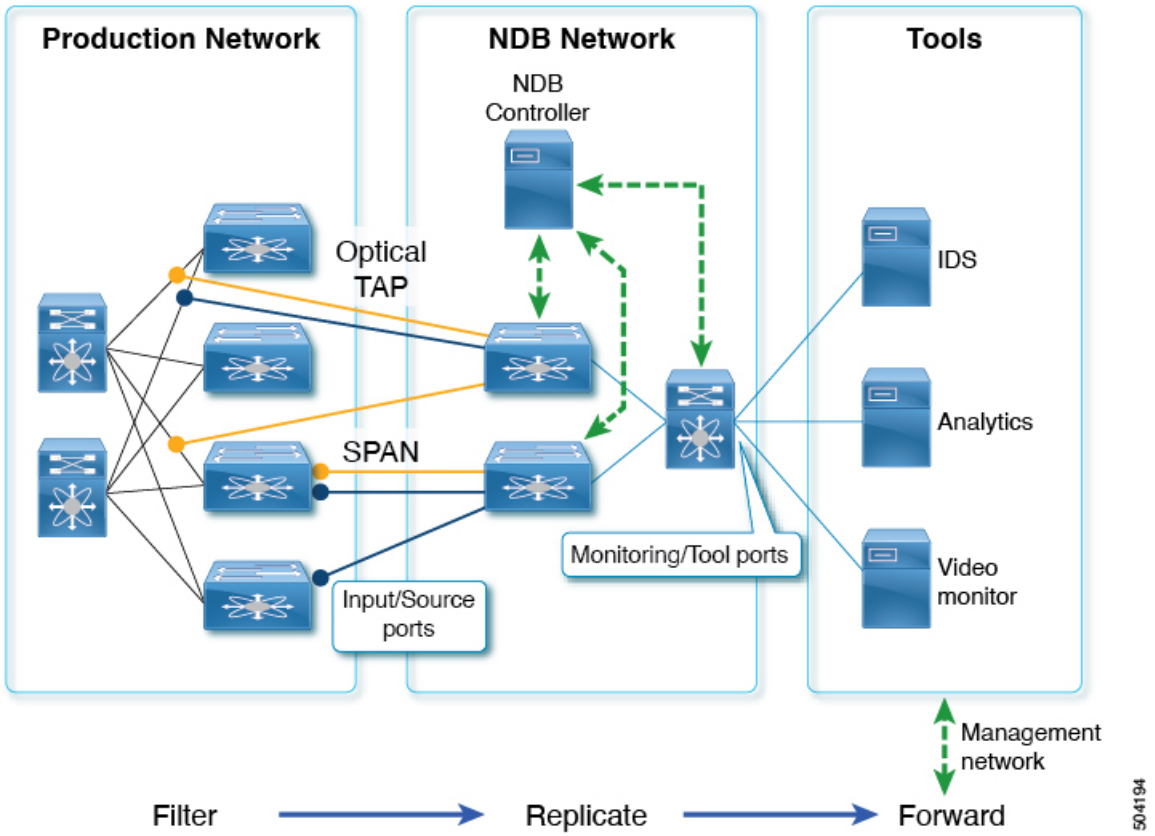
Configuring Header Stripping Features for Nexus Data Broker

Introduction to Header Stripping Features for Nexus Data Broker

Cisco Nexus Data Broker (NDB) builds scalable packet broker network solutions that are easy to operate. The Cisco Nexus Dashboard Data Broker controller software and Cisco Nexus switches provide a new software-defined approach for monitoring both out-of-band and inline network traffic.

Figure 1: NDB Centralized Deployment Model



NDB switches are used for packet monitoring. Packet monitoring is needed for performance monitoring, intrusion detection, check compliance, and so on.

For header strip, Out-of-Band monitoring is done, which means it is non-intrusive, and the copy of the packet is monitored using TAP or SPAN. So, the traffic is filtered and replicated from production network, stripped off any headers on NDB switches, and forwarded to Tools for monitoring. Input/source ports mentioned here are the ports on which the header stripping takes place. Monitoring/Tool ports are the ports which are connected directly to Tools.

