

Header Stripping for Nexus Data Broker

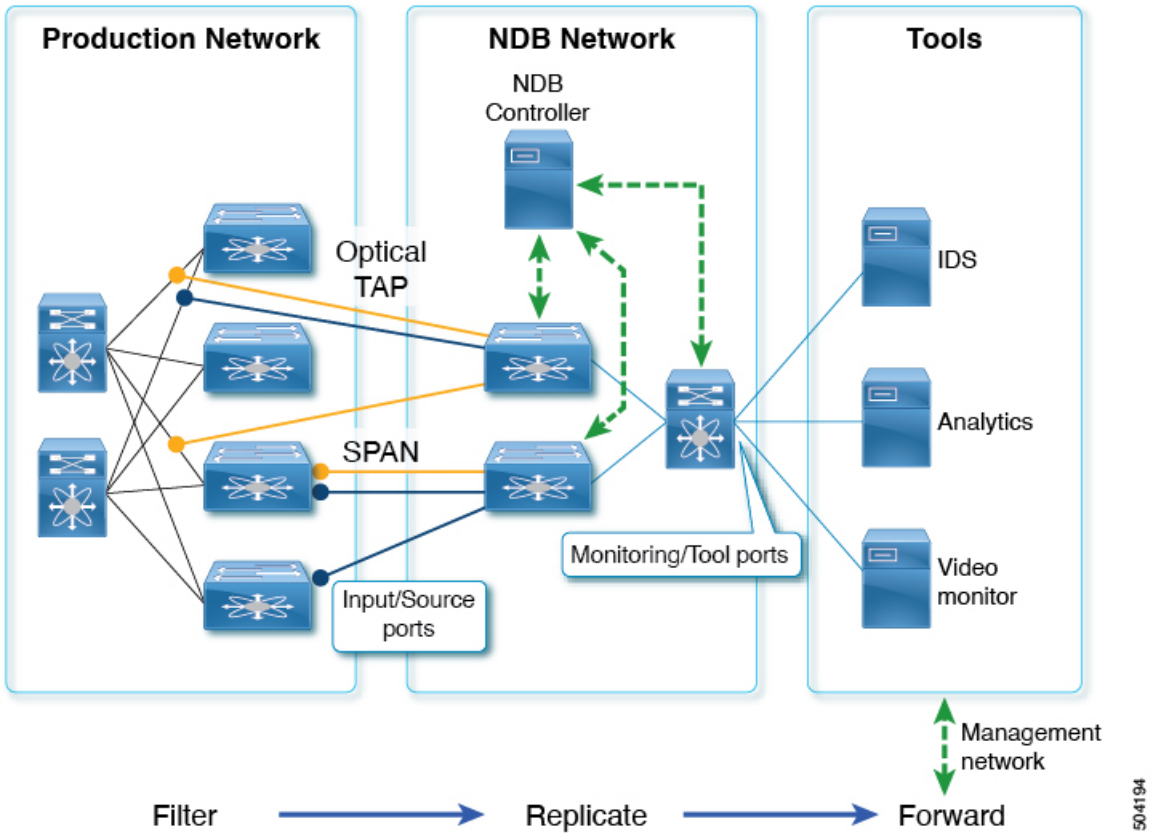
- Overview, on page 1
- Guidelines and Limitations, on page 3
- VXLAN and iVXLAN Header Stripping, on page 4
- ERSPAN Header Stripping, on page 8
- GRE Header Stripping, on page 11
- MPLS Header Stripping, on page 15

Overview

Cisco Nexus Data Broker (NDB) is a solution that builds scalable packet broker network solutions, providing a software-defined approach for monitoring both out-of-band and inline network traffic using Cisco Nexus Dashboard Data Broker controller software and Cisco Nexus switches.

- NDB enables packet monitoring for performance monitoring, intrusion detection, and compliance checking.
- Header stripping is performed on NDB switches, allowing traffic to be filtered, replicated, and stripped of headers before being forwarded to monitoring tools.
- Input/source ports are where header stripping occurs, and monitoring/tool ports connect directly to the tools.

Figure 1: NDB Centralized Deployment Model



Header Stripping and Feature Benefits

The reasons for removing the header are as follows:

- Some monitoring tools do not understand an encapsulated packet.
- Presence of an additional header skews the analytics data.
- Addition of a header adds to the packet size, hampering the optimization of the amount of data that is sent to and processed by the tools.

The benefits of the packet header or label stripping feature of Cisco Nexus Data Broker switch are as follows:

- Enable Multiprotocol Label Switching (MPLS) label stripping
- Native support for VXLAN header stripping from copy traffic
- Support for Generic Route Encapsulation (GRE) header stripping
- Q-in-Q VLAN header stripping at egress

NDB aligns the legacy VXLAN, IVXLAN, ERSPAN, GRE, and MPLS stripping functionality to the Overlay Forwarding Manager (OFM) based model. The OFM hosts the command line interface (CLI) for header stripping functionality.

