Network Management Configuration Guide, Cisco Catalyst IE9300 Rugged Series Switches

Cisco Catalyst IE9300 Rugged Series
Updated November 20, 2023

Full Cisco Trademarks with Software License

Full Cisco Trademarks with Software License

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

Communications, services, and additional information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions, and services, visit Cisco DevNet.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a gateway to the Cisco bug-tracking system, which maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. The BST provides you with detailed defect information about your products and software.

Documentation feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.

Bias Free Language

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

Topics included

1 ERSPAN

Topics:

ERSPAN

Explains the Cisco Encapsulated Remote Switched Port Analyzer (ERSPAN) feature that allows monitoring traffic on ports or VLANs across a Layer 3 network using GRE encapsulation.

The Cisco Encapsulated Remote Switched Port Analyzer (ERSPAN) is a network monitoring feature that

- allows you to monitor traffic on ports or VLANs and send the monitored traffic to destination ports over a Layer 3 (IP) network using Generic Routing Encapsulation (GRE) encapsulation,
- sends traffic to a network analyzer, such as a Switch Probe device or a Remote Monitoring (RMON) probe,
- supports source ports, source VLANs, and destination ports on different devices, which help remote monitoring of multiple devices across a network, and
- supports encapsulated packets of up to 9180 bytes.

ERSPAN components and session limits

ERSPAN consists of an ERSPAN source session, routable ERSPAN GRE-encapsulated traffic, and an ERSPAN destination session.

You can configure an ERSPAN source session, an ERSPAN destination session, or both on a device. A device on which only an ERSPAN source session is configured is called an ERSPAN source device. A device on which only an ERSPAN destination session is configured is called an ERSPAN termination device. A device can act as both an ERSPAN source device and an ERSPAN termination device.

Oversubscription of traffic can lead to a drop in management traffic on the destination device. To avoid oversubscription, ensure that the destination session is configured and is working on the destination device, before configuring a source session on the source device.

For a source port or a source VLAN, the ERSPAN can monitor the ingress, egress, or both ingress and egress traffic. By default, ERSPAN monitors all traffic, including multicast and Bridge Protocol Data Unit (BPDU) frames.

Session limits:

- A device supports up to 66 sessions.
- You can configure a maximum of eight source sessions. The remaining sessions can be configured as RSPAN destination sessions.
- A source session can be a local SPAN, RSPAN, or ERSPAN source session.
-

An ERSPAN source session is defined by these parameters:

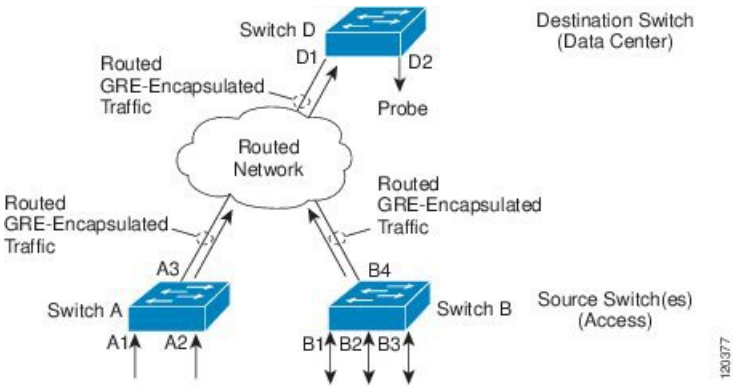
- A session ID
- ERSPAN flow ID
- list of source ports or source VLANs that are monitored by the session
- optional attributes such as IP type of service (ToS) and IP Time to Live (TTL) related to the Generic Routing Encapsulation (GRE) envelope
- destination and origin IP addresses are used as the destination and source IP addresses of the GRE envelope for the captured traffic.

Note

- ERSPAN source sessions do not copy ERSPAN GRE-encapsulated traffic from source ports. Each ERSPAN source session can have either ports or VLANs as sources, but not both.
- IPv4 delivery and transport headers are supported, including Type-II and Type-III headers.

  Port channels and switch virtual interfaces (SVIs) are supported.

Figure 1: ERSPAN configuration



Information about configuring ERSPAN

Describes the information sections that provide details about configuring ERSPAN functionality on network devices. This reference assists administrators in locating specific configuration parameters and operational requirements.

This topic provides information about configuring ERSPAN functionality on network devices.

Restrictions for configuring ERSPAN

Outlines the restrictions that apply when configuring ERSPAN functionality on network devices.

The restrictions for this feature are:

- Truncation is supported only on IPv4 spanned packets and not on Layer 2 packets without an IP header.
- Configure the ERSPAN destination interface for only one session. Do not configure the same destination interface for multiple ERSPANs or SPANs.
- You can configure either a list of ports or a list of VLANs as a source, but cannot configure both for a given session.
- Do not configure filter IP, MAC, or VLAN access-group and filter SGT together.
- When a session is configured through the ERSPAN CLI, the session ID and the session type cannot be changed. To change them, you must use the **no** form of the commands to remove the session and then reconfigure it.
- If you configure ERSPAN source sessions, locally-sourced RSPAN VLAN traffic from source trunk ports carrying RSPAN VLANs is not copied.
- If you configure ERSPAN source sessions, locally-sourced ERSPAN GRE-encapsulated traffic from source ports is not copied.

- If you disable the **ip routing** command for IPv4 connections, ERSPAN traffic will not flow to the destination port.

ERSPAN sources

Lists the sources that the Cisco ERSPAN feature supports for traffic monitoring and analysis.

The Cisco ERSPAN feature supports the following sources:

- **Source ports**: A source port that is monitored for traffic analysis. Source ports in any VLAN can be configured and trunk ports can be configured as source ports along with nontrunk source ports.
- **Source VLANs**: A VLAN that is monitored for traffic analysis.

ERSPAN destination ports

Explains the Layer 2 or Layer 3 ports to which ERSPAN source sends traffic for analysis.

An ERSPAN destination port is a Layer 2 or Layer 3 port that

- receives traffic from ERSPAN source for analysis
- becomes dedicated exclusively for ERSPAN feature use when configured, and
- can be configured as a trunk port to transmit encapsulated traffic.

ERSPAN destination port behavior

When you configure a port as a destination port, it can no longer receive any traffic. The port is dedicated for use only by the ERSPAN feature. An ERSPAN destination port does not forward any traffic except that required for the ERSPAN session. You can configure trunk ports as destination ports, which allows destination trunk ports to transmit encapsulated traffic.

SGT-based ERSPAN

Describes SGT-based ERSPAN filtering functionality for network traffic monitoring.

A Security Group Tag (SGT) is a 16-bit value that

- the Cisco Identity Services Engine (ISE) assigns to the user or endpoint session upon login
- the network infrastructure treats SGT as another attribute assigned to the session and inserts the Layer 2 tag into all traffic from that session
- supports a maximum of 50 SGT policies per session on a platform.

SGT filtering configuration restrictions

You cannot configure SGT filtering on an existing flow-based SPAN (FSPAN) or VLAN filter session

Prerequisites for configuring ERSPAN

Provides requirements for configuring ERSPAN functionality to ensure traffic is properly filtered. This process prevents unauthorized data transmission by applying specific traffic controls before tunnel transport.

Apply the access control list (ACL) filter before forwarding the monitored traffic to the tunnel.

Troubleshoot ERSPAN

Explains how to configure Encapsulated Remote Switched Port Analyzer (ERSPAN) using debug commands to troubleshoot issues and capture network traffic in real-time.

ERSPAN configuration is a process that

- enables troubleshooting of Encapsulated Remote Switched Port Analyzer (ERSPAN) issues using debug commands,
- captures and displays network traffic in real-time on specific network device interfaces, and
- uses specific debug commands to monitor and analyze network traffic.

Available debug commands

You can use these commands to configure ERSPAN:

- debug monitor ERSPAN
- debug monitor capture

Configure an ERSPAN source session

Configure an IPv4 ERSPAN source session to define session configuration parameters and monitor specific ports or VLANs.

Configure an ERSPAN source session to enable monitoring of specific ports or VLANs by defining session parameters and traffic direction settings.

The ERSPAN source session defines the session configuration parameters and identifies the ports or VLANs to monitor. Complete the steps below to configure an IPv4 ERSPAN source session.

Procedure

1. Use the **enable** command to enter privileged EXEC mode.

   Example

   ```
   Device> enable
   ```

2. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

3. Use the **monitor session** *span-session-number* **type ERSPAN-source** command to create the ERSPAN source session and enter ERSPAN source session configuration mode.

   Example

   ```
   Device(config)# monitor session 1 type erspan-source
   ```

   - The *span-session-number* argument range is from 1 to 66. The session number must be unique across both source and destination sessions. After the session ID and session type are configured, use the **no** form of this command to remove the session and then re-create the session, with a new session ID or a new session type.

4. (Optional) Use the **description** *string* command to describe the ERSPAN source session. Use the [**no**] **header-type 3** command to configure Type-III ERSPAN header, if required.

   Example

   ```
   Device(config-mon-erspan-src)# description source1
   Device(config-mon-erspan-src)# header-type 3
   ```

   The description string can be up to 240 characters long, excluding spaces or special characters. The default ERSPAN header type is Type II

5. Use the **source** {**interface** *interface-type interface-number* | **vlan** *vlan-id*} [**,** | **-** | **both** | **rx** | **tx**] command to configure the source interface or VLAN and the traffic direction to monitor.

   Example

   ```
   Device(config-mon-erspan-src)# source interface GigabitEthernet1/0/1 rx
   ```

6. (Optional) Use the **filter** {**IP access-group** {*standard-access-list* | *expanded-access-list* | *acl-name* } | **mac access-group** *acl-name* | **sgt** *sgt-id* [**,** | **-**] | **vlan** *vland-id* [**,** | **-**]} command to configure filtering for the ERSPAN source session.

   Example

   ```
   Switch(config-mon-erspan-src)# filter vlan 3
   ```

   (Optional) Configures source VLAN filtering when the ERSPAN source is a trunk port.

   > **Note**
   >
   > You cannot include source VLANs and filter VLANs in the same session.

7. Use the **destination** command to enter ERSPAN source session destination configuration mode. Then use the **erspan-id** *erspan-flow-id* command to configure the ERSPAN flow ID.

   Example

   ```
   Device(config-mon-erspan-src)# destination
   Device(config-mon-erspan-src-dst)# erspan-id 100
   ```

   The ERSPAN ID identifies the ERSPAN traffic and must match the ERSPAN ID configured on the destination session.

   Optional settings include ip dscp dscp-value, ip ttl ttl-value, and vrf vrf-id. Use mtu mtu-size to configure truncation; the MTU range is 176 to 9000 bytes, and the default is 9000 bytes. Use origin ip address ip-address to configure the source IP address for the ERSPAN traffic.

8. Use the **ip address ip-address** to configure the destination IP address for ERSPAN traffic. You can also configure optional settings such as **ip dscp** *dscp-value*, **ip ttl** *ttl-value*, and **vrf** *vrf-id*. Use the **MTU** *mtu-size* command to configure truncation. Use the **origin ip-address** *ip-address* command to configure the source IP address for the ERSPAN traffic.

Example

```
Device(config-mon-erspan-src-dst)# ip address 192.0.2.1
Device(config-mon-erspan-src-dst)# ip dscp 10
Device(config-mon-erspan-src-dst)# ip ttl 32
Device(config-mon-erspan-src-dst)# mtu 512
Device(config-mon-erspan-src-dst)# origin ip address 198.51.100.1
Device(config-mon-erspan-src-dst)# vrf 1
```

The MTU size values range from 176 to 9000 bytes, with a default of 9000 bytes.