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

Thus, 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:

Guidelines and Limitations for Header Stripping

The guidelines and limitations applicable to all the header stripping features are as follows:

- A maximum of 500 flow terminate interfaces are supported across all tunnel-profiles with various encap types such as VxLAN, iVxLAN, GRE, and MPLS. For ERSPAN, the maximum flow terminate interfaces supported is 31.
- Beginning with Cisco NX-OS Release 10.2(3)F, 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 type of stripping features.
- The co-existence can be on the same interface or different interfaces.

Note Beginning with Cisco NX-OS Release 10.2(3)F, ERSPAN coexistence on the same interface is supported. However, this is supported on 9300-FX2 and later platforms only.

- The legacy MPLS stripping feature and OFM stripping features are mutually exclusive.
- Beginning with Cisco NX-OS Release 10.2(3)F, traffic with IPv6 inner packet is supported for all stripping functions.
- 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.

VXLAN and iVXLAN Header Stripping for Nexus Data Broker

This subchapter describes VXLAN and iVXLAN header stripping procedure for Nexus Data Broker (NDB).

This chapter contains the following sections:

About Nexus Data Broker – VXLAN and iVXLAN Header Stripping

Nexus Data Broker (NDB) VXLAN, and iVXLAN termination allow switches the ability to strip headers when VXLAN, and iVXLAN packets are received.

NDB switch receives packets in the below mentioned 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 to Strip VXLAN and iVXLAN

Beginning with Cisco NX-OS Release 10.2(2)F, the VXLAN stripping feature is supported on Cisco Nexus 9364C and 9300-EX, 9300-FX, 9300-FX2, 9300-FX3, 9300-GX, 9300-GX2, 9500-EX, and 9500-FX line cards.

Beginning with Cisco NX-OS Release 10.2(2)F, the iVXLAN stripping feature is supported on Cisco Nexus 9364C and 9300-EX, 9300-FX, 9300-FX2, 9300-GX, 9300-GX2, 9500-EX, and 9500-FX line cards.

Guidelines and Limitations for VXLAN and iVXLAN Header Strip

- VXLAN header strip is supported when VXLAN underlay is V4.
- You must be able to strip VXLAN, and iVXLAN headers without being PTEP/VTEP.
- VXLAN header strip is enabled per port.
- VXLAN and iVXLAN strip is not supported if the following features are enabled:
  - NV overlay
  - VN-segment-vlan
  - Legacy MPLS strip and tap-aggregation

- VXLAN stripping is supported when the default UDP value is used.
- 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.
- 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.
- A maximum of 4 tunnel-profiles can be created on a switch, one per type of encapsulation. However, beginning with Cisco NX-OS Release 10.2(3)F, a maximum of 5 tunnel-profiles are supported.
- 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.
- 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.
- 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.
- 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-C9364C-H1 – 10.4(3)F

The below statements are true for post VXLAN, and iVXLAN header strip:

- 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.

Configuring 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.

SUMMARY STEPS

1. **configure terminal**
2. **feature ofm**
3. **tunnel-profile profile-name**
4. **encapsulation vxlan**
5. **destination any**
6. **flow terminate interface ethernet 1/1**
7. **flow terminate interface remove ethernet 1/1**
8. **flow terminate interface add ethernet 1/2-5**
9. **flow terminate interface add port-channel 100-110**
10. **no flow terminate interface**
11. **feature tap-aggregation**
12. **ip access-list** <access-list name>
13. **[no] permit** protocol source destination redirect interfaces
14. **ip port access-group** <access-group name> **in**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**feature ofm** **Example:** `switch (config)# feature ofm`	Enables feature ofm.

	Command or Action	Purpose

Step 3	**tunnel-profile profile-name**
	Example:
	`switch(config)# tunnel-profile vtep_vxlan_term`
	`switch(config-tnl-profile)#`
		Enables static VXLAN tunnels.

Step 4	**encapsulation vxlan**
	Example:
	`switch(config-tnl-profile)# encapsulation vxlan`
	`switch(config-tnl-profile)#`
		To set appropriate encapsulation type for the tunnel profile.

Step 5	**destination any** **Example:** `switch(config-tnl-profile)# destination any`	To set required destination for the tunnel profile.
Step 6	**flow terminate interface ethernet 1/1** **Example:** `switch(config-tnl-profile)# flow terminate interface ethernet 1/1`	To add ethernet1/1 to the flow term list (if the **no flow terminate interface** command was configured).
Step 7	**flow terminate interface remove ethernet 1/1** **Example:** `switch(config-tnl-profile)# flow terminate interface remove ethernet 1/1`	To remove Ethernet 1/1 port only.
Step 8	**flow terminate interface add ethernet 1/2-5** **Example:** `switch(config-tnl-profile)# flow terminate interface add ethernet 1/2-5`	To add e1/2, e1/3, e1/4, e1/5 to an existing list of flow terminate interfaces. **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 9	**flow terminate interface add port-channel 100-110** **Example:** `switch(config-tnl-profile)# flow terminate interface add po100-110`	To add port channel 100-110 to old list. New list will be e1/10-11 and po100-110.
Step 10	**no flow terminate interface** **Example:** `switch(config-tnl-profile)# no flow terminate interface`	To remove all flow and terminate interfaces from profile.
Step 11	**feature tap-aggregation** **Example:** `switch(config)# feature tap-aggregation`	Enables feature tap-aggregation.

	Command or Action	Purpose

Step 12	**ip access-list** <access-list name>
	Example:
	`switch(config)# ip access-list test`
	`switch(config-acl)#`
		Creates an IPACL and enters the IP access list configuration mode.

Step 13	**[no] permit** protocol source destination redirect interfaces **Example:** permit ip any any redirect interface ethernet 1/1, ethernet 1/19	Creates an IP ACL rule that permits traffic to be redirected per its conditions. 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 14	**ip port access-group** <access-group name> **in** **Example:** configure terminal interface Ethernet 1/32 ip port access-group test in	Applies the port access list to the ERSPAN strip/terminating port.

Configuration Example for VXLAN and iVXLAN Header Strip

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 for Nexus Data Broker

This subchapter describes ERSPAN header stripping procedure for Cisco Nexus platform switch. The primary use case for this is on Nexus Data Broker (NDB) switch.

This chapter contains the following sections:

About ERSPAN Header Stripping

This feature implements inline ERSPAN header stripping from the incoming ERSPAN packets on NX-OS switch or Nexus Data Broker (NDB) switch.

When the ERSPAN packets come in, this feature strips the ERSPAN header and forwards it to the outside box inline, that is, a packet comes on to a terminating port, and then, based on the ACL configuration, it is redirected to the ports that are connected to the outside server.

This feature does a single pass ERSPAN header stripping and PACL redirect.

Supported PIDs to Strip the ERSPAN Header

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 for ERSPAN Header Stripping

- 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.
- 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.

- This feature will work only 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.

Configuring 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.

SUMMARY STEPS

1. **configure terminal**
2. **feature ofm**
3. **tunnel-profile** <profile-name>
4. **encapsulation erspan**
5. **erspan session-id all**
6. **flow terminate interface add ethernet1/16**
7. **ip access-list** <access-list-name>
8. [**no**] **permit** protocol source destination **redirect** interfaces
9. **ip port access-group** <access-group name>**_redir in**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**feature ofm** **Example:** `switch (config)# feature ofm`	Enables feature ofm.

	Command or Action	Purpose

Step 3	**tunnel-profile** <profile-name>
	Example:
	`switch(config)# tunnel-profile foo`
	`switch(config-tnl-profile)#`
		Enables static ERSPAN tunnels.

Step 4	**encapsulation erspan**
	Example:
	`switch(config-tnl-profile)# encapsulation erspan`
	`switch(config-tnl-profile)#`
		To set appropriate encapsulation type for the tunnel profile.

Step 5	**erspan session-id all** **Example:** `switch(config-tnl-profile)# erspan session-id all`	The ERSPAN session ID denotes the monitored session that the related ERSPAN packet is associated with on the source switch.
Step 6	**flow terminate interface add ethernet1/16** **Example:** `switch(config-tnl-profile)# flow terminate interface add ethernet1/16`	To add ethernet1/16 to the flow term list (if no flow CLI is configured).

Step 7	**ip access-list** <access-list-name>
	Example:
	`switch(config)# ip access-list test`
	`switch(config-acl)#`
		Creates an IPACL and enters the IP access list configuration mode.

Step 8	[**no**] **permit** protocol source destination **redirect** interfaces **Example:** `permit ip any any redirect ethernet1/1,ethernet1/19`	Creates an IP ACL rule that permits traffic to be redirected per its conditions. 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	**ip port access-group** <access-group name>**_redir in** **Example:** `interface e1/16 (config-if)# ip port access-group test in`	Applies the port access list to the ERSPAN strip/terminating port.

Configuration Example for ERSPAN Header Stripping

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
```

Verifying the Configuration for ERSPAN Header Stripping

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

Command	Purpose
show run ofm	Displays the tunnel profiles.
show run acl mgr	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 for Nexus Data Broker

This subchapter describes GRE header stripping procedure for Cisco Nexus platform switch. The primary use case for this is on Nexus Data Broker (NDB) switch.

This chapter contains the following sections:

About NDB GRE Header Stripping

This feature allows you to strip the GRE header from packets that come in with a GRE encapsulation. The inner packet in a GRE encapsulated packet does not contain an ethernet header. So, after a GRE strip, an ethernet header is added to the inner packet with the following custom fields:

1. 802.1q header with vlan configured on the incoming port.
2. Destination MAC address will be set to 00:00:00:ab:cd:ef or 000.000.abc.def.
3. Source MAC address will be set to VDC MAC address of the switch.

NDB GRE Header Stripping Guidelines and Limitations

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

- Flow terminate interfaces cannot share ESPRAN and GRE/VXLAN/IVXLAN profiles.
- If GRE strip-enabled interface receives ERSPAN traffic, stripping succeeds, but traffic will not be forwarded to the redirect port.

- Feature OFM and feature tunnel cannot co-exist on the same switch.
- The NDB GRE Header Stripping feature is supported on Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, 9300-FX3, 9300-GX, and N9K-C9332D-GX2B platformsTORs. However, this feature is not supported on line cards.
- The configuration of **mode tap-aggregation** should not be present on interface where GRE header stripping functionality is enabled.
- Tunnel-encapsulation type modification is not allowed.

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

- 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.
- 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.
- Traffic with IPv6 inner packet is not supported for release 10.2(2)F.
- 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 particular interface.
- 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.

CLIs for GRE Header Strip Feature

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
```

Configuration for Egress and Ingress Ports

The following is the configuration for ingress ports:

```
interface eth1/1
    switchport access vlan 101
    switchport mode dot1q-tunnel
```

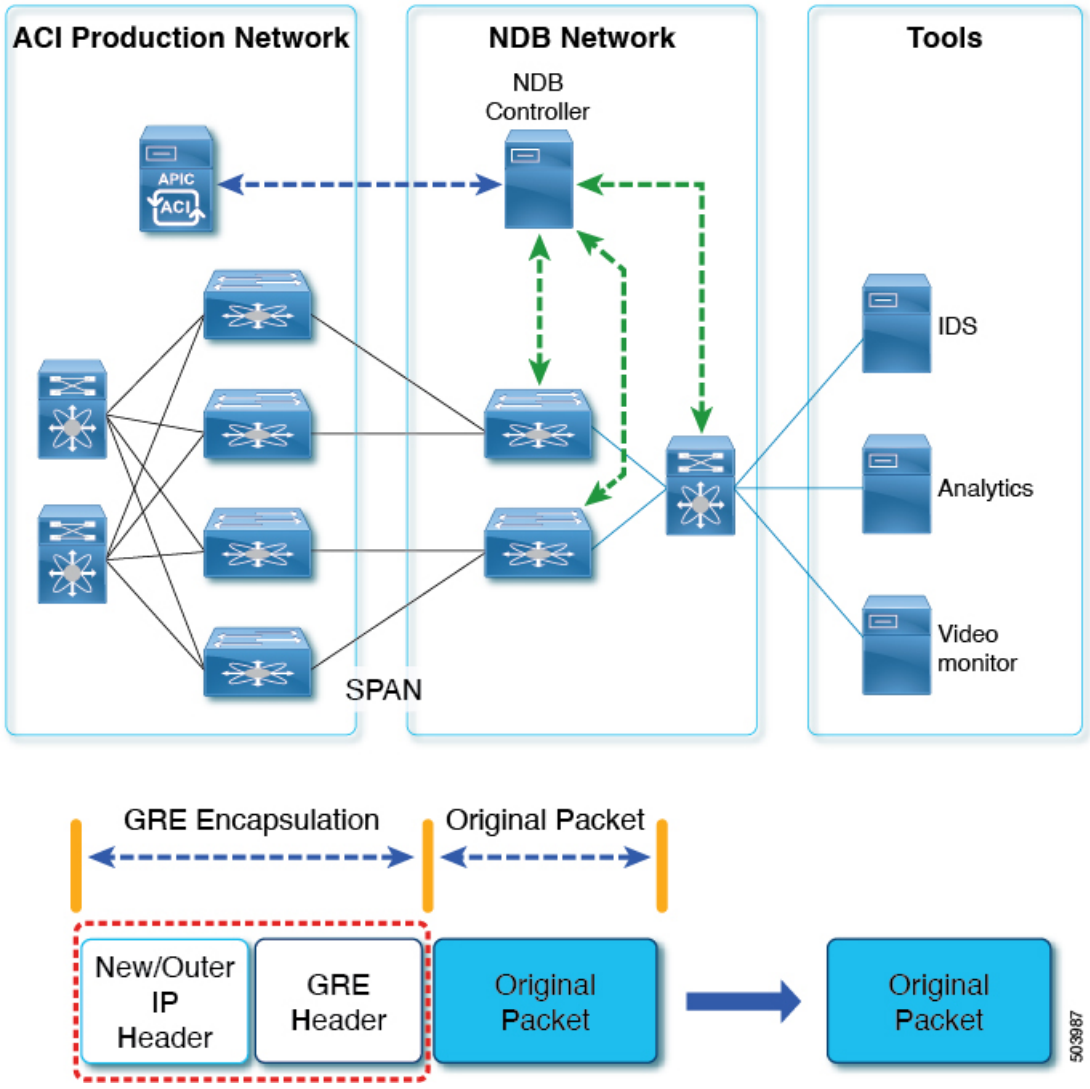
```
    ip port access-group ndb_acl in <<<
    no shutdown
```

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



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

MPLS Header Stripping for Nexus Data Broker

This subchapter describes MPLS header stripping procedure for Cisco Nexus platform switch. The primary use case for this is on Nexus Data Broker (NDB) switch.

This chapter contains the following sections:

About NDB MPLS Header Stripping

This feature allows you to strip the MPLS header from packets that come in with a MPLS encapsulation. MPLS label stripping is supported for both IPoMPLS and EoMPLS packet formats. After MPLS label strip, an ethernet header is added to the inner packet with the following custom fields:

1. 802.1q header with vlan configured on the incoming port.
2. Destination MAC address will be set to 00:00:00:ab:cd:ef or 000.000.abc.def.

Note For EoMPLS header stripping, this is applicable only on Cisco Nexus 9300-EX, 9300-FX, and 9300-GX platforms.

3. Source MAC address will be set to VDC MAC address of the switch.

Note For EoMPLS header stripping, this is applicable only on Cisco Nexus 9300-EX, 9300-FX, and 9300-GX platforms.

NDB MPLS Header Stripping Guidelines and Limitations

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

- Legacy MPLS stripping implementation cannot co-exist with any OFM-based stripping.
- Feature OFM and feature tunnel cannot co-exist on the same switch.
- 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
- Beginning with Cisco NX-OS Release 10.2(3)F, the NDB MPLS Header Stripping feature is supported.
  - IPoMPLS (packet format) header stripping is supported on Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, 9300-FX3, 9300-GX, and C9332D-GX2B platforms.

- EoMPLS (packet format) header stripping is supported only on Cisco Nexus 9300-EX platform switches. However, VPLS strip and control-word packet strip is not supported.

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

- 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.
- 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
```

- 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.
- 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.
- 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.
- 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.
- Until Cisco NX-OS Release 10.3(1)F, EoMPLS header stripping was supported only on Cisco Nexus 9300-EX platform switches (VPLS strip and control-word packet strip were not supported). Beginning with Cisco NX-OS Release 10.3(2)F, EoMPLS header stripping feature is also supported on Cisco Nexus 9300-FX, 9300-FX2, 9300-FX3, 9300-GX, and 9300-GX2 ToRs; it is not supported on line cards. The following guidelines and limitations are applicable:

- EoMPLS stripping can co-exist with all other header stripping features on same or different interfaces.
- For Cisco Nexus 9300-EX, 9300-FX, and 9300-GX platform switches, post EoMPLS header strip, L2 header addresses are re-written as follows: Source MAC as VDC MAC address and Destination MAC as 000000abcdef.
- 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.

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
```

Configuration for Egress and Ingress Ports

The following is the configuration for ingress ports:

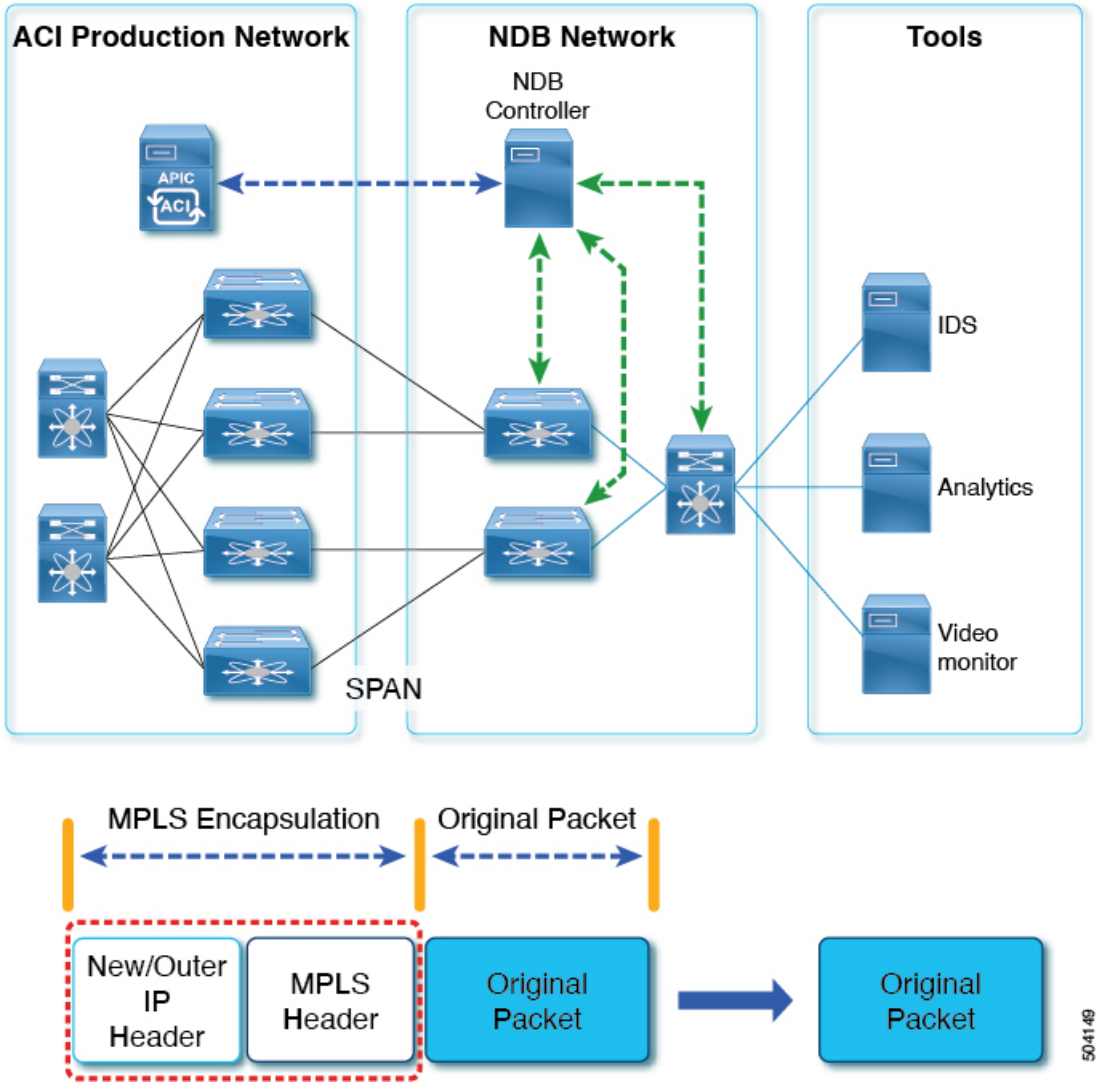
```
interface eth1/1
    switchport access vlan 101
    switchport mode dot1q-tunnel
    ip port access-group ndb_acl in
    no shutdown
```

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 3: NDB MPLS Header Strip Solution



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.