This chapter contains the following sections:

• VXLAN and iVXLAN Header Stripping, on page 4
• ERSPAN Header Stripping
• GRE Header Stripping
• MPLS Header Stripping

Guidelines and Limitations

This topic lists the guidelines and limitations applicable to all header stripping features.

General Guidelines and Limitations

- A maximum of 500 flow terminate interfaces are supported across all tunnel-profiles with various encapsulation types such as VxLAN, iVxLAN, GRE, and MPLS. For ERSPAN, the maximum flow terminate interfaces supported is 31.
- The co-existence can be on the same interface or different interfaces.
- The legacy MPLS stripping feature and OFM stripping features are mutually exclusive.
- After performing non-disruptive ISSU from an earlier release to Cisco NX-OS Release 10.2(3)F and performing any header stripping functions, if dot1q tunnel VLAN_tag is missing or set to vlan_id=1, then remove and add the port ACL from L2 interfaces for that particular stripping-enabled interface.
- If no VLAN is configured on an interface, but the switchport mode dot1q-tunnel command is configured on that interface, then stripped packets will have VLAN=1 by default.
- In a scenario where incompatible OFM commands are present in the show running command output, and disruptive ISSU from Cisco NX-OS Release 10.2(3)F to an earlier release is done, wherein OFM commands were not supported in the earlier NX-OS version, then appropriate errors are displayed. However, the show incompatibility command does not flag such errors for OFM-related incompatibility commands.
- The OFM-based GRE, ERSPAN, and MPLS stripping features are supported only on TORs, not on line cards.
- As part of the encapsulation (iVXLAN, VXLAN, GRE, MPLS, ERSPAN), the following restrictions are common:
  - Two or more tunnel-profiles cannot have the same encapsulation-type.
  - OFM-based header stripping features are not supported when feature tunnel is enabled.

Release Specific Guidelines

Beginning with Cisco NX-OS Release 10.2(3)F

- Traffic with IPv6 inner packet is supported for all stripping functions.
- The MPLS stripping using the OFM model co-exists with the other stripping features. However, the existing MPLS stripping feature will continue to support MPLS stripping when co-existence is not needed with other types of stripping features.

- ERSPAN coexistence on the same interface is supported. However, this is supported on 9300-FX2 and later platforms only.

VXLAN and iVXLAN Header Stripping

Overview

Nexus Data Broker (NDB) VXLAN and iVXLAN header stripping is a feature that enables switches to remove VXLAN and iVXLAN headers from packets upon receipt.

- It allows switches to terminate VXLAN and iVXLAN packets.
- It strips VXLAN and iVXLAN headers from incoming packets.
- It facilitates packet processing in scenarios involving TAP and SPAN configurations.

Scenarios for Header Stripping

NDB switches receive packets in the following scenarios:

- Test Access Point (TAP) ports between spines and leaf are placed on the Fabric Links in the ACI fabric.
- Switched Port Analyzer (SPAN) sessions are configured, or TAPs placed in the VXLAN overlay network.

Supported PIDs

Guidelines and Limitations

Post VXLAN and iVXLAN Header Strip Guidelines

- The interface will allow slapping Q-in-Q VLAN on inside packet.
- Packet CRC will be properly performed.
- Inside packets will be allowed to filter using ingress port ACLs.

Not Supported Features

VXLAN and iVXLAN strip is not supported if the following features are enabled:

- NV overlay
- VN-segment-vlan
- Legacy MPLS strip and tap-aggregation

Release Specific Guidelines

Beginning with Cisco NX-OS Release 10.2(1)F

- The VXLAN and iVXLAN stripping features are supported on Cisco Nexus 9364C and 9300-EX, 9300-FX, 9300-FX2, 9500-EX, and 9500-FX line cards.

Beginning with Cisco NX-OS Release 10.2(2)F

- The VXLAN and iVXLAN stripping features are supported on Cisco Nexus 9300-GX and 9300-GX2 platform switches.

Beginning with Cisco NX-OS Release 10.2(3)F

- VXLAN strip is supported on Cisco N9K-C93180YC-FX3 and N9K-C93108TC-FX3P platform switches.
- A maximum of 5 tunnel-profiles are supported.

Beginning with Cisco NX-OS Release 10.2(4)M

- The iVXLAN stripping feature is supported on Cisco N9K-C93180YC-FX3 and N9K-C93108TC-FX3P platform switches.

Platform Specific Guidelines

- You must be able to strip VXLAN, and iVXLAN headers without being PTEP/VTEP.
- VXLAN header strip is enabled per port.
- Ports must be able to manage both tunneled and non-tunneled packets.
- Layer 2 switch port mode trunk or Layer 2 PO interfaces must be able to strip the VXLAN header.
- Ensure that the Tap-ACL contains proper ACE with redirect keyword, where the redirect interfaces are pointing toward the egress/analyzer ports, else the packet will be flooded back on the same ingress port.
- OFM enables VXLAN strip capability for standard ISSU and LXC-ISSU.
- A maximum of 4 tunnel-profiles can be created on a switch, one per type of encapsulation.
- A maximum of 12 redirect interfaces (prior to Release 10.2(1)) and 32 redirect interfaces (Release 10.2(1) and later) can only be configured in a single ACE of the TAP aggregation policy.
- For Cisco Nexus 9300-GX platform switches, post VXLAN strip, L2 header addresses are re-written as follows: Source MAC as VDC MAC address and Destination MAC as 000000abcdef.

Supported Features

- VXLAN header strip is supported when VXLAN underlay is V4.
- VXLAN stripping is supported when the default UDP value is used.
- The following switches support VXLAN and iVXLAN header stripping feature from the mentioned releases:
  - N9K-C9348GC-FX3 – 10.4(1)F
  - N9K-C9332D-H2R – 10.4(1)F
  - N9K-C93108TC-FX3 – 10.4(2)F
  - N9K-C93400LD-H1 – 10.4(2)F

- N9K-C92348GC-FX3 – 10.5(2)F
- N9K-X9736C-FX3 line card - 10.5(2)F

Configure Nexus Data Broker Termination

The following steps outline the termination of NDB for VXLAN. The same procedure is followed for iVXLAN header strip.

Note To change encapsulate tunnel type from VXLAN to iVXLAN or vice versa, the configured tunnel must be removed using no encapsulate CLI.

Note Ensure that the below CLIs are configured to enable stripping of VXLAN or iVXLAN on interfaces:

- destination any
- encapsulation vxlan
- flow terminate interface add Ethernet 1/1

If any of the above CLIs are missing, stripping of VXLAN or iVXLAN will not happen on the ports specified in flow term CLI.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal** and enable feature ofm using the command **feature ofm**

Example:

```
switch# configure terminal
```

Example:

```
switch (config)# feature ofm
```

Step 2 Enable static VXLAN tunnels using the command **tunnel-profile profile-name** and set appropriate encapsulation type for the tunnel profile using the command **encapsulation vxlan**

Example:

```
switch(config)# tunnel-profile vtep_vxlan_term
                        switch(config-tnl-profile)#
```

Example:

```
switch(config-tnl-profile)# encapsulation vxlan
                        switch(config-tnl-profile)#
```

Step 3 Set required destination for the tunnel profile using the command **destination any**

Example:

```
switch(config-tnl-profile)# destination any
```

Step 4 Add ethernet1/1 to the flow term list (if the **no flow terminate interface** command was configured) using the command **flow terminate interface ethernet 1/1** and remove Ethernet 1/1 port only using the command **flow terminate interface remove ethernet 1/1**

Example:

```
switch(config-tnl-profile)# flow terminate interface ethernet 1/1
```

Example:

```
switch(config-tnl-profile)# flow terminate interface remove ethernet 1/1
```

Step 5 Add e1/2, e1/3, e1/4, e1/5 to an existing list of flow terminate interfaces using the command **flow terminate interface add ethernet 1/2-5**

Example:

```
switch(config-tnl-profile)# flow terminate interface add ethernet 1/2-5
```

Note
While adding flow terminate interface, CLI doesn't check whether L2 port interface exists or enabled. For example, e1/10 is a non-breakout mode. CLI allows interface e1/10/1-4 to add for flow terminate list. When e1/10 is a breakout, VXLAN header strip feature functions.

Step 6 Add port channel 100-110 to old list uisng the command **flow terminate interface add port-channel 100-110**

Example:

```
switch(config-tnl-profile)# flow terminate interface add po100-110
```

New list will be e1/10-11 and po100-110.

Step 7 Remove all flow and terminate interfaces from profile uisng the command **no flow terminate interface** and enable feature tap-aggregation using the command **feature tap-aggregation**

Example:

```
switch(config-tnl-profile)# no flow terminate interface
```

Example:

```
switch(config)# feature tap-aggregation
```

Step 8 Create an IPACL and enters the IP access list configuration mode using the command **ip access-list** <access-list name> and create an IP ACL rule that permits traffic to be redirected per its conditions using the command **[no] permit** protocol source destination redirect interfaces

Example:

```
switch(config)# ip access-list test
                        switch(config-acl)#
```

Example:

```
permit ip any any redirect interface ethernet 1/1, ethernet 1/19
```

The no version of this command removes the permit rule form the policy.

Note
When you enter an interface for the TAP aggregation policy, do not abbreviate it. When you enter a list of interfaces, separate them with commas, but no spaces.

Step 9 Apply the port access list to the ERSPAN strip/terminating port using the command **ip port access-group** <access-group name> **in**

Example:

```
configure terminal
                        interface Ethernet 1/32
                        ip port access-group test in
```

The following example shows VXLAN and iVXLAN header stripping, the procedure is same for iVXLAN:

```
switch(config-tnl-profile)# show run ofm
show running-config ofm
feature ofm
tunnel-profile vxlan1
encapsulation vxlan
destination any
flow terminate interface add port-channel101
flow terminate interface add Ethernet1/1

tunnel-profile vxlan2
encapsulation ivxlan
destination any
flow terminate interface add port-channel101
flow terminate interface add Ethernet1/1
switch(config-tnl-profile)#
switch(config-tnl-profile)# show tunnel-profile
Profile : vxlan1
Encapsulation : Vxlan
State : UP
Destination : Any
Terminate Interfaces : 2
Terminate List : port-channel101 Ethernet1/1
Profile : vxlan2
Encapsulation : iVxlan
State : UP
Destination : Any
Terminate Interfaces : 2
Terminate List : port-channel101 Ethernet1/1
switch(config-tnl-profile)#
```

ERSPAN Header Stripping

Overview

ERSPAN header stripping is a feature that removes the ERSPAN header from incoming ERSPAN packets on NX-OS or NDB switches and forwards the packets inline based on ACL configuration.

- It strips ERSPAN headers from incoming packets.
- It operates inline on NX-OS and Nexus Data Broker switches.
- It redirects packets to outside servers based on ACL configuration.

Supported PIDs

Beginning with Cisco NX-OS Release 10.2(1)F, ERSPAN header stripping is supported on Cisco Nexus 9300-FX2, 9300-FX3, 9300-GX, and 9300-GX2 platform switches. However, this feature is only supported on TOR switches.

Guidelines and Limitations

General Limitations

- The ERSPAN header stripping feature is supported only on Cisco Nexus 9300-FX2, 9300-FX3, 9300-GX, and 9300-GX2 platform switches, and only on TOR switches.
- The incoming port must be a layer 2 port, with connectivity to layer 3 through SVI.
- Strips all the incoming ERSPAN headers on the terminating port.
- The incoming port must be a layer 2 port, but its connectivity to layer 3 must be through SVI.
- ERSPAN destination session and ERSPAN stripping cannot co-exist.
- The total number of terminating ports including port channel members cannot be more than 31.
- Mode tap-agg should not be configured for this feature.
- Tunnel profile for all ERSPAN ID is supported. Termination of specific ERSPAN session ID is not supported. Traffic with any ERSPAN session ID will be terminated at the termination node.

General Guidelines

- Only 1 tunnel profile per node is supported.
- A maximum of 31 flow terminate interfaces are supported on tunnel-profile with encap type: ERSPAN.
- The ERSPAN header stripping feature is supported on Cisco Nexus 9300-FX2, 9300-FX3, 9300-GX, and 9300-GX2 platform switches. Note that this feature is only supported on TOR switches.
- You need to enable ERSPAN stripping on the port so that ERSPAN strip/redirect works properly. Do not send ERSPAN traffic on ports where other strips are enabled.
- Strips all the incoming ERSPAN headers on the terminating port.
- This feature works only when OFM tunnel profiles and ACL redirect are configured and when port ACL is applied to the layer 2 terminating port.
- There can be only one tunnel profile for ERSPAN encapsulation on the switch.
- Appropriate tcam needs to be carved to use port acl, for example, **tcam region ing-ifacl** should be used for carving.

Configure ERSPAN Header Stripping

The following steps outline the configuration for ERSPAN header stripping.

Note Ensure that the below CLIs are configured to enable stripping of ERSPAN on interfaces:

- encapsulation erspan
- erspan session-id all
- flow terminate interface add e1/16

If any of the above CLIs are missing, stripping of ERSPAN does not happen on the ports specified in flow term CLI.

Procedure

Step 1 Enter global configuration mode uisng the command **configure terminal**

Example:

```
switch# configure terminal
```

Step 2 Enable feature ofm using the command **feature ofm**

Example:

```
switch (config)# feature ofm
```

Step 3 Enable static ERSPAN tunnels uisng the command **tunnel-profile** <profile-name>

Example:

```
switch(config)# tunnel-profile foo
                        switch(config-tnl-profile)#
```

Step 4 Set appropriate encapsulation type for the tunnel profile uisng the command **encapsulation erspan**

Example:

```
switch(config-tnl-profile)# encapsulation erspan
                        switch(config-tnl-profile)#
```

Step 5 Denote the monitored session that the related ERSPAN packet is associated with on the source switch using the command **erspan session-id all**

Example:

```
switch(config-tnl-profile)# erspan session-id all
```

Step 6 Add ethernet1/16 to the flow term list (if no flow CLI is configured) using the command **flow terminate interface add ethernet1/16**

Example:

```
switch(config-tnl-profile)# flow terminate interface add ethernet1/16
```

Step 7 Create an IPACL and enter the IP access list configuration mode using the command **ip access-list** <access-list-name>

Example:

```
switch(config)# ip access-list test
                        switch(config-acl)#
```

Step 8 Create an IP ACL rule that permits traffic to be redirected per its conditions using the command [**no**] **permit** protocol source destination **redirect** interfaces

Example:

```
permit ip any any redirect ethernet1/1,ethernet1/19
```

The no version of this command removes the permit rule from the policy.

Note
When you enter an interface for the TAP aggregation policy, do not abbreviate it. When you enter a list of interfaces, separate them with commas, but no spaces.

Step 9 Apply the port access list to the ERSPAN strip/terminating port using the command **ip port access-group** <access-group name>_ **redir in**

Example:

```
interface e1/16 (config-if)# ip port access-group test in
```

The following example shows ERSPAN header stripping:

```
switch(config)# feature ofm
switch(config)# tunnel-profile foo
switch(config-tnl-profile)# encapsulation erspan
switch(config-tnl-profile)# erspan session-id all
switch(config-tnl-profile)# flowterminate interface add ethernet1/16
switch(config)# ip access-list test
permit ip any any redirect ethernet1/1,ethernet1/19
interfacee1/16 (config-if)# ip port access-group test in
```

What to do next

To display the ERSPAN header stripping configuration, perform one of the following tasks:

Command	Purpose
show run ofm	Displays the tunnel profiles.
show run aclmgr	Displays all the ACLs and the application of those ACLs on the interfaces.
show ip access-list acl_nam	Displays ACL hit and redirected packets count.
show tunnel-profile	Displays the states of all tunnel profiles.

GRE Header Stripping

Overview

The GRE header stripping feature removes the GRE header from encapsulated packets and adds a custom Ethernet header to the inner packet.

- After GRE strip, an Ethernet header is added to the inner packet.
- The inner packet in a GRE encapsulated packet does not contain an Ethernet header.
- The custom Ethernet header includes specific fields such as VLAN, destination MAC, and source MAC addresses.

Custom Ethernet Header Fields

After the GRE header is stripped, the following custom fields are added to the Ethernet header:

- 802.1q header with VLAN configured on the incoming port.
- Destination MAC address will be set to 00:00:00:ab:cd:ef or 000.000.abc.def.
- Source MAC address will be set to VDC MAC address of the switch.

The GRE header is removed for the following reasons :

- Some of the monitoring tools do not have the ability to understand an encapsulated packet.
- Addition of another header adds to the packet size which hampers the optimization of the amount of data that is sent to the tools.
- Presence of another header skews the analytics data.

Guidelines and Limitations

Platform Specifications

- A maximum of 500 flow terminate interfaces are supported on tunnel-profile with encap type: iVXLAN/VXLAN/GRE.
- A maximum of 31 flow terminate interfaces are supported on tunnel-profile with encap type: ERSPAN.

Not Supported Features

1. Flow terminate interfaces cannot share ESPRAN and GRE/VXLAN/IVXLAN profiles.
2. Feature OFM and feature tunnel cannot co-exist on the same switch.
3. The configuration of **mode tap-aggregation** should not be present on interfaces where GRE header stripping functionality is enabled.
4. Tunnel-encapsulation type modification is not allowed.

Configuration Guidelines

1. To remove a flow interface from a tunnel-profile, use **remove** instead of **no** . Using **no** in the flow terminate command deletes all interfaces from the flow terminate list.
2. If a GRE strip-enabled interface receives ERSPAN traffic, stripping succeeds, but traffic will not be forwarded to the redirect port.
3.

4. When flow terminate interface CLI is configured without **add** keyword, it acts as **replace** , which means previously added flow terminate interfaces are deleted and only new ones will act as flow terminate interfaces.
5. After non-disruptive upgrade from previous NX-OS version to 10.2(3)F, port ACL must be removed from all interfaces and added before enabling GRE header strip feature for a particular interface.
6. The **hardware acl tap-agg redirect disable-dot1q-sharing** command is required on 9300-GX to allow dot1q tunnel propagation. The switch needs reload after enabling this command.

Configuration

- Configuration for ingress ports includes setting the interface to access VLAN, enabling dot1q-tunnel mode, applying an IP access group, and enabling the interface.
- Configuration for egress ports includes setting the interface to trunk mode, enabling the interface, and applying an IP access list with specific permit and redirect rules.

Configuration Details for Egress and Ingress Ports

The following is the configuration for ingress ports:

- `interface eth1/1`

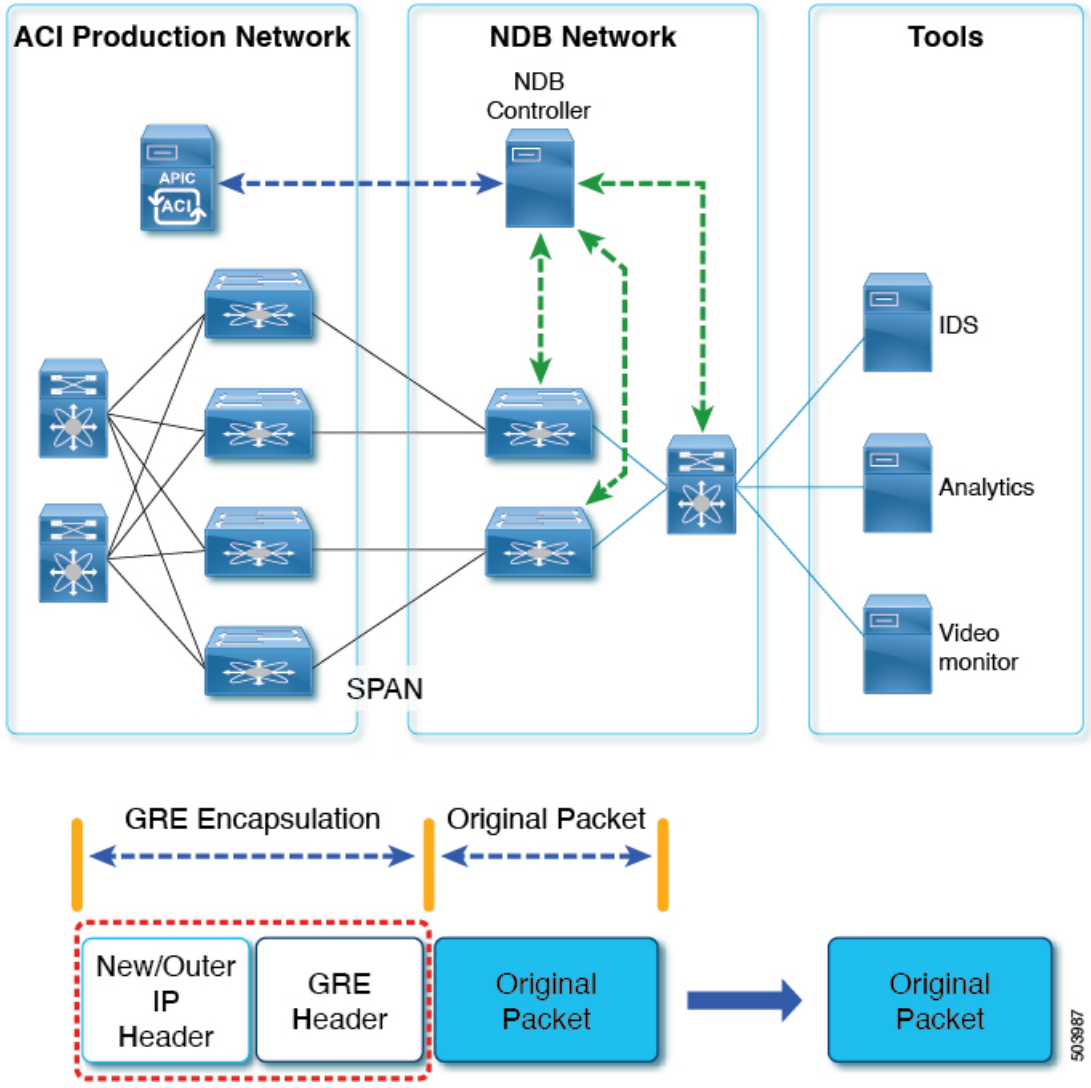
```
switchport access vlan 101
switchport mode dot1q-tunnel
ip port access-group ndb_acl in <<<
no shutdown
```

The following is the configuration for egress ports:

- `interface Ethernet1/7`

```
switchport mode trunk
no shutdown
IP access list ndb_acl
statistics per-entry
10 permit udp any any eq 4789 redirect Ethernet1/7
15 permit ip any any redirect Ethernet1/7
```

Figure 2: NDB GRE Header Strip Solution



CLIs for GRE Header Strip

The following are the CLIs to be configured for enabling GRE header on an interface:

```
feature ofm
tunnel-profile gre_strip
    encapsulation gre
    destination any
    flow terminate interface add Ethernet1/1-10
```

The following is the show command for tunnel-profile:

```
switch# show tunnel-profile gre_strip
  Profile               : gre_strip
  Encapsulation         : GRE
  State                 : UP
  Destination           : Any
  Terminate Interfaces  : 10
```

```
  Terminate List        : Ethernet1/1 Ethernet1/2 Ethernet1/3 Ethernet1/4 Ethernet1/5
Ethernet1/6 Ethernet1/7 Ethernet1/8 Ethernet1/9 Ethernet1/10
```

MPLS Header Stripping

Overview

MPLS label stripping is a process that removes the MPLS header from incoming packets and adds an Ethernet header with specific custom fields.

- It is supported for both IPoMPLS and EoMPLS packet formats.
- After label strip, an Ethernet header is added to the inner packet.
- The Ethernet header includes a VLAN tag, destination MAC, and source MAC address.

Custom Fields

After MPLS label stripping, the following custom fields are added to the Ethernet header:

- 802.1q header with VLAN configured on the incoming port.
- Destination MAC address is set to 00:00:00:ab:cd:ef or 000.000.abc.def.
- Source MAC address is set to VDC MAC address of the switch.

Guidelines and Limitations

Guidelines for Migration from legacy MPLS header stripping to OFM-based configuration

The following guidelines and limitations apply when migrating from legacy MPLS header stripping to OFM-based configuration:

Not Supported Features :

- Legacy MPLS stripping implementation cannot co-exist with any OFM-based stripping.
- Feature OFM and feature tunnel cannot co-exist on the same switch.
- Tunnel-encapsulation type modification is not allowed.

```
QP-CF-1(config-tnl-profile)# encapsulation mpls
Error: encap-type modify not allowed, delete and add again
```

- On an interface where MPLS head strip is enabled, mode tap-aggregation should not be present.
- MPLS Stripping is based on IP PACL, so do not use MAC-ACL for stripping.
- During MPLS stripping, incoming VLAN in the original packet is not preserved.
- After non-disruptive upgrade from previous NX-OS version to 10.2(3)F, port ACL must be removed from all interfaces and added before enabling MPLS header stripping feature for a particular interface.

- The **hardware acl tap-agg redirect disable-dot1q-sharing** command is required on Cisco Nexus 9300-GX platform switches to allow dot1q tunnel propagation. The switch needs reload after enabling this command.
- If ERSPAN ACL redirect tunnel-profile is not configured and the interface is receiving ERSPAN packets, then the ERSPAN packets will hit ERSPAN ACL redirect entries in TapAgg policy and will not be stripped.
- With ERSPAN tunnel-profile, when ingress interface is converted from dot1q-tunnel to trunk mode, egress packets will have dot1q tag with VLAN=1. This tagging takes place for both stripped packets and regular IP packets that are redirected.
- When an MPLS strip-enabled interface receives ERSPAN traffic, stripping succeeds, but traffic is not forwarded to the redirect port.
- To remove flow interface from a tunnel-profile, use **remove** instead of **no** . The use of **no** in flow terminate command will delete all interfaces from flow terminate list.

  For example:

```
switch(config)# tunnel-profile mpls_strip
switch(config-tnl-profile)# flow terminate interface remove Ethernet 1/48
```

- When flow terminate interface command is configured without the **add** keyword, it acts as **replace** , which means previously added flow terminate interfaces are deleted and only the new ones will act as flow terminate interfaces.
- MPLS packets with multicast bit set cannot be terminated, but MPLS packets with unicast bit set can be terminated.
- Ingress interface can be either in trunk mode or access mode. Both modes allow redirection of tagged and untagged packets. When access-mode is used along with dot1q-tunnel mode, after header stripping VLAN_tag is added as specified by the access-mode.

NDB MPLS Header Stripping Feature Guidelines

Beginning with Cisco NX-OS Release 10.2(3)F, the NDB MPLS Header Stripping feature is supported.

-

Note The OFM MPLS stripping feature is supported only on TORs; it is not supported on line cards.

Cleanup for Migrating from legacy MPLS stripping functionality

Migrating from legacy MPLS stripping functionality requires the following cleanup before enabling OFM-based MPLS stripping:

- Removal of **mode tap-aggregation** at interface(s) level
- Removal of **mpls strip; mpls strip dot1q** at the global level
- Save the configuration and reload the switch with the above configuration

EoMPLS Guidelines

- EoMPLS stripping can co-exist with all other header stripping features on same or different interfaces.
-
- Pseudo Wire Control Word is not supported.
- On Cisco Nexus 9300-GX platform switches, two ingress ports cannot share acl unless the dot1q vlan config is the same on them, else tagging does not work.

Configuration

- This topic provides configuration steps for ingress ports, including access VLAN assignment, dot1q-tunnel mode, and applying an IP access group.
- It also covers configuration for egress ports, such as trunk mode setup and applying IP access lists with redirect actions.
- These configurations are essential for managing traffic flow and security on network switches.

Configuration Details for Egress and Ingress Ports

The following sections provide the configuration commands for both ingress and egress ports on the switch.

- Ingress port configuration
- Egress port configuration

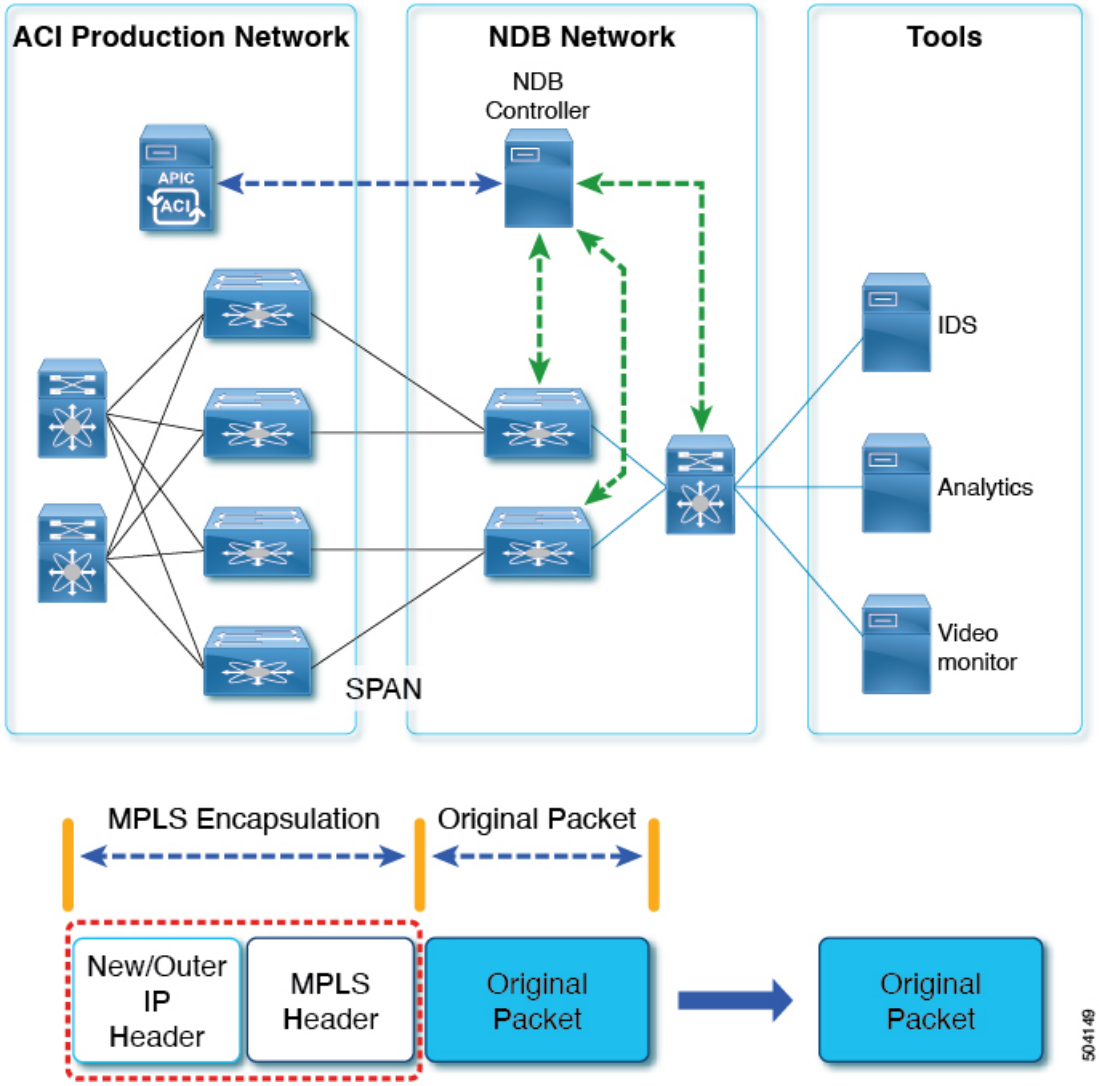
1. Configure the ingress port with access VLAN, dot1q-tunnel mode, and apply the IP access group.
2. Configure the egress port as a trunk and apply the IP access list with redirect actions.

- Ingress port: Assign VLAN, set mode, apply ACL
- Egress port: Set trunk mode, apply ACL with redirect



Note In case of decapsulated packet such as MPLS, the NDB-switch adds an Ethernet/VLAN header to the **original packet** , so egressing packet will have Ethernet/VLAN - original packet.

Figure 3: NDB MPLS Header Strip Solution



Example configuration for ingress port:

```
interface eth1/1
                    switchport access vlan 101
                    switchport mode dot1q-tunnel
                    ip port access-group ndb_acl in
                    no shutdown
```

Example configuration for egress port:

```
interface Ethernet1/7
                    switchport mode trunk
                    no shutdown
                    IP access list ndb_acl
                    statistics per-entry
                    10 permit udp any any eq 4789 redirect Ethernet1/7
                    15 permit ip any any redirect Ethernet1/7
```

Commands for MPLS Header Strip Feature

The following commands should be configured for enabling MPLS header on an interface:

```
feature ofm
tunnel-profile
mpls_strip encapsulation mpls destination any
flow terminate interface add Ethernet1/1-10
```

The show command for tunnel-profile is as follows:

```
switch# show tunnel-profile mpls_strip
  Profile               : mpls_strip
  Encapsulation         : MPLS
  State                 : UP
  Destination           : Any
  Terminate Interfaces  : 10
  Terminate List        : Ethernet1/1 Ethernet1/2 Ethernet1/3 Ethernet1/4 Ethernet1/5
Ethernet1/6 Ethernet1/7 Ethernet1/8 Ethernet1/9 Ethernet1/10
```