

3. Create a standard or extended ACL and define needed rules.

```
device(config)# ip access-list extended ip_ext_acl_01
device(conf-ipacl-ext)# seq 20 permit ip any any count
device(conf-ipacl-ext)# exit
```

4. Configure a route-map, specifying the ACL you created.

```
device(config)# route-map rm_ACL_01 permit 1
device(config-route-map-rm_ACL_01/permit/1)# match ip address acl ip_ext_acl_01
device(config-route-map-rm_ACL_01_ACL_01/permit/1)# set ip next-hop 20.20.20.20
device(config-route-map-rm_ACL_01_ACL_01/permit/1)# exit
```

5. Apply the route-map on the ingress interface, specifying the IP address and the relevant header-stripping command.

```
device(config)# interface Ethernet 8/15
device(conf-if-eth-8/15)# ip address 10.10.10.15/24
device(conf-if-eth-8/15)# no shutdown
device(conf-if-eth-8/15)# ip policy route-map rm_ACL_01
device(conf-if-eth-8/15)# strip-vxlan
device(conf-if-eth-8/15)# exit
```

# Configuring route-maps with UDAs

For relevant use-cases, you can configure and apply a route-map that references a user-defined ACL (UDA).

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Configure a UDA profile, as described in Configuring a UDA profile on page 17.

```
device(config)# uda-key profile uda-prof_02
device(config-uda-key)# uda-offsets second-header 49 ignore ignore ignore
device(config-uda-key)# exit
```

3. Create a UDA, as described in Creating a UDA on page 18.

```
device(config)# uda access-list extended uda_01
device(conf-udaacl-ext)# permit 0x00001111 0x0000ffff 0x00002222 0x0000ffff 0x00003333 0x0000ffff
0x00004444 0x0000ffff count
device(conf-udaacl-ext)# exit
```

4. Create a route-map, specifying the UDA that you created.

```
device(config)# route-map rm_UDA_01 permit 1
device(config-route-map-rm_UDA_01/permit/1)# match uda uda_01
device(config-route-map-rm_UDA_01/permit/1)# set interface ethernet 1/5
device(config-route-map-rm_UDA_01/permit/1)# set interface ethernet  1/6
device(config-route-map-rm_UDA_01/permit/1)# set uda interface null0
device(config-route-map-rm_UDA_01/permit/1)# exit
```

5. (If needed, but only at physical-interface level) Implement transparent loopback on the interface.

```
device(config)# interface ethernet 1/2
device(conf-if-eth-1/2)# loopback phy
```

6. Apply the UDA profile and the route-map to the needed physical or port-channel interface.

```
device(conf-if-eth-1/2)# uda-profile-apply uda-prof_01
device(conf-if-eth-1/2)# switchport
device(conf-if-eth-1/2)# switchport mode trunk
device(conf-if-eth-1/2)# switchport trunk allowed vlan all
device(conf-if-eth-1/2)# no switchport trunk tag native-vlan
device(conf-if-eth-1/2)# uda policy route-map rm_UDA_01
device(conf-if-eth-1/2)# no shutdown
```

# Route-map show commands

There are several show commands that display route-map information, as listed in the following table.

**TABLE 10** Route-map show commands in the *Command Reference*

| Command | Description |
|---|---|
| **show route-map** | Displays details of all route-maps, a specific route-map, or of the route-map applied to a specific interface. |
| **show running-config route-map** | Displays the running configuration of all route-maps, a specific route-map, or of the route-map applied to a specific interface. You can filter the output by elements within a route-map. |

# Header Modification



## Header-modification overview

Protocol headers help packets reach their destinations, but are not needed by the security and monitoring tools to which NPB forwards traffic.

Tagging and encapsulation techniques have long been a part of networking. However, the recent adoption of new encapsulation protocols—such as VXLAN, VN-Tag, and 802.1BR—can create visibility blind spots, because some visibility applications were not designed to interpret these new protocols.

By removing the encapsulation header, the NPB removes the burden of interpreting the various encapsulation protocols from the visibility applications. Therefore, the header-stripping feature enables network operators to deploy new encapsulation protocols without interfering with proper functioning of previously deployed visibility applications. Other header stripping benefits include:

- Reduced packet overhead and better visibility-application bandwidth utilization.
- Preservation of standard filtering and load balancing capabilities.

For the current SLX-OS version, 802.1BR, VN-Tag, and VXLAN header stripping are the only types of header modification supported.

> NOTE
> Although NPB is supported only on SLX 9850, header stripping is supported also on SLX 9540.

## Header-stripping configuration guidelines

Follow these guidelines when implementing header stripping.

- The primary use case for header stripping is Network Packet Broker (NPB).
- Header stripping applies to both IPv4 and IPv6 packets.

802.1BR header-stripping removes only 802.1BR headers. VN-Tag header-stripping removes only the VN-Tag headers. VXLAN header-stripping removes multiple headers, as described in the following table:

**TABLE 11** Header-stripping scope

| Type | Headers stripped |
|---|---|
| 802.1BR | 802.1BR headers |
| VN-Tag | VN-Tag headers |
| VXLAN | Layer 2, Layer 3, UDP, and VXLAN headers |

Among the supported types of header stripping, you can never enable more than one type on a specific interface. However, if one of two applied types is VXLAN header-stripping, you can apply one other type on other slot interfaces. These limitations are summarized in the following table.

**TABLE 12** Header-stripping compatibility matrix

| Type | 802.1BR | VN-Tag | VXLAN |
|---|---|---|---|
| 802.1BR | — | Not on the same slot. | Not on the same interface. |
| VN-Tag | Not on the same slot. | — | Not on the same interface. |
| VXLAN | Not on the same interface. | Not on the same interface. | — |

After you enable header stripping on an interface, there are cases for which you need to reboot the linecard:

```
device# show packet-encap-processing
Total number of packet-encap-processing interfaces: 2
------------------------------------------------------------
Port       Link     Encapsulation        Status
------------------------------------------------------------
eth1/1     Up       VN-Tag Stripping     Inactive
eth1/2     Up       VN-Tag Stripping     Inactive

device# power-off linecard 1
device# power-on linecard 1
```

**TABLE 13** Linecard rebooting following header-stripping enablement

| Type | Rebooting |
|---|---|
| VN-Tag | Rebooting is required only after transition from 802.1BR header-stripping. |
| VXLAN | Rebooting is not required. |
| 802.1BR | Rebooting is required only after transition from VN-Tag header-stripping. |

**NOTE**
If linecard rebooting is required for a header-stripping type, one rebooting is sufficient for all linecard interfaces enabled.

The traffic-redirection technology varies among the types of header stripping.

**TABLE 14** Traffic-redirection technologies

| 802.1BR | VN-Tag | VXLAN |
|---|---|---|
| OpenFlow | OpenFlow | Policy-based routing (PBR) |

# Configuring 802.1BR header stripping

Use this task to enable or disable 802.1BR header stripping on an Ethernet interface.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter the **interface ethernet** command to access Ethernet interface configuration mode.

```
device(config)# interface ethernet 1/2
```

3. To enable 802.1BR header stripping on that interface, enter the **strip-802-1br** command.

```
device(conf-if-eth-1/2)# strip-802-1br
```

4. To disable 802.1BR header stripping on that interface, enter the **no strip-802-1br** command.

```
device(conf-if-eth-1/2)# no strip-802-1br
```

5. NOTE
   (If you are moving from VN-Tag header-stripping) For each linecard, you need to perform the following step only for the first interface enabled for 802.1BR header-stripping:

   Enter the **power-off linecard** command and then the **power-on linecard** command.

```
device# power-off linecard 1
Linecard 1 is being powered-off
device# power-on linecard 1
Linecard 1 is being powered-on
```

The following example enables 802.1BR header stripping on an interface.

```
device# configure terminal
device(config)# interface ethernet 1/2
device(conf-if-eth-1/2)# strip-802-1br
```

# Configuring VN-Tag header stripping

Use this task to enable or disable Virtual NIC (VN)-Tag header stripping on an Ethernet interface.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter the **interface ethernet** command to access Ethernet interface configuration mode.

```
device(config)# interface ethernet 1/2
```

3. To enable VN-Tag header stripping on that interface, enter the **strip-vn-tag** command.

```
device(conf-if-eth-1/2)# strip-vn-tag
%WARNING: Please reboot the linecard <> for stripping configuration to take effect.
```

4. To disable VN-Tag header stripping on that interface, enter the **no strip-vn-tag** command.

```
device(conf-if-eth-1/2)# no strip-vn-tag
```

5. Enter **end** to return to privileged EXEC mode.

```
device(config-hardware)# end
```

6. NOTE
   For each linecard, you need to perform the following step only for the first interface enabled for VN-Tag header-stripping:

   Enter the **power-off linecard** command and then the **power-on linecard** command.

```
device# power-off linecard 1
Linecard 1 is being powered-off
device# power-on linecard 1
Linecard 1 is being powered-on
```

The following example enables VN-Tag header stripping on an interface.

```
device# configure terminal
device(config)# interface ethernet 1/2
device(conf-if-eth-1/2)# strip-vn-tag
    %WARNING: Please reboot the linecard <> for stripping configuration to take effect.
device(config-hardware)# end
device# power-off linecard 1
Linecard 1 is being powered-off
device# power-on linecard 1
Linecard 1 is being powered-on
```

# Configuring VXLAN header stripping

Use this task to enable or disable Virtual Extensible LAN (VXLAN) header stripping on an Ethernet interface.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter the **interface ethernet** command to access Ethernet interface configuration mode.

```
device(config)# interface ethernet 1/2
```

3. To enable VXLAN header stripping on that interface, enter the **strip-vxlan** command.

```
device(conf-if-eth-1/2)# strip-vxlan
```

4. To disable VXLAN header stripping on that interface, enter the **no strip-vxlan** command.

```
device(conf-if-eth-1/2)# no strip-vxlan
```

The following example enables VXLAN header stripping on an interface.

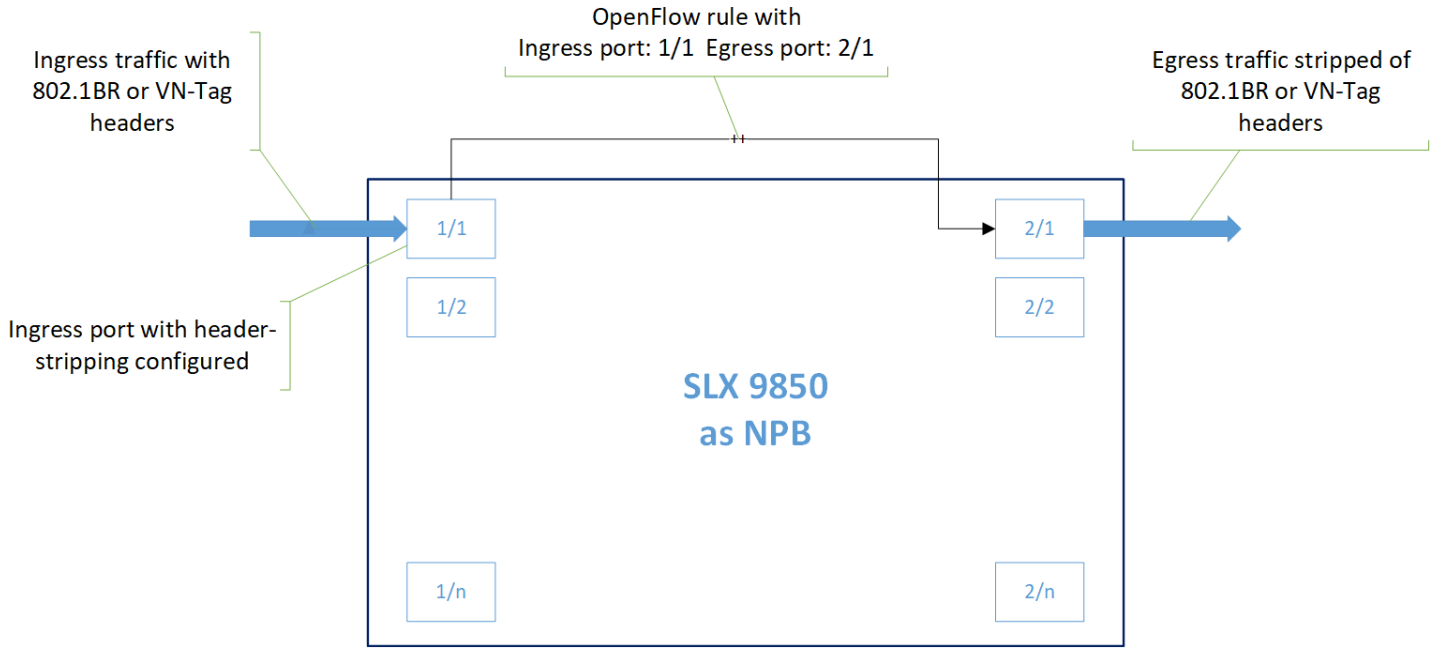
```
device# configure terminal
device(config)# interface ethernet 1/2
device(conf-if-eth-1/2)# strip-vxlan
```

# Header-stripping implementation flow

This NPB implementation-flow includes 802.1BR or VN-Tag header stripping.

NOTE
For VXLAN header-stripping, use policy-based routing (PBR) to redirect traffic. For details, refer to Configuring route-maps with ACLs on page 24.

**FIGURE 2** NPB header-stripping flow




Port 1/1 is configured to strip one of the supported header types from incoming traffic and to forward the modified traffic—under OpenFlow—to port 2/1. Port 2/1 is configured to forward the modified traffic to the analytics tools.

1. Make sure that the **npb-optimised-1** TCAM profile is configured on the device, as described in Configuring NPB hardware profiles on page 13.
2. Enable OpenFlow on the ingress interface.

```
device# configure terminal
device(config)# interface ethernet 1/1
device(conf-if-eth-1/1)# openflow enable layer3
```

3. Enable the header-stripping feature on the ingress interface.

```
device(conf-if-eth-1/1)# strip-802-1br
device(conf-if-eth-1/1)# exit
```

4. Enable OpenFlow on the egress interface.

```
device(config)# interface ethernet 2/1
device(conf-if-eth-2/1)# openflow enable layer3
device(conf-if-eth-2/1)# end
```

5. Run the **show openflow interface** command to get needed details.

```
device# show openflow interface
Total number of Openflow interfaces: 2

Port       Link    Port-State   Speed   MAC              OF-Port-ID  Mode
Eth 1/1    Down    Forward      10G     748e.f88f.9a45   4           l3
Eth 2/1    Down    Forward      10G     748e.f88f.9a46   8           l3
```

6. Program the OpenFlow rule from the OpenFlow controller.

```
$ ovs-ofctl add-flow tcp:10.24.4.180:6633 "in_port=4 action=output:8"  --protocol Openflow13 –v
```

# Transparent Loopback



# Basics of transparent loopback

Transparent loopback enables a transmission-only link from a network device into an NPB device.

By default, if you want to connect device A to device B, you need to implement egress and ingress cables from A to B. If there is only an egress cable but no ingress cable, the link is down.

With transparent loopback configured on a device A interface, traffic both transmits over the single cable to device B ("transparent") and loops back as ingress traffic ("loopback"). In spite of the fact that there is no physical cable from B back to A, the link stays up.

Transparent loopback enables the following NPB scenario:

1. Connect an egress cable from network device A to the NPB device (device B)—without connecting an ingress cable.
2. Implement transparent loopback on the device A interface.
3. On the device A interface, apply an ACL that drops the loopbacked traffic.

This scenario ensures traffic flow from device A to the NPB with no option for traffic from the NPB to device A.

## Transparent-loopback configuration guidelines

Follow these guidelines when implementing transparent loopback.

- Implement transparent loopback only at physical-interface level.
- Although you cannot implement transparent loopback at LAG-level, LAGs that contain one or more transparent-loopback interfaces are supported, as follows:
  - Static LAG is fully supported.
  - Dynamic LAG does not currently support transparent loopback.
  - If needed, you can configure transparent loopback on all of the physical member ports of a LAG.

# Configuring transparent loopback

Use this task to configure transparent loopback on a physical interface.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter the **interface ethernet** command to access Ethernet interface configuration mode.

```
device(config)# interface ethernet 1/2
```

3. Enter the **loopback phy** command to implement transparent loopback on that interface.

```
device(conf-if-eth-1/2)# loopback phy
```

# NPB show commands

There are several show commands that display Network Packet Broker (NPB) information, as listed in the following table.

**TABLE 15** NPB show commands in the *Command Reference*

| Command | Description |
|---|---|
| **show interface ethernet** | Displays the detailed configuration and capabilities of a specific interface, including transparent-loopback status. |
| **show packet-encap-processing** | Displays information about interfaces on which header processing is enabled. |
| **show running-config interface ethernet** | Displays the contents of the running configuration—including header-stripping settings—on a specific Ethernet interface. |