9. Use the **exit** , **no shutdown**, and **end** commands to exit destination configuration mode, enable the ERSPAN session, and return to privileged EXEC mode.

Example

```
Device(config-mon-erspan-src-dst)# exit
Device(config-mon-erspan-src)# no shutdown
Device(config-mon-erspan-src)# end
```

Configure an ERSPAN destination session

Configure an ERSPAN destination session to define session configuration parameters and ports that receive monitored traffic.

Configure an ERSPAN destination session to specify which ports receive monitored traffic, so you can monitor and analyze your network.

The ERSPAN destination session defines the session configuration parameters and the ports that receive the monitored traffic. To define an IPv4 ERSPAN destination session, complete this procedure:

Procedure

1. Use the **enable** command to enter privileged EXEC mode.

Example

```
Device> enable
```

2. Use the **configure terminal** command to enter global configuration mode.

Example

```
Device# configure terminal
```

3. Use the **monitor session** *session-number* **type ERSPAN-destination** command to define an ERSPAN destination session using the session ID and the session type, and enter ERSPAN monitor destination session configuration mode.

Example

```
Device(config)# monitor session 1 type erspan-destination
```

- The *session-number* argument range is from 1 to 66. Each session number must be unique and cannot be reused.

- Each session ID must be globally unique for both source sessions and destination sessions, because they share the same global ID space.
- After entering the session ID and session type, you cannot change them. To modify these settings, use the **no** form of this command and then create a new session with the desired session ID or session type.

4. Use the **description** *string* command to describe the ERSPAN destination1 session.

   Example

   ```
   Device(config-mon-erspan-dst)# description source1
   ```

   Enter a *string* up to 240 characters without special characters or spaces.

5. Use the **destination interface** *interface-type interface-number* command to assign the destination interface to the ERSPAN session.

   Example

   ```
   Device(config-mon-erspan-dst)# destination interface GigabitEthernet1/0/1
   ```

6. Use the **source** command to enter ERSPAN destination session source configuration mode.

   Example

   ```
   Device(config-mon-erspan-dst)# source
   ```

7. Use the **erspan-id** *erspan-flow-id* command to configure the ID used by source and destination sessions to identify ERSPAN traffic. You must also enter this ID in the ERSPAN source session configuration.

   Example

   ```
   Device(config-mon-erspan-dst-src)# erspan-id 100
   ```

8. Use the **ip address** *ip-address* **force**] command to configure the IP address that is used as the destination of the ERSPAN traffic.

   Example

   ```
   Device(config-mon-erspan-dst-src)# ip address aaa.bbb.ccc
   ```

   - This IP address must be an address on a local interface or loopback interface, and match the address on the destination switch.
   - The **ip address** *ip-address* **force** command changes the destination IP address for all ERSPAN destination sessions.

   (Optional) Use the **vrf** *vrf-id* command to configure the VRF name instead of the global routing table.

   Example

   ```
   Device(config-mon-erspan-dst-src)# vrf 1
   ```

   Use the **no shutdown** command to enable the configured sessions on an interface

Example

```
Device(config-mon-erspan-dst-src)# no shutdown
```

9. Use the **end** command to exit ERSPAN destination session source configuration mode, and return to privileged EXEC mode.

Example

```
Device(config-mon-erspan-dst-src)# end
```

Configuration examples for ERSPAN

Describes configuration examples for Encapsulated Remote Switched Port Analyzer (ERSPAN) to help network administrators implement traffic monitoring and analysis across Layer 3 networks.

The following sections provide configuration examples for ERSPAN.

ERSPAN source session configuration example

Provides an example of how to configure an ERSPAN source session on a device.

This example shows how to configure an ERSPAN source session.

```
Device> enable
Device# configure terminal
Device(config)# monitor session 1 type erspan-source
Device(config-mon-erspan-src)# description source1
Device(config-mon-erspan-src)# source interface GigabitEthernet 1/0/1 rx
Device(config-mon-erspan-src)# source interface GigabitEthernet 1/0/4 - 8 tx
Device(config-mon-erspan-src)# source interface GigabitEthernet 1/0/3
Device(config-mon-erspan-src)# destination
Device(config-mon-erspan-src-dst)# erspan-id 100
Device(config-mon-erspan-src-dst)# ip address 10.1.0.2
Device(config-mon-erspan-src-dst)# ip dscp 10
Device(config-mon-erspan-src-dst)# ip ttl 32
Device(config-mon-erspan-src-dst)# mtu 512
Device(config-mon-erspan-src-dst)# origin ip address 10.10.0.1
Device(config-mon-erspan-src-dst)# vrf monitoring
Device(config-mon-erspan-src-dst)# exit
Device(config-mon-erspan-src)# no shutdown
Device(config-mon-erspan-src)# end
```

Configure an ERSPAN destination session

Provides examples of how to configure an ERSPAN destination session with and without source VRF configuration.

This section provides configuration examples for ERSPAN destination sessions, including basic configuration and configuration with source VRF.

Basic ERSPAN destination session configuration

This example shows how to configure an ERSPAN destination session:

```
Device(config)# monitor session 1 type erspan-destination
Device(config-mon-erspan-dst)# destination interface GigabitEthernet1/0/11
```

```
Device(config-mon-erspan-dst)# destination interface GigabitEthernet1/0/1
Device(config-mon-erspan-dst)# source
Device(config-mon-erspan-dst-src)# erspan-id 100
Device(config-mon-erspan-dst-src)# ip address 10.1.0.2
```

ERSPAN destination session with source VRF

This example shows how to configure a source VRF for an ERSPAN destination session:

```
Device(config)# monitor session 1 type erspan-destination
Device(config-mon-erspan-dst)# destination interface GigabitEthernet1/0/11
Device(config-mon-erspan-dst)# destination interface GigabitEthernet1/0/1
Device(config-mon-erspan-dst)# source
Device(config-mon-erspan-dst-src)# erspan-id 100
Device(config-mon-erspan-dst-src)# ip address 10.1.0.2
Device(config-mon-erspan-dst-src)# vrf 1
```

Verify ERSPAN

Provides commands and sample outputs to verify ERSPAN configuration and capability features on network devices.

To verify the ERSPAN configuration, use these commands.

This is sample output from the **show monitor session** command:

```
Device# show monitor session 53

Session 53
----------
Type                    : ERSPAN Source Session
Status                  : Admin Enabled
Source Ports            :
MTU                     : 9000
```

This is sample output from the **show platform software monitor session** command:

```
Device# show platform software monitor session 53

Span Session 53 (FED Session 0):
Type: ERSPAN Source
Prev type: Unknown
Ingress Src Ports:
Egress Src Ports:
Ingress Local Src Ports: (null)
Egress Local Src Ports: (null)
Destination Ports:
Ingress Src Vlans:
Egress Src Vlans:
Ingress Up Src Vlans: (null)
Egress Up Src Vlans: (null)
Src Trunk filter Vlans:
RSPAN dst vlan: 0
RSPAN src vlan: 0
RSPAN src vlan sav: 0
Dest port encap = 0x0000
Dest port ingress encap = 0x0000
Dest port ingress vlan = 0x0
SrcSess: 1 DstSess: 0 DstPortCfgd: 0 RspnDstCfg: 0 RspnSrcVld: 0
DstCliCfg: 0 DstPrtInit: 0 PsLclCfgd: 0
```

```
Flags: 0x00000000
Remote dest port: 0 Dest port group: 0
FSPAN disabled
FSPAN not notified
ERSPAN Id : 0
ERSPAN Org Ip: 0.0.0.0
ERSPAN Dst Ip: 0.0.0.0
ERSPAN Ip Ttl: 255
ERSPAN DSCP : 0
ERSPAN MTU : 1500 >>>>
ERSPAN VRFID : 0
ERSPAN State : Disabled
ERSPAN Tun id: 61
ERSPAN header-type: 2
ERSPAN SGT :
```

This is sample output from the **show monitor session ERSPAN-source detail** command:

```
Device# show monitor session erspan-source detail

Type                      : ERSPAN Source Session
Status                    : Admin Enabled
Description               : -
Source Ports              :
    RX Only               : None
    TX Only               : None
    Both                  : None
Source Subinterfaces      :
    RX Only               : None
    TX Only               : None
    Both                  : None
Source VLANs              :
    RX Only               : None
    TX Only               : None
    Both                  : None
Source Drop-cause         : None
Source EFPs               :
    RX Only               : None
    TX Only               : None
    Both                  : None
Source RSPAN VLAN         : None
Destination Ports         : None
Filter VLANs              : None
Filter SGT                : None
Dest RSPAN VLAN           : None
IP Access-group           : None
MAC Access-group          : None
IPv6 Access-group         : None
Filter access-group  :None
smac for wan interface    : None
dmac for wan interface    : None
Destination IP Address    : 192.0.2.1
Destination IPv6 Address  : None
Destination IP VRF        : None
MTU                       : 1500
Destination ERSPAN ID     : 251
Origin IP Address         : 10.10.10.216
Origin IPv6 Address       : None
IP QOS PREC               : 0
IPv6 Flow Label           : None
IP TTL                    : 255
```

```
ERSPAN header-type       : 3
```

This output from the **show capability feature monitor ERSPAN-source** command displays information about the configured ERSPAN source sessions:

```
Device# show capability feature monitor erspan-source

ERSPAN Source Session:ERSPAN Source Session Supported: TRUE
No of Rx ERSPAN source session: 8
No of Tx ERSPAN source session: 8
ERSPAN Header Type supported: II and III
ACL filter Supported: TRUE
SGT filter Supported: TRUE
Fragmentation Supported: TRUE
Truncation Supported: FALSE
Sequence number Supported: FALSE
QOS Supported: TRUE
```

This output from the **show capability feature monitor ERSPAN-destination** command displays all the configured global built-in templates:

```
Device# show capability feature monitor erspan-destination

ERSPAN Destination Session:ERSPAN Destination Session Supported: TRUE
Maximum No of ERSPAN destination session: 8
ERSPAN Header Type supported: II and III
```

Additional references

Provides links to relevant RFCs and technical assistance resources for Cisco products and technologies.

This section provides links to relevant RFCs and technical assistance resources for Cisco products and technologies.

RFCs

Standard or RFC	Title
RFC 2784	Generic Routing Encapsulation (GRE)

Technical assistance

Description	Link
Use the Cisco support website to access documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. Subscribe to services such as the Product Alert Tool, the Cisco Technical Services Newsletter, and RSS Feeds to receive security and technical information about your products. Sign in with your Cisco.com user ID and password to access most tools on the Cisco Support website.	http://www.cisco.com/support

Feature history for ERSPAN

Provides release information about ERSPAN configuration features and platform support details.

The table lists the first software release in each release train that supports each feature. All later releases in the same train usually support the feature unless stated.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, refer www.cisco.com/go/cfn. You do not need an account on Cisco.com.

Feature information for ERSPAN

Feature Name	Releases	Feature Information
ERSPAN	Cisco IOS XE Cupertino 17.7.1	This feature was introduced for Cisco Catalyst IE9300 Rugged Series Switches.

2 Swap Drive

Topics:

Swap Drive

This topic explains how the swap drive feature enables quick replacement of a failed switch by using an external SD card or USB drive to restore the original switch's image and configuration.

The Swap Drive feature is a disaster recovery mechanism that allows you to replace a failed switch by transferring an external secure digital (SD) card or USB drive containing a backup of the original switch to a new switch. It

- restores the same image and configuration to the new switch as the original switch, minimizing downtime.
- requires prior synchronization of the original switch with the SD card or USB drive, and
- supports SD cards formatted with DOSFS and USB 2.0 drives and does not support third-party SD or SDHC cards.

Switch recovery with Swap Drive

Starting from release Cisco IOS XE Cupertino 17.9.1, the swap drive feature is available for Cisco Catalyst IE9300 Rugged Series Switches.

Each Cisco Catalyst IE9300 Rugged Series Switch has a SD card slot and a USB-A port.

For information about the switch SD card slot and USB port, refer to the Cisco Catalyst IE9300 Rugged Series Switch Hardware Installation Guide.

Although synchronization can be done manually at any time, it is recommended to configure automatic synchronization at set times. When you request synchronization either manually or automatically, the switch checks for discrepancies between the internal flash drive, SD card or the USB drive.

> **Note**
>
> To restore settings to the new switch, you must have previously synchronized the original switch with the SD card or USB drive.

Swap Drive procedures and CLI commands

For Swap Drive instructions, refer to the Swap the External Drive.

For CLI commands, refer to the Swap Drive CLI Commands.

Guidelines for using SD cards and USB drives

Outlines best practices for using SD cards and USB drives with the switch.

- The switch uses the Disk Operating System Filing System (DOSFS) to format SD cards.
- The switch supports only Cisco-approved SD cards and excludes third-party SD cards or SD High Capacity (SDHC) cards.
- The switch reads data from a write-protected SD card during the boot process but prevents any updates to it.
- You can use a USB 2.0 drive as an alternative to an SD card for the swap drive feature.

How Swap Drive works

Describes the process by which swap drives enable migration of system images and configurations between switches, ensuring efficient replacement or deployment.

This process ensures that system images and configurations are preserved and can be easily transferred between devices, facilitating efficient switch replacement or deployment.

Summary

Swap drives enable the migration of system images and configurations between switches by utilizing external storage media. The key components involved in the process are:

- **Switch**: The source or target device for system data.
- **External storage**: An SD card or USB drive that stores the system image and configuration.

Workflow

The process involves these stages:

1. Backup: The system duplicates the entire configuration and image to the external SD card or USB drive. You can trigger this operation using a CLI command for one-time or periodic synchronization. If the switch has already been backed up, only changes made since the last backup are duplicated to the external drive.
2. Recovery: Restore occurs automatically when the external drive is inserted into the new switch and powered on. The switch scans for an SD card with image and configuration.
   - If the image and configuration are present on the SD card, the switch then copies them to internal flash and comes up with the image and configuration of the original system.
   - If an SD card is not present or does not have the original switch's image and configuration, the switch checks the USB drive.
   - If the image and configuration are present on the USB drive, the switch comes up with the original switch's image and configuration.
3. Initial configuration: If the image and configuration are not present either in SD card or USB, the system prompts the user to continue with initial configuration.

Swap the external drive

Learn how to swap the external SD card or USB drive between switches to recover configuration and image after a hardware failure.

You can remove the SD card from the failed switch, insert it into the new switch, and power on the new switch. A new switch has no startup configuration file because it has never been deployed.

Before you begin

Power off the switch, transfer the SD card or USB drive, and connect the network cables.

Procedure

1. On the failed switch, remove the SD card or USB drive.

> **Note**
>
> The SD card and USB drive are hot-swappable. However, do not remove either device from the switch while sdflash write is in progress.

2. On the new switch, ensure that the SD card or USB drive is oriented properly, and then press it into the slot on the switch until it is seated.
3. Power on the new switch.

   The image and configuration of the failed switch are transferred to the new one.

Swap Drive commands

Lists and describes Swap Drive commands for synchronizing, configuring, and checking switch images and configurations.

Use this reference to understand command functions and requirements.

> **Note**
>
> Run all **auto sync** commands in configuration mode.

Table 1: Swap Drive commands and their descriptions

Command	Description
• **sync sdflash:** • **sync usbflash1:**	Synchronizes the switch image and configuration files from internal flash to the SD or USB drive.
• **sync sdflash: ios-image** • **sync usbflash1: ios-image**	Synchronizes the switch image from the internal flash to the SD or USB drive.
• **sync sdflash:skip config** • **sync usbflash1: skip config**	Synchronizes the switch image from the internal flash to the SD or USB drive but does not sync the configuration.
• **sync sdflash:skip ios-image** • **sync usbflash1: skip ios-image**	Synchronizes the configuration files from the internal flash to the SD or USB drive but does not sync the image.
sync restore-bundle	Copies the bundle image to the new switch without installing it during the restore process.

Command	Description
[no] auto sync enable	Enables or disables the auto sync feature. **Note** Auto synchronization is disabled by default. You must enable auto sync before using other options for this feature.
auto sync config [usbflash1|sdflash]:	Specifies the configuration that runs during synchronization.
[no] auto sync run time: [hh:mm:ss]	Specifies the time at which the synchronization is performed. By default, the time is set to 00:00:00.
show sync status	Displays the last synchronization time and status. **Note** If a type-6 password is configured, the status displays the configuration as out of sync and includes a message stating that type-6 passwords are not synced.
show auto sync configuration	Displays all the configuration settings.
show auto sync status	Displays the last auto synchronization time and status.

3 VLAN Mapping

Topics:

VLAN mappings

This topic explains how VLAN mapping translates customer VLAN IDs into service-provider VLAN IDs on Cisco Catalyst IE9300 Rugged Series Switches. VLAN mapping enables VLAN ID reuse across a shared backbone while maintaining secure traffic isolation for dispersed customer sites.

A VLAN mapping is a Layer 2 feature that translates customer VLAN IDs (VLAN ID assigned by the customer) into service VLAN IDs (VLAN ID assigned by the service provider) on trunk ports connected to your network. The VLAN mapping:

- enables service providers to reuse of VLAN IDs across shared backbones by maintaining traffic isolation, and
- requires that all features on mapped ports reference the S-VLAN instead of the original C-VLAN.

Typical deployment scenario

In a typical deployment, service providers want to provide a transparent switching infrastructure where customers' remote switches function as part of the local site. Customers can use the same VLAN ID space and seamlessly run Layer 2 control protocols across the provider network. Service providers should not impose their VLAN IDs on customers.

Service providers internal VLAN assignments might conflict with a customer VLANs. VLAN mapping solves this issue by translating customer VLANs into different service provider VLANs as traffic travels through the provider's network.

VLAN mapping operations and behaviors

VLAN mapping is supported on all models of Cisco Catalyst IE9300 Rugged Series Switches with `Network Essentials` or `Network Advantage` licenses. You must reference the S-VLAN instead of the original C-VLAN when configuring any feature on a VLAN-mapped port. A service provider’s internal assignments might conflict with a customer’s VLAN. To isolate customer traffic, a service provider can map a specific VLAN to another while the traffic is in its cloud.

When packets enter a port configured for VLAN mapping, the switch maps the specified C-VLAN to the specified S-VLAN based on the port number and the packet's original C-VLAN. All forwarding operations on the switch use S-VLAN information, not C-VLAN information, because the VLAN ID is mapped to the S-VLAN at ingress. When packets exit the port, symmetrical mapping back to the customer C-VLAN occurs automatically.

> **Note**
>
> Always configure features on a VLAN-mapped port with the S-VLAN instead of the customer VLAN ID (C-VLAN). One-to-one VLAN mapping is not supported.

VLAN mapping deployment

For example, if customer A and customer B use the same VLANs (such as VLAN 10) at multiple sites on different sides of a service provider network, you can map the customer VLAN IDs to unique service provider VLAN IDs (such as VLAN 100 for customer A, VLAN 200 for customer B) for packet travel across the backbone. The original customer VLAN IDs are restored at the other side of the service provider backbone for use in the other customer site. Configure the same set of VLAN mappings at customer-connected ports on each side of the service provider network.

Selective Q-in-Q

This topic explains the Selective Q-in-Q feature, which enables tagging of only specified customer VLANs with a service-provider VLAN ID at the user-network interface, allowing double-tagged packet traversal while preserving the original customer tag.

Selective Q-in-Q is a VLAN mapping feature. This feature tags only specified customer VLANs (C-VLANs) with a service-provider VLAN ID (S-VLAN) at the user-network interface (UNI). As a result, packets traverse the provider network with double tagging and retain the original customer tag.

- Maps configured customer VLANs at the UNI to the specified S-VLAN and adds the S-VLAN tag while leaving the C-VLAN tag intact.
- Drops any packets that do not match the configured customer VLAN list.
- Removes the S-VLAN tag at egress and forwards the packet with the original customer VLAN ID preserved.

Q-in-Q on a trunk port

This topic explains the Q-in-Q VLAN stacking mode on trunk ports, where all customer VLANs entering the UNI are tagged with the same service-provider VLAN ID, enabling double-tagged traffic across the provider network.

Q-in-Q on trunk ports is a VLAN stacking mode that tags every customer VLAN entering the UNI with the same service-provider VLAN ID (S-VLAN), so all customer traffic crosses the provider cloud double-tagged. The switch removes the S-VLAN tag at egress so the customer receives frames with their original VLAN IDs.

Configuration guidelines for VLAN mapping

Use these guidelines when configuring VLAN mapping on Cisco Catalyst IE9300 switches to ensure correct traffic isolation and feature compatibility. This topic describes requirements for EtherChannels, S-VLAN setup, and control traffic processing, while highlighting restrictions for Selective QnQ and trunk port operations.

Guidelines for VLAN mapping

- By default, no VLAN mapping is configured. Enable VLAN mapping only on the port-channel (EtherChannel) interface, not on individual member ports.
- If the VLAN mapping is enabled on an EtherChannel, the configuration applies only to the EtherChannel interface.
- If the VLAN mapping is enabled on an EtherChannel and a conflicting mapping is enabled on a member port, the port is removed from the EtherChannel.
- Changing a member port's mode of an EtherChannel from trunk removes it from the EtherChannel bundle.
- Enable Layer 2 protocol tunneling or insert a BPDU filter for spanning tree to ensure consistent control traffic processing. For configurations, see Layer 2 NAT commands
- Do not use default or user-configured native VLANs, or reserved VLANs in the 1002 to 1005 range, for VLAN mapping.
- Private VLAN (PVLAN) is not supported when VLAN mapping is configured.

Guidelines for selective Q-in-Q:

- Create the S-VLAN and add it to the trunk's allowed VLAN list before enabling Selective Q-in-Q.
- When Selective Q-in-Q is enabled, Layer 2 protocol tunneling is supported for CDP, STP, LLDP, and VTP.
- IP routing and IPSG are not supported on Selective Q-in-Q enabled ports.

Guidelines for Q-in-Q on a trunk port:

- Create the S-VLAN and add it to the trunk's allowed VLAN list before enabling Q-in-Q on the trunk port.
- When Q-in-Q is enabled on a trunk port, Layer 2 protocol tunneling is supported for CDP, STP, LLDP, and VTP.
- Ingress SPAN, egress SPAN, and RSPAN are supported on trunk ports with Q-in-Q enabled.
- SPAN filtering can be enabled to monitor only the traffic on the mapped VLAN (S-VLANs).

- IGMP snooping is not supported on the C-VLAN when Q-in-Q is configured.

Mapping customer VLANs to service-provider VLANs

This process describes how VLAN mapping translates customer VLAN IDs into unique service-provider IDs to resolve conflicts in multi-tenant environments. It outlines the stages from ingress mapping to egress restoration, ensuring traffic isolation while preserving original customer VLAN configurations across the backbone.

In multi-tenant environments, different customers often use overlapping VLAN IDs across geographically distributed sites. When these sites connect through a service provider network, VLAN ID conflicts can occur. VLAN mapping resolves this by translating customer VLAN IDs to unique service provider VLAN IDs as traffic traverses the backbone.

Summary

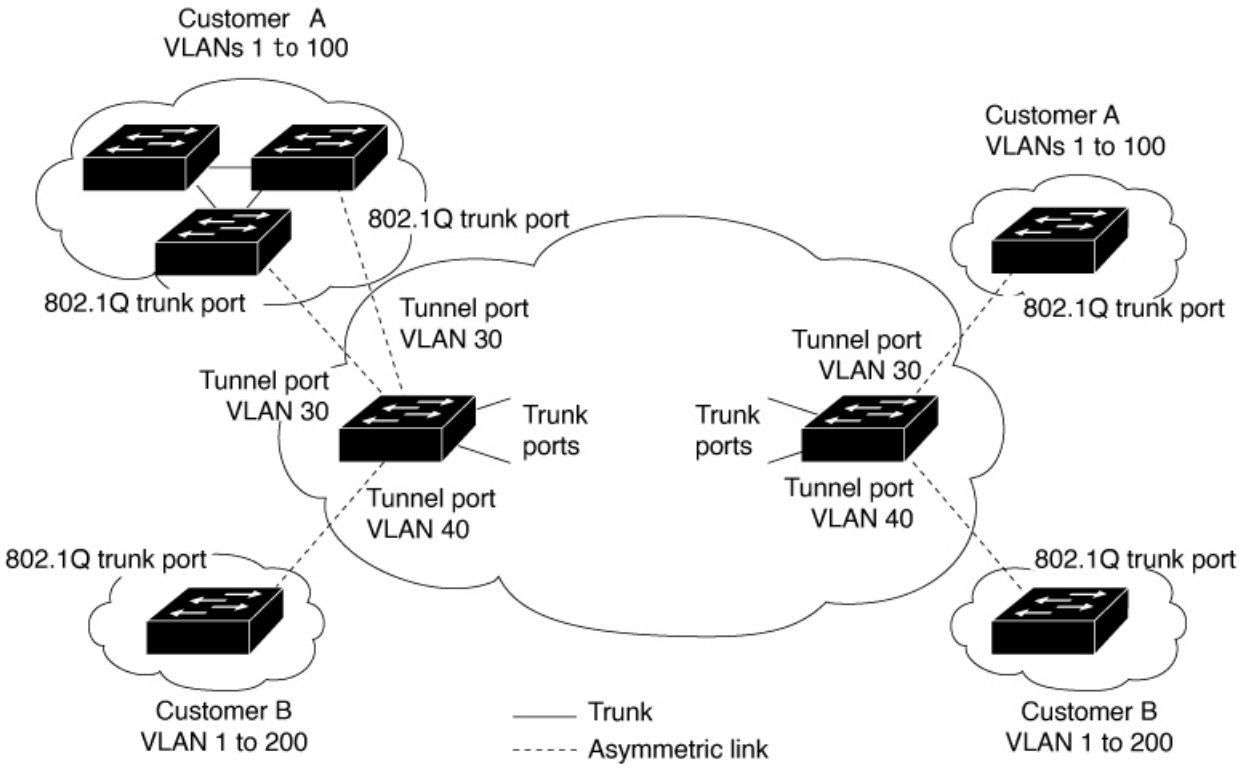
The key components in the VLAN mapping process are:

- **Customer sites**: Network locations that use customer-specific VLAN IDs.
- **Service-provider network**: The backbone infrastructure that transports traffic between customer sites.
- **Customer-connected ports**: Edge interfaces on the service provider network where VLAN mapping occurs.

The VLAN mapping process translates customer VLAN IDs to service-provider VLAN IDs at ingress, transports the traffic across the backbone using provider VLANs, and restores the original customer VLAN IDs at egress to maintain connectivity between customer sites.

Workflow

Figure 2: QnQ topology



The VLAN mapping process in a service-provider network occurs in several stages.

1. · Traffic from customer A enters the service provider network through a customer-connected port at Site 1.

- The ingress port maps the customer VLAN ID to a unique service provider VLAN ID.
- The packet traverses the service provider backbone using the mapped service-provider VLAN ID.
- At the egress customer-connected port at Site 2, the service provider VLAN ID is mapped back to the original customer VLAN ID.
- The traffic reaches customer A's destination site with the original VLAN ID intact.
- The same mapping process occurs for customer B's traffic, using a different set of service provider VLAN IDs to prevent conflicts.

Configure selective Q-in-Q on a trunk port

Learn how to configure VLAN mapping for selective Q-in-Q on a trunk port. This procedure helps you map customer VLAN IDs to service provider VLAN IDs, set default forwarding for unmapped packets, and verify your configuration.

You can enable selective Q-in-Q VLAN mapping on a trunk port to manage customer VLAN traffic and differentiate it from service provider VLAN traffic.

You can map specific customer VLAN IDs (C-VLANs) to provider VLAN IDs (S-VLANs) on a trunk port using selective Q-in-Q. This configuration enables flexible service provider isolation and delivery for multiple customers.

> **Note**
>
> You cannot configure both one-to-one mapping and selective Q-in-Q on the same interface.

Before you begin

Ensure you have identified the interface connected to the service provider network.

- Review your desired C-VLAN and S-VLAN mapping requirements.
- You cannot configure one-to-one mapping and selective Q-in-Q on the same interface.

Perform these steps to configure selective Q-in-Q mapping.

Procedure

1. Configuration
   a) Use the **enable** command to enter privileged EXEC mode.

      Example

      ```
      Switch> enable
      ```

      If prompted, enter your password.

   b) Use the **configure terminal** command to enter global configuration mode.

      Example

      ```
      Switch# configure terminal
      ```

c) Use the **interface** *interface-id* command to enter interface configuration mode for the interface connected to the service provider network.

Example

```
Switch(config)# interface gigabitethernet1/0/1
```

interface-id: Enter a physical interface or an EtherChannel port channel.

d) Use the **switchport mode trunk** command to configure the interface as a trunk port.

Example

```
Switch(config-if)# switchport mode trunk
```

e) Use the **switchport vlan mapping** *vlan-id* **dot1q-tunnel** *outer-vlan-id* command to enter the VLAN IDs to be mapped.

Example

```
Switch(config-if)# switchport vlan mapping 16 dot1q-tunnel 64
```

- *vlan-id* : The customer VLAN ID (C-VLAN) entering the switch from the customer network. The range is from 1 to 4094. Enter a string of VLAN IDs.
- *outer-vlan-id*: The outer VLAN ID (S-VLAN) of the service provider network. The range is from 1 to 4094.

Use the **no** form of this command to remove the VLAN mapping configuration. Entering the **no switchport vlan mapping all** command deletes all mapping configurations.

f) Use the **switchport vlan mapping default dot1q-tunnel** *vlan-id* command to specify that all unmapped packets on the port are forwarded with the specified S-VLAN.

Example

```
Switch(config-if)# switchport vlan mapping default dot1q-tunnel 22
```

By default, packets that do not match the mapped VLANs are dropped.

The system forwards the untagged traffic.

g) Use the **spanning-tree bpdufilter enable** command to insert a BPDU filter for spanning tree.

Example

```
Switch(config-if)# spanning-tree bpdufilter enable
```

Note

To process control traffic consistently, you can enable Layer 2 protocol tunneling or insert a BPDU filter for spanning tree.

h) Use the **exit** command to return to global configuration mode.

Example

```
Switch(config-if)# exit
```

i) Use the **end** command to return to privileged EXEC mode.

Example

```
Switch(config)# end
```

2. Verification

a) (Optional) Use the **show interfaces** *interface-id* **vlan mapping** command to verify the configuration.

Example

```
Switch# show interfaces gigabitethernet1/0/1 vlan mapping
```

b) (Optional) Use the **copy running-config startup-config** command to save your entries in the configuration file.

Example

```
Switch# copy running-config startup-config
```

Configure Q-in-Q on a trunk port

Learn how to configure VLAN mapping for Q-in-Q on a trunk port to forward all customer packets with a specific service provider VLAN. This procedure describes how to set up the trunk interface, apply the mapping, and verify the results.

You can configure Q-in-Q VLAN mapping on a trunk port to encapsulate customer VLAN packets with a designated service provider VLAN tag.

Use this task to configure selective Q-in-Q VLAN mapping on a trunk interface. This setup encapsulates all customer VLAN packets with a designated provider VLAN tag. Service provider networks typically require this feature to separate customer traffic and apply unified service policies.

Before you begin

Ensure that you know the service provider VLAN ID (S-VLAN) to use for mapping.

Perform these steps to configure Q-in-Q on a trunk port.

Procedure

1. Use the **enable** command to enter privileged EXEC mode.

Example

```
Switch> enable
```

Enter your password if prompted.

2. Use the **configure terminal** command to enter global configuration mode.

Example

```
Switch# configure terminal
```

3. Use the **interface** *interface-id* command to enter interface configuration mode for the interface connected to the service provider network.

   Example

   ```
   Switch(config)# interface gigabitethernet1/0/1
   ```

   You can enter either a physical interface or an EtherChannel port channel.

4. Use the **switchport mode trunk** command to configure the interface as a trunk port.

   Example

   ```
   Switch(config-if)# switchport mode trunk
   ```

5. Use the **switchport vlan mapping default dot1q-tunnel** *vlan-id* command to specify that all unmapped C-VLAN packets on the port are forwarded with the specified S-VLAN.

   Example

   ```
   Switch(config-if)# switchport vlan mapping default dot1q-tunnel 16
   ```

6. Use the **spanning-tree bpdufilter enable** command to insert a BPDU filter for spanning tree.

   Example

   ```
   Switch(config-if)# spanning-tree bpdufilter enable
   ```

   Note

   To process control traffic consistently, either enable Layer 2 protocol tunneling (recommended) or insert a BPDU filter for spanning tree.

7. Use the **exit** command to exit the interface configuration mode.

   Example

   ```
   Switch(config-if)# exit
   ```

8. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Switch(config)# end
   ```

9. (Optional) Use the **show interfaces** *interface-id* **vlan mapping** command to verify the configuration.

Example

```
Switch# show interfaces gigabitethernet1/0/1 vlan mapping
```

Use the **copy running-config startup-config** command to save your entries in the configuration file.

Configure selective Q-in-Q mapping on the port

Learn how to configure selective Q-in-Q mapping on the port.

Map specific customer VLAN (C-VLAN) IDs to a service provider VLAN (S-VLAN) ID on a port.

Selective Q-in-Q mapping allows you to tunnel specific traffic through the switch. By default, traffic from any VLAN ID that you do not map is dropped. If you configure a default S-VLAN ID, traffic from all other VLANs is forwarded.

Before you begin

Identify the C-VLAN range and the target S-VLAN ID for the tunnel.

Perform these steps to configure selective Q-in-Q mapping.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Switch# configure terminal
   ```

2. Use the **interface GigabitEthernet** *<interface_id>* command to enter interface configuration mode for the target port.

   Example

   ```
   Switch(config)# interface GigabitEthernet 1/0/1
   ```

3. Use the **switchport vlan mapping dot1q-tunnel** *channel-id* command to map a range of IDs to a specific S-VLAN ID to enable selective Q-in-Q.

   Example

   ```
   Switch(config-if)# switchport vlan mapping 2-5 dot1q-tunnel 100
   ```

4. (Optional) Use the **switchport vlan mapping default dot1q-tunnel** *s-vlan-id* command to configure a default S-VLAN ID to forward traffic from all other VLANs.

   Example

   ```
   Switch(config-if)# switchport vlan mapping default dot1q-tunnel 100
   ```

> **Note**
>
> If you do not configure a default S-VLAN ID, all traffic outside the specified VLAN range is dropped.

5. Use the **exit** command to return from interface configuration mode to global configuration mode.

   Example

   ```
   Switch(config-if)# exit
   ```

6. Use the **end** command to return to return to privileged EXEC mode.

   Example

   ```
   Switch(config)# end
   ```

7. (Optional) Use the **show vlan mapping** command to verify the VLAN mapping configuration on the interface.

   Example

   ```
   Switch# show vlan mapping
   Total no of vlan mappings configured: 5
   Interface Hu1/0/50:
   VLANs on wire                    Translated VLAN     Operation
   -------------------------------  ---------------     --------------
   2-5                                   100            selective QinQ
   *                                     200            default Q
   ```

Feature history for VLAN mapping

Provides release and related information for VLAN mapping features explained in this chapter.

This table provides release and related information for features explained in this chapter. These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE 17.13.1	Selective Q-in-Q Q-in-Q on a trunk port	Support for these features was introduced.

4 Flexible Network Flow

Topics:

Flexible NetFlow

This topic explains the Flexible NetFlow feature, which provides statistics for accounting, network monitoring, and planning by using flows and customizable flow records.

Flexible NetFlow is a network feature that uses flows to provide statistics for accounting, network monitoring, and network planning.

- A flow is a unidirectional stream of packets with the same key values arriving on a source interface.
- A key is an identified value for a field within the packet, and you create a flow using a flow record to define unique keys.
- Flexible NetFlow allows you to define optimal flow records by selecting keys from a large collection of predefined fields.

Flexible NetFlow key features

Flexible NetFlow supports enhanced network anomalies and security detection, and enables export of flow data to remote collectors.

- All key values must match for a packet to be counted in a flow.
- Flows may gather additional fields depending on the export record version configured.
- Flows are stored in the Flexible NetFlow cache.
- Data gathered by Flexible NetFlow can be exported using an exporter to a remote system such as a Flexible NetFlow collector.
- The Flexible NetFlow collector can use an IPv4 address or an IPv6 address.
- The size of data collected for a flow is defined using a monitor, which combines the flow record and exporter with cache information.
- Starting with Cisco IOS XE 16.12.1, Source Group Tag (SGT) and Destination Group Tag (DGT) fields over Flexible NetFlow are supported for IPv6 traffic.
- Starting with the Cisco IOS XE 17.12.1 release, Flexible NetFlow version 5 is supported.

Flexible NetFlow benefits

This topic explains the benefits and applications of Flexible NetFlow, including its customizable flow recognition and enhanced security monitoring capabilities.

Flexible NetFlow allows you to define flows, enabling efficient understanding of network behavior with tailored flow information for various services.

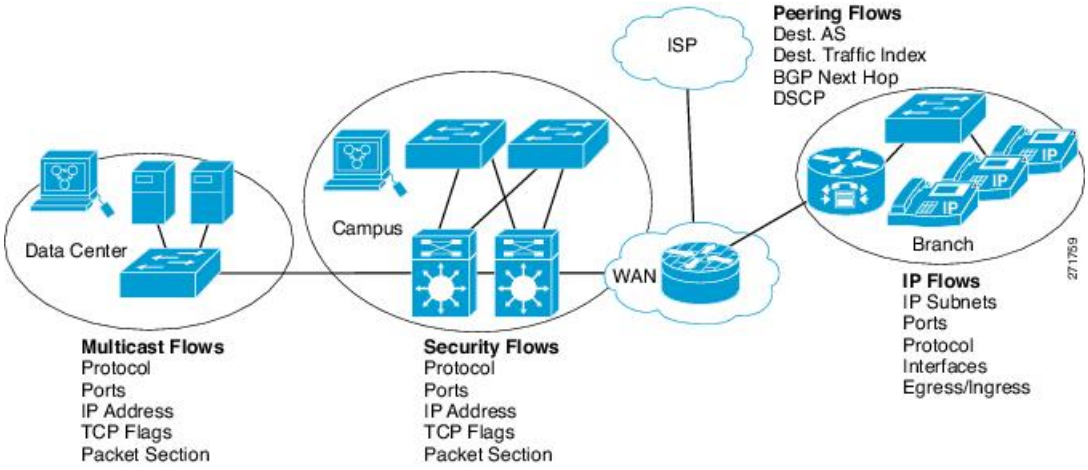
- High capacity flow recognition, including scalability and aggregation of flow information.
- Enhanced flow infrastructure for security monitoring and DDoS detection and identification.
- New information from packets to adapt flow information to a particular service or operation in the network. The flow information available will be customizable by Flexible NetFlow users.
- Extensive use of Cisco’s flexible and extensible NetFlow Version 9 .
- A comprehensive IP accounting feature that can be used to replace many accounting features, such as IP accounting, Border Gateway Protocol (BGP) Policy Accounting, and persistent caches.

Flexible NetFlow allows you to understand network behavior more efficiently, with specific flow information tailored for various services used in the network. The following are some example applications for a Flexible NetFlow feature:

- Flexible NetFlow enhances Cisco NetFlow as a security monitoring tool. For instance, new flow keys can be defined for packet length or MAC address, allowing users to search for a specific kind of attack in the network.
- Flexible NetFlow allows you to quickly identify how much application traffic is being sent between hosts by specifically tracking TCP or UDP applications by the class of service (CoS) in the packets.
- The accounting of traffic entering a Multiprotocol Label Switching (MPLS) or IP core network and its destination for each next hop per class of service. This capability allows the building of an edge-to-edge traffic matrix.

The figure illustrates how Flexible NetFlow might be deployed in a network.

Figure 3: Typical Deployment for Flexible NetFlow



Restrictions for Flexible NetFlow

Use these guidelines when configuring Flexible NetFlow to ensure compatibility, feature support, and correct operation across different interfaces and platforms.

Guidelines

For IPv4 and IPv6 traffic, either Cisco Express Forwarding IPv4 or IPv6 distributed Cisco Express Forwarding must be enabled on your device and on any interfaces on which you want to enable Flexible NetFlow.

Restrictions

Interface and traffic

- Flexible NetFlow is not supported on layer 2 port-channel interfaces, but supported on member ports.
- Flexible NetFlow is supported on Layer 3 port-channel interfaces and their member ports, but not on both simultaneously for the same traffic type and direction.
- Configure Flexible NetFlow on either the port-channel interface or its member ports, but not on both for the same traffic.
- Layer 2, VLAN, layer 3, and SVI interfaces are supported; tunnels are not supported.
- Layer 2, IPv4, and IPv6 traffic types are supported. Multiple flow monitors of different traffic types are allowed per interface or direction.
- A flow monitor cannot be shared across layer 3 physical and logical interfaces but can be shared across logical or physical interfaces.

- Flexible NetFlow export is not supported on Ethernet management port (example: GigabitEthernet 0/0).

Export protocol and flow monitor

- Flexible NetFlow version 9 and version 10 (IPFIX) export formats are supported. If you do not configure an export protocol, Version 9 is used by default.
- Traditional NetFlow accounting is not supported.
- Configuration of IPFIX exporter on an AVC flow monitor is not supported.
- When configuring a flow record with match application name field, the flow monitor will not support a flow sampler.
- The device supports up to 15 flow monitors.

Hardware

- Depending on switch type, one or more forwarding ASICs with multiple cores per ASIC; NetFlow tables per core or ASIC cannot be combined.
- NetFlow hardware supports four hardware samplers with sampling rates from 1 out of 2 to 1 out of 1024. Random and deterministic modes are supported.
- Hash collisions can occur in hardware hash tables. Effective NetFlow table utilization is about 80%.
- A single flow may take two consecutive entries. IPv6 and datalink flows also take two entries, effectively halving the table size.
- The switch can support either one or two cores; each Overflow TCAM supports 256 ingress and 256 egress entries per core.

Flow and data capture

- NetFlow records do not support MPLS-enabled interfaces; data capture based on MPLS label inside MPLS network is not supported.
- Egress flow monitors do not capture flows egressing in EoMPLS mode or L3VPN Per-Prefix mode.
- When Flexible NetFlow and NAT are configured, actual flow details are exported, not translated flow details; ALG flow details are excluded unless translated through CPU.
- For IPv6 flow monitors, Source Group Tag (SGT) and Destination Group Tag (DGT) fields cannot co-exist with MAC address fields.
- When a flow record has only SGT and DGT fields and neither is applicable, the flow is created with zero values for these fields.

QoS and security

- On non-Cisco TrustSec interfaces, SGT value zero means no command header; on Cisco TrustSec interfaces, zero means unknown tag.
- QoS marked packets on ingress interfaces capture QoS value; on egress interfaces, rewritten QoS values on ingress are not captured.

Flow export and timeout

- The flow exporter exports flow data only after the template data timeout period ends. Configuration changes, such as VPN ID or VRF deletion, take effect after the timeout.
- The reported bytes count field bytes long is layer-2-packet-size minus 18 bytes, accurate for classic Ethernet (802.3) only. Use bytes layer2 field for accurate layer 2 packet size.

Flexible NetFlow components

This topic explains the components of Flexible NetFlow and how they can be used together for traffic analysis and data export.

Flexible NetFlow consists of components that can be used together in several variations to perform traffic analysis and data export.

You can use user-defined flow records and the Flexible NetFlow component structure to create custom configurations for traffic analysis and data export on your networking device with fewer configuration commands. You can configure each flow monitor with a unique combination of a flow record, flow exporter, and cache type. When you change a parameter, such as the destination IP address for a flow exporter, the change applies to all flow monitors that use that exporter. You can use the same flow monitor with different flow samplers to sample the same type of network traffic at different rates on various interfaces. You can read more about Flexible NetFlow components in these sections.

Flow records

This topic explains how Flexible NetFlow records store flow data and define the cache used for storing flow information.

Flexible NetFlow records are assigned to Flexible NetFlow flow monitors to define the cache that is used for storing flow data.

In Flexible NetFlow, a combination of key and nonkey fields is called a record. Predefined records are not supported for regular Flexible NetFlow on Cisco Catalyst 9000 Series Switches. Configure a user-defined record unless the selected feature explicitly supports a predefined record.

A flow record defines the keys that Flexible NetFlow uses to identify packets in the flow, as well as other fields of interest that Flexible NetFlow gathers for the flow. You can define a flow record with any combination of keys and fields of interest. The device supports a rich set of keys. A flow record also defines the types of counters gathered per flow. You can configure 64-bit packet or byte counters. The device enables the following match fields as the defaults when you create a flow record:

- **match datalink** –Layer 2 attributes
- **match flow direction** –Specifies a match to the fields identifying the direction of flow.
- **match interface** –Interface attributes
- **match ipv4** –IPv4 attributes
- **match ipv6** –IPv6 attributes
- **match transport** –Transport layer fields
- **match flow cts** –Cisco TrustSec fields

User-defined records in Flexible NetFlow

This topic explains how to define user-defined records for Flexible NetFlow flow monitor cache, including key and nonkey fields, and their application in traffic monitoring and security.

Flexible NetFlow enables you to define records for a flow monitor cache by specifying key and nonkey fields. This customization allows the data collection to meet specific requirements.

When you define your own records for a Flexible NetFlow flow monitor cache, they are referred to as *user-defined records*. The values in nonkey fields add extra information about the traffic in flows, and changes in nonkey fields do not create new flows. Typically, nonkey field values come from the first packet in the flow. Flexible NetFlow allows capturing counter values such as the number of bytes and packets in a flow as nonkey fields.

Flexible NetFlow adds a new version 9 export format field type for the header and packet section types. Flexible NetFlow will communicate to the NetFlow collector the configured section sizes in the corresponding version 9 export template fields. The payload sections will have a corresponding length field that can be used to collect the actual size of the collected section.

Applications and capabilities of user-defined records

User-defined records can be created for applications like QoS and bandwidth monitoring, application and end-user traffic profiling, and security monitoring for DDoS attacks. Flexible NetFlow user-defined records enable monitoring a contiguous section of a packet with a user-configurable size. This section can be used as a key or nonkey field along with other packet attributes. This section may include any Layer 3 data. It allows monitoring of packet fields not covered by predefined keys and facilitates detailed traffic monitoring and security applications, such as URL monitoring.

Predefined packet sections and configuration commands

- **collect ipv4 section header size** *bytes* --Starts capturing the number of bytes specified by the *bytes* argument from the beginning of the IPv4 header of each packet.
- **collect ipv4 section payload size** *bytes* --Starts capturing bytes immediately after the IPv4 header from each packet. The number of bytes captured is specified by the *bytes* argument.
- **collect ipv6 section header size** *bytes* --Starts capturing the number of bytes specified by the *bytes* argument from the beginning of the IPv6 header of each packet.

Handling of packet section sizes and export format

The *bytes* values are the sizes in bytes of these fields in the flow record. If the corresponding fragment of the packet is smaller than the requested section size, Flexible NetFlow will fill the rest of the section field in the flow record with zeros. If the packet type does not match the requested section type, Flexible NetFlow will fill the entire section field in the flow record with zeros.

Flexible NetFlow match parameters

This topic explains Flexible NetFlow match parameters and describes the commands and options used to configure flow records for network traffic analysis.

You must configure at least one of these match parameters for the flow records.

Table 2: Match parameters

Command	Purpose
match datalink { dot1q | ethertype | mac | vlan }	Specifies a match to datalink or Layer 2 fields. The following command options are available: • **dot1q** –Matches to the dot1q field. • **ethertype** –Matches to the ethertype of the packet. • **mac** –Matches the source or destination MAC fields. • **vlan** –Matches to the VLAN that the packet is located on (input or output).
match flow direction	Specifies a match to the flow identifying fields.

Command	Purpose
match interface { input | output }	Specifies a match to the interface fields. The following command options are available: • **input** –Matches to the input interface. • **output** –Matches to the output interface.
match ipv4 { destination | protocol | source | tos | ttl | version }	Specifies a match to the IPv4 fields. The following command options are available: • **destination** –Matches to the IPv4 destination address-based fields. • **protocol** –Matches to the IPv4 protocols. • **source** –Matches to the IPv4 source address based fields. • **tos** –Matches to the IPv4 Type of Service fields. • **ttl** –Matches to the IPv4 Time To Live fields. • **version** –Matches to the IP version from the IPv4 header.
match ipv6 { destination | hop-limit | protocol | source | traffic-class | version }	Specifies a match to the IPv6 fields. The following command options are available: • **destination** –Matches to the IPv6 destination address-based fields. • **hop-limit** –Matches to the IPv6 hop limit fields. • **protocol** –Matches to the IPv6 payload protocol fields. • **source** –Matches to the IPv6 source address based fields. • **traffic-class** –Matches to the IPv6 traffic class. • **version** –Matches to the IP version from the IPv6 header.
match transport { destination-port | igmp | icmp | source-port }	Specifies a match to the Transport Layer fields. The following command options are available: • **destination-port** –Matches to the transport destination port. • **icmp** –Matches to ICMP fields, including ICMP IPv4 and IPv6 fields. • **igmp** –Matches to IGMP fields. • **source-port** –Matches to the transport source port.

Flexible NetFlow collect parameters

This topic explains the Flexible NetFlow collect parameters and their purposes for Cisco Catalyst switches.

The Flexible NetFlow collect parameters are used to specify which fields and values are collected for flow monitoring and analysis.

Table 3: Collect parameters

Command	Purpose
collect counter { bytes { layer2 { long } | long } | packets { long } }	Collects the counter fields total bytes and total packets.
collect interface { input | output }	Collects the fields from the input or output interface.
collect timestamp absolute { first | last }	Collects the fields for the absolute time the first packet was seen or the absolute time the most recent packet was last seen (in milliseconds).
collect transport	▪ **packets** –packet fields ▪ **rcp** –RCP fields ▪ **tcp** –TCP fields

Command	Purpose
collect transport tcp flags	Collects the following transport TCP flags: • **ack** –TCP acknowledgement flag • **cwr** –TCP congestion window reduced flag • **ece** –TCP ECN echo flag • **fin** –TCP finish flag • **psh** –TCP push flag • **rst** –TCP reset flag • **syn** –TCP synchronize flag • **urg** –TCP urgent flag **Note** On the device, you cannot specify which TCP flag to collect. You can only specify to collect transport TCP flags. All TCP flags will be collected with this command.
collect flow { sampler | username }	Collects the following flow values: • **sampler** –Configures a flow sampler ID as a non-key field for the record. This field contains the ID of the flow sampler used to monitor the flow. • **username** –Configures the username associated with the flow.
collect ipv4 { source | destination }	Collects the following IPv4 fields: • **source** –Configures the IPv4 source as a non-key field for a flow record. • **destination** –Configures the IPv4 destination as a non-key field for a flow record.
collect ipv6 { source | destination }	Collects the following IPv6 information: • **source** –Configures the IPv6 source as a non-key field for a flow record. • **destination** –Configures the IPv6 destination as a non-key field for a flow record.

Command	Purpose
collect routing { next-hop address | destination | source | forwarding-status | multicast }	Collects the following routing values: • **next-hop address** –Information regarding the next hop from the flows. • **destination** –Destination routing attributes. • **source** –Source routing attributes. • **forwarding status** –Forwarding status of the packet. • **multicast** –Multicast routing attributes.
collect policy firewall event	Configures the collect policy firewall event as a non-key field and enables collecting information on SGACL denied or permitted traffic based on the traffic pattern.

Flow exporters

This topic explains how flow exporters are used to export flow data from the flow monitor cache to remote systems for analysis and storage, and describes the NetFlow export formats Version 9 and IPFIX (Version 10).

Flow exporters are entities that export flow data from the flow monitor cache to remote systems for analysis and storage.

- Flow exporters are created as separate entities in the configuration.
- Flow exporters are assigned to flow monitors to provide data export capability.
- Multiple flow exporters can be created and assigned to one or more flow monitors, or a single exporter can be applied to several monitors.

NetFlow data export formats: IPFIX (version 10) and version 9

NetFlow exporters support multiple export formats, such as IPFIX (Version 10) and NetFlow Version 9. Each format offers distinct features and benefits you can use.

- **IPFIX (version 10):** Internet Protocol Flow Information Export (IPFIX) is an IETF standard based on NetFlow version 9. The default destination port is 4739, DSCP value is 0, and TTL is 255.

  IPFIX supports:

  - predefined and user-defined flow records.
  - UDP and TCP transport protocols and is backward-compatible with NetFlow v9.
  - predefined information elements, including BGP community and AS path attributes for enhanced flow analysis.

- **NetFlow version 9:** The basic output is a flow record using a template-based export format, allowing extensibility and future enhancements without changing the basic flow-record format.

Key benefits of the NetFlow Version 9 template-based export format include:

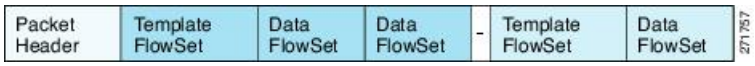
- Third-party applications do not need recompilation for new NetFlow features because you can use external data files that document template formats.
- You can add new features quickly without breaking current implementations.
- You can adapt NetFlow to new or developing protocols using the flexible template-based format.

NetFlow Version 9 export format provides the following features and functionality:

- Variable field specification format
- Support for IPv4 or IPv6 destination address export
- More efficient network utilization

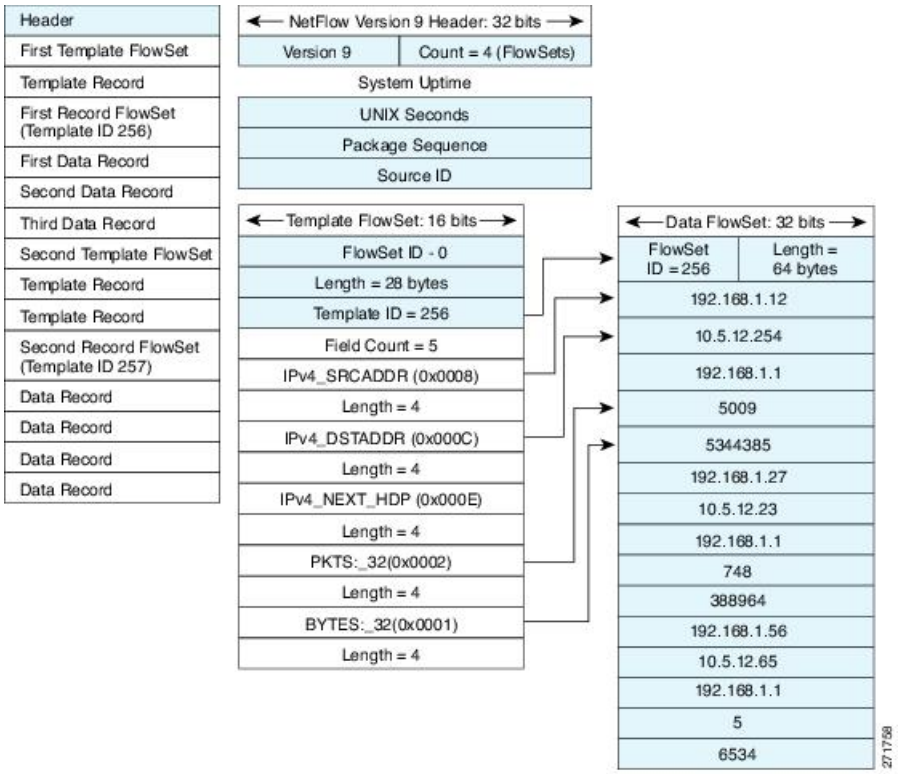
The version 9 export format consists of a packet header followed by one or more template flow or data flow sets. Template flow sets describe the fields present in future data flow sets, which may occur in the same or subsequent export packets. Template and data flow sets can be intermingled within a single export packet.

Figure 4: Version 9 export packet



NetFlow Version 9 periodically exports template data so that your collector understands what data to expect. It also exports the data flow set for the template. With Flexible NetFlow, you can configure a flow record. The system converts it to a Version 9 template and forwards it to the collector.

Figure 5: Detailed Example of the NetFlow version 9 export format



For more information on the version 9 export format, refer to the white paper titled Cisco IOS NetFlow Version 9 Flow-Record Format.

Flow monitors

This topic explains how flow monitors are used as Flexible NetFlow components applied to interfaces for network traffic monitoring, including their structure and usage scenarios.

Flow monitors are a category of Flexible NetFlow components that are applied to interfaces to perform network traffic monitoring.

- Consist of a user-defined record, an optional flow exporter, and a cache created when applied to the first interface.
- Collect flow data from network traffic and add it to the flow monitor cache based on key and nonkey fields in the flow record.
- Enable different types of analysis on the same traffic by applying multiple flow monitors with custom records.

Flow monitor operation

Flow monitors are applied to device interfaces to monitor and analyze network traffic using Flexible NetFlow.

- Flow monitors consist of a user-defined record, an optional flow exporter, and a cache.
- Flow data is collected and stored in the cache during monitoring based on the flow record fields.
- Multiple flow monitors can be used for different types of analysis on the same traffic.

The following describes the default cache type and its behavior:

- The default cache type is **normal**. Entries are aged out according to timeout active and timeout inactive settings. When aged out, entries are removed and exported via configured exporters.

Usage of flow monitors

Flexible NetFlow can analyze the same traffic using different records. For example, packet 1 can be analyzed with a record for standard traffic analysis on the input interface and a record for security analysis on the output interface. Another example shows applying different types of flow monitors with custom records for more complex monitoring scenarios.

Figure 6: Two flow monitors to analyze the same traffic

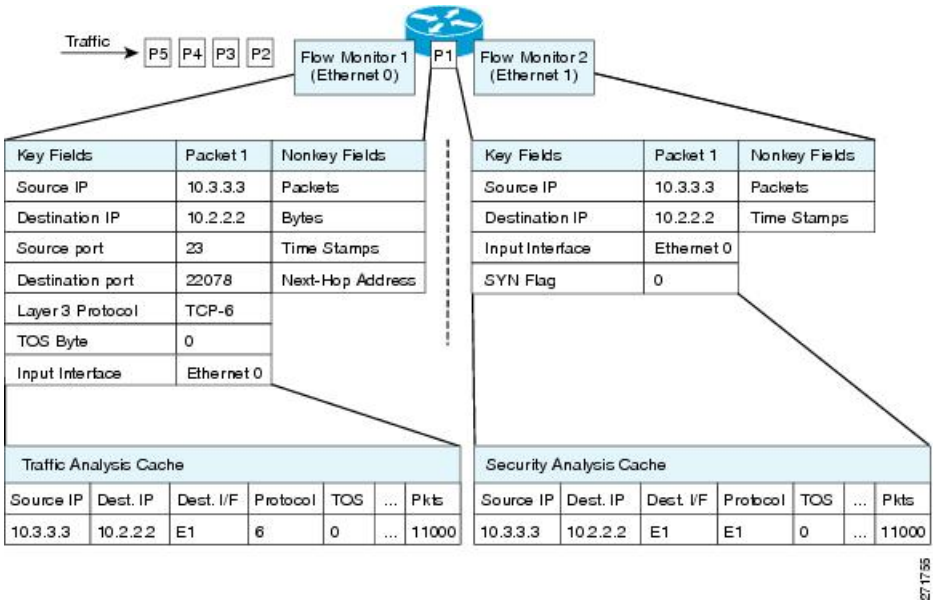
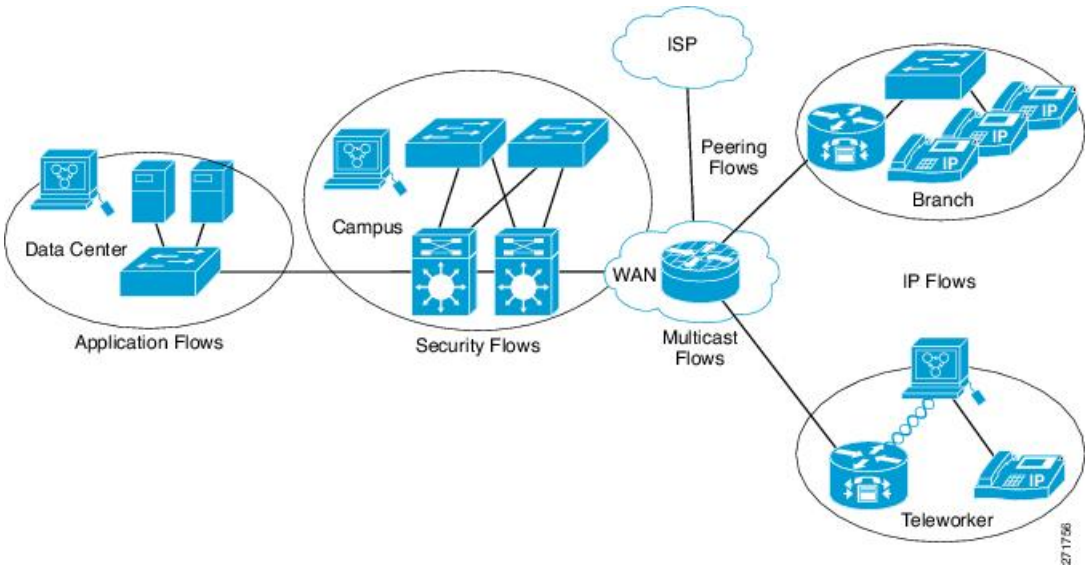


Figure 7: Multiple types of flow monitors with custom records



Flow samplers

This topic explains flow samplers, which are used to reduce the load on devices running Flexible NetFlow by limiting the number of packets selected for analysis.

Flow samplers are components in a router's configuration that reduce device load by limiting the number of packets selected for analysis in Flexible NetFlow.

- Flow samplers are created as separate components in a router's configuration.
- They reduce the load on the device by limiting the number of packets selected for analysis.
- Samplers are combined with flow monitors when applied to an interface with the **ip flow monitor** command.

Flexible NetFlow fields

This topic explains the supported Flexible NetFlow (FNF) fields for various traffic types and directions, including key and collect fields, and notes on VLAN field length accounting.

These tables provide a consolidated list of supported fields in Flexible NetFlow (FNF) for various traffic types and traffic direction.

Note

If the packet has a VLAN field, then that length is not accounted for.

Field	Layer 2 In	Layer 2 Out	IPv4 In	IP v4 Out	IPv6 In	IPv6 Out	Notes
Interface input	Yes	–	Yes	–	Yes	–	If you apply a flow monitor in the input direction: • Use the **match** keyword and use the input interface as a key field. • Use the **collect** keyword and use the output interface as a collect field. This field will be present in the exported records but with a value of 0.
Interface output	–	Yes	–	Yes	–	Yes	If you apply a flow monitor in the output direction: • Use the **match** keyword and use the output interface as a key field. • Use the **collect** keyword and use the input interface as a collect field. This field will be present in the exported records but with a value of 0.

Field	Layer 2 In	Layer 2 Out	IPv4 In	IP v4 Out	IPv6 In	IPv6 Out	Notes
Flow direction	Yes	Yes	Yes	Yes	Yes	Yes	–
Ethertype	Yes	Yes	–	–	–	–	–
VLAN input	Yes	–	Yes	–	Yes	–	Supported only for a switch port.
VLAN output	–	Yes	–	Yes	–	Yes	Supported only for a switch port.
dot1q VLAN input	Yes	–	Yes	–	Yes	–	Supported only for a switch port.

Field	Layer 2 In	Layer 2 Out	IPv4 In	IP v4 Out	IPv6 In	IPv6 Out	Notes
dot1q VLAN output	–	Yes	–	Yes	–	Yes	Supported only for a switch port.
dot1q priority	Yes	Yes	Yes	Yes	Yes	Yes	Supported only for a switch port.
MAC source address input	Yes	Yes	Yes	Yes	Yes	Yes	
MAC source address output	–	–	–	–	–	–	–
MAC destination address input	Yes	–	Yes	–	Yes	–	–
MAC destination address output	–	Yes	–	Yes	–	Yes	–
IPv4 version	–	–	Yes	Yes	Yes	Yes	–
IPv4 TOS	–	–	Yes	Yes	Yes	Yes	–
IPv4 protocol	–	–	Yes	Yes	Yes	Yes	Must use if any of src/dest port, ICMP code/type, IGMP type or TCP flags are used.
IPv4 TTL	–	–	Yes	Yes	Yes	Yes	
IPv4 TTL	–	–	Yes	Yes	Yes	Yes	Same as IPv4 TTL.
IPv4 protocol	–	–	Yes	Yes	Yes	Yes	Same as IPv4 protocol. Must use if any of src/dest port, ICMP code/type, IGMP type or TCP flags are used.
IPv4 source address	–	–	Yes	Yes	–	–	–

Field	Layer 2 In	Layer 2 Out	IPv4 In	IP v4 Out	IPv6 In	IPv6 Out	Notes
IPv4 destination address	–	–	Yes	Yes	–	–	–
ICMP IPv4 type	–	–	Yes	Yes	–	–	–
ICMP IPv4 code	–	–	Yes	Yes	–	–	–
IGMP type	–	–	Yes	Yes	–	–	–

Field	Layer 2 In	Layer 2 Out	IPv4 In	IP v4 Out	IPv6 In	IPv6 Out	Notes
IPv6 version	–	–	Yes	Yes	Yes	Yes	Same as IP version.
IPv6 protocol	–	–	Yes	Yes	Yes	Yes	Same as IP protocol. Must use if any of src/dest port, ICMP code/type, IGMP type or TCP flags are used.
IPv6 source address	–	–	–	–	Yes	Yes	–
IPv6 destination address	–	–	–	–	Yes	Yes	–
IPv6 traffic-class	–	–	Yes	Yes	Yes	Yes	Same as IP TOS.
IPv6 hop-limit	–	–	Yes	Yes	Yes	Yes	Same as IP TTL.
ICMP IPv6 type	–	–	–	–	Yes	Yes	–
ICMP IPv6 code	–	–	–	–	Yes	Yes	–
source-port	–	–	Yes	Yes	Yes	Yes	–
dest-port	–	–	Yes	Yes	Yes	Yes	–

Field	Layer 2 In	Layer 2 Out	IPv4 In	IP v4 Out	IPv6 In	IPv6 Out	Notes
Bytes long	Yes	Yes	Yes	Yes	Yes	Yes	Packet size = (Ethernet frame size including FCS - 18 bytes) **Recommended:** Avoid this field and use Bytes layer2 long.
Packets long	Yes	Yes	Yes	Yes	Yes	Yes	–
Timestamp absolute first	Yes	Yes	Yes	Yes	Yes	Yes	–
Timestamp absolute last	Yes	Yes	Yes	Yes	Yes	Yes	–
TCP flags	Yes	Yes	Yes	Yes	Yes	Yes	Collects all flags.
Bytes layer2 long	Yes	Yes	Yes	Yes	Yes	Yes	–

Default settings

This topic explains the Flexible NetFlow default settings for the device.

This table lists the Flexible NetFlow default settings for the device.

Table 4: Default Flexible NetFlow settings

Setting	Default
Flow active timeout	1800 seconds
Flow timeout inactive	15 seconds

Flexible NetFlow for Ingress VRF

This topic explains the Flexible NetFlow–Ingress VRF Support feature, which enables collecting the VRF ID from incoming packets on a device using an input flow monitor with a flow record that collects the VRF ID as a key field.

The Flexible NetFlow for Ingress VRF is a method that enables the collection of the VRF ID from incoming packets on a device.

- Collects the VRF ID from incoming packets.

- Uses an input flow monitor with a flow record.
- Collects the VRF ID as a key field.

Flexible Netflow for Egress VRF

This topic explains how the Flexible Netflow–Egress VRF Support feature enables collecting the VRF ID from outgoing packets on a device using an output flow monitor.

Flexible Netflow for Egress VRF is a mechanism that enables collecting the VRF ID from outgoing packets on a device by applying an output flow monitor with a flow record that collects the VRF ID as a key field.

- Collects the VRF ID from outgoing packets.
- Uses an output flow monitor with a flow record.
- VRF ID is collected as a key field.

Autonomous system number

This topic explains the concept of Autonomous System Numbers (AS numbers), their structure, assignment, and role in supporting the Internet's public inter-domain routing system, especially in the context of NetFlow and BGP.

An Autonomous System Number (AS number) is a unique identifier assigned by IANA to a network under a single technical administration, used primarily with Border Gateway Protocol (BGP) for inter-domain routing on the Internet.

- AS numbers are 32-bit values, providing 4,294,967,296 unique identifiers.
- They are required for running BGP and peering with Internet Service Providers (ISPs) and at Internet Exchanges (IX).
- Each AS number must be globally unique to ensure proper routing of IP address blocks.

Autonomous System number reference information

NetFlow version 9 and IPFIX export types support 32-bit AS numbers. The NetFlow version 5 export protocol does not support the 32-bit AS field because it uses a fixed 16-bit source and destination AS format. However, on some platforms, NetFlow version 5 export protocol may support the 32-bit AS field, with the exported field containing the lower 32 bits of a 64-bit counter value.

You can export the following BGP parameters:

- BGP source origin or peer AS number
- BGP destination origin or peer AS number

Use the **[no] collect routing** { **`destination | source`** } as **[[4-octet] peer] [4-octet]** command to configure the AS number system.

Configure a flow record

Learn how to configure a customized flow record to analyze traffic data using Flexible NetFlow. This procedure guides you through creating and verifying a flow record for your requirements.

Perform this task to configure a customized flow record.

Customized flow records are used to analyze traffic data for a specific purpose. A customized flow record must have at least one **match** criterion for use as the key field and typically has at least one **collect** criterion for use as a nonkey field.

There are hundreds of possible permutations of customized flow records. This task shows the steps that are used to create one of the possible permutations. Modify the steps in this task as appropriate to create a customized flow record for your requirements.

Procedure

1. Use the **enable** command to enter privileged EXEC mode.

   Example

   ```
   Switch> enable
   ```

   Enter your password if prompted.

2. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Switch# configure terminal
   ```

3. Use the **flow record** *record-name* command to create a flow record.

   Example

   ```
   Switch(config)# flow record FLOW-RECORD-1
   ```

   This command also allows you to modify an existing flow record.

4. (Optional) Use the **description** *description* command to create a description for the flow record.

   Example

   ```
   Switch(config-flow-record)# description Used for basic traffic analysis
   ```

5. Use the **match** { **ipv4** | **ipv6** }{ **destination** | **source** } **address** command to configure the IPv4 destination address as a key field for the record.

   Example

   ```
   Switch(config-flow-record)# match ipv4 destination address
   ```

6. Repeat Step 5 as required to configure additional key fields for the record.
7. Use the **end** command to exit flow monitor configuration mode and return to privileged EXEC mode.

   Example

   ```
   Switch(config-flow-record)# end
   ```

8. (Optional) Use the **show flow record** *record-name* command to display the current status of the specified flow record.

Example

```
Switch# show flow record FLOW_RECORD-1
```

Create a flow exporter

Learn how to create a flow exporter to define export parameters for a flow, including destination, protocol, and interface settings.

You can create a flow exporter to define the export parameters for a flow.

> **Note**
>
> Each flow exporter supports only one destination. If you want to export the data to multiple destinations, you must configure multiple flow exporters and assign them to the flow monitor.
>
> You can export to a destination using IPv4 address.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Switch# configure terminal
   ```

2. Use the **flow exporter** *name* command to create a flow exporter.

   Example

   ```
   Switch(config)# flow exporter ExportTest
   ```

   You can modify an existing flow exporter using **flow exporter** command.

3. (Optional) Use the **description** *string* command to configure a description for the flow exporter.

   Example

   ```
   Switch(config-flow-exporter)# description ExportV9
   ```

   The description is a maximum 63-character string.

4. Use the **destination** { *ipv4-address* } command to set the IPv4 destination address or hostname for this exporter.

   Example

   ```
   Switch(config-flow-exporter)# destination 192.0.2.1
   ```

5. (Optional) Use the **dscp** *value* command to specify the differentiated services codepoint value.

   Example

   ```
   Switch(config-flow-exporter)# dscp 0
   ```

   The *dscp* value ranges from 0 to 63. The default value is 0.

6. Use the **source** *interface id* command to specify the interface to reach the NetFlow collector at the configured destination.

   Example

   ```
   Switch(config-flow-exporter)# source gigabitEthernet1/0/1
   ```

   Note

   Flow Exporter does not support unnumbered IP interface as source interface.

   These interfaces can be configured as source:

   Table 5: Interface type keywords

Keyword	Description
Auto Template	Auto-Template interface
Capwap	CAPWAP tunnel interface
GigabitEthernet	Gigabit Ethernet IEEE 802
GroupVI	Group virtual interface
Internal Interface	Internal interface
Loopback	Loopback interface
Null	Null interface
Port-channel	Ethernet Channel of interface
TenGigabitEthernet	10-Gigabit Ethernet
Tunnel	Tunnel interface
Vlan	VLANs

7. (Optional) Use the **transport udp** *number* command to specify the UDP port to use to reach the NetFlow collector.

   Example

   ```
   Switch(config-flow-exporter)# transport udp 200
   ```

   The UDP value ranges from 0 to 65535. For IPFIX exporting protocol, the default destination port is 4739.

8. (Optional) Use the **ttl** *seconds* command to configure the time-to-live (TTL) value for datagrams sent by the exporter.

Example

```
Switch(config-flow-exporter)# ttl 210
```

The TTL value ranges from 1 to 255 seconds. The default value is 255.

9. Use the **export-protocol { netflow-v9 }** command to specify the version of the NetFlow export protocol used by the exporter.

Example

```
Switch(config-flow-exporter)# export-protocol netflow-v9
```

Default: **netflow-v9**

Create a customized flow monitor

Learn how to create a customized flow monitor using Flexible NetFlow, including prerequisites, configuration steps, and verification commands.

Create a customized flow monitor to analyze network traffic using Flexible NetFlow. This task enables you to define and configure a flow monitor with user-defined or predefined records.

Use this procedure to create a customized flow monitor. Each flow monitor has a separate cache assigned to it and requires a record to define the contents and layout of its cache entries.

Record formats can be one of the predefined formats or a user-defined format. Advanced users can create a customized format using the **flow record** command.

Before you begin

If you want to use a customized record instead of using one of the Flexible NetFlow predefined records, you must create the customized record before you can perform this task. If you want to add a flow exporter to the flow monitor for data export, you must create the exporter before you can complete this task.

Note

You must use the **no ip flow monitor** command to remove a flow monitor from all of the interfaces to which you have applied it before you can modify the parameters for the **record** command on the flow monitor.

Procedure

1. Use the **enable** command to enter privileged EXEC mode.

Example

```
Switch> enable
```

Enter your password if prompted.

2. Use the **configure terminal** command to enter global configuration mode.

Example

```
Switch# configure terminal
```

3. Use the **flow monitor** *monitor-name* command to create the flow monitor.

   Example

   ```
   Switch(config)# flow exporter EXPORTER-1
   ```

   You can modify an existing flow monitor using **flow monitor** command.

4. (Optional) Use the **description** *description* command to configure a description to the monitor.

   Example

   ```
   Switch(config-flow-monitor)# description Used for basic IPv4 traffic analysis
   ```

5. Use the **record** { *record-name* | **netflow-original** | **netflow** { **ipv4** | **ipv6** } *record* [**peer**] } command to specify the record for the flow monitor.

   Example

   ```
   Switch(config-flow-monitor)# record FLOW-RECORD-1
   ```

6. (Optional) Use the **cache** { **timeout** { **active** | **inactive** | **update** } *seconds* | **type** **normal** } command to modify the flow monitor cache parameters such as timeout values, and the cache type. Associates a flow cache with the specified flow monitor.

   Example

   ```
   Switch(config-flow-monitor)# cache type normal
   ```

   ```
   Switch(config-flow-monitor)# cache timeout active 1800
   ```

   Example

   > **Note**
   >
   > Repeat this step for each cache parameter that you want to configure.

7. Use the **statistics packet protocol** command to monitor the collection of protocol distribution statistics for Flexible NetFlow.

   Example

   ```
   Switch(config-flow-monitor)# statistics packet protocol
   ```

   (Optional) Use the **statistics packet size** command to monitor the collection of size distribution statistics for Flexible NetFlow.

Example

```
Switch(config-flow-monitor)# statistics packet size
```

8. (Optional) Use the **exporter** *exporter-name* command to specify the name of an exporter that was created previously.

Example

```
Switch(config-flow-monitor)# exporter EXPORTER-1
```

9. Use the **end** command to exit flow-monitor configuration mode and return to privileged EXEC mode.

Example

```
Switch(config-flow-monitor)# end
```

> **Note**
>
> Verify the flow monitor configuration by using the **show flow monitor** [[**name**] *monitor-name* [**cache** [**format** { **csv** | **record** | **table** }]] [**statistics**]] command to display the status and statistics for the flow monitor.

Example

```
Switch# show flow monitor FLOW-MONITOR-2 cache
```

Create a flow sampler

Learn how to configure and enable a flow sampler to monitor network traffic using sampling methods.

Perform this required task to configure and enable a flow sampler.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

Example

```
Switch# configure terminal
```

2. Use the **sampler** *sampler-name* command to create a sampler.

Example

```
Switch(config)# sampler SAMPLER-1
```

You can modify an existing sampler using **sampler** command.

3. (Optional) Use the **description** *description* command to configure a description to the monitor.

   Example

   ```
   Switch(config-sampler)# description Sample at 50 percent
   ```

4. Use the **mode { random } 1 out-of** *window-size* command to specify the sampler mode and the flow sampler window size.

   Example

   ```
   Switch(config-sampler)# mode random 1 out-of 2
   ```

   The range for `window-size` is 2 to 1024.

5. Use the **exit** command to exit sampler configuration mode and return to global configuration mode.

   Example

   ```
   Switch(config-sampler)# exit
   ```

   Exits sampler configuration mode and return to global configuration mode.

6. Use the **interface** *type number* command to specify an interface and enter interface configuration mode.

   Example

   ```
   Switch(config)# interface GigabitEthernet 0/0/0
   ```

7. Use the **{ ip | ipv6 } flow monitor** *monitor-name* **[[sampler]** *sampler-name* **]{ input | output }** command to assign the flow monitor and the flow sampler that you created to the interface to enable sampling.

   Example

   ```
   Switch(config-if)# ip flow monitor FLOW-MONITOR-1 sampler SAMPLER-1 input
   ```

8. Use the **end** command to exit flow exporter configuration mode and return to privileged EXEC mode.

   Example

   ```
   Switch(config-if)# end
   ```

9. (Optional) Use the **show sampler** *sampler-name* command to display the status and statistics of the flow sampler that you configured and enabled.

   Example

   ```
   Switch# show sampler SAMPLER-1
   ```

Apply a flow to an interface

Learn how to apply a flow monitor and an optional sampler to an interface to enable Flexible NetFlow monitoring for IPv4, IPv6, and datalink traffic.

You can apply a flow monitor and an optional sampler to an interface.

Procedure

1. Use the **enable** command to enter privileged EXEC mode.

   Example

   ```
   Switch> enable
   ```

   Enter your password if prompted.

2. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Switch# configure terminal
   ```

3. Use the **interface** *type* command to configure an interface and enter interface configuration mode.

   Example

   ```
   Switch(config)# interface GigabitEthernet1/0/1
   ```

   You can use Flexible NetFlow with L2 port-channel member ports, but not with the L2 port-channel interface.

   You can use Flexible NetFlow on either the L3 port-channel interface or its member ports, but not at the same time.

4. Use the { **ip flow monitor** | **ipv6 flow monitor** | **datalink flow monitor** } *name* [**sampler** *name*] { **input output** } command to associate monitors to an interface in both input and output directions.

   Example

   ```
   Switch(config-if)# ip flow monitor MonitorTest input
   ```

   Associates an IPv4, IPv6 and datalink flow monitor, and an optional sampler to the interface for input or output packets.

   ip flow monitor - Enables Flexible NetFlow to monitor IPv4 traffic.

   ipv6 flow monitor - Enables Flexible NetFlow to monitor IPv6 traffic.

   datalink flow monitor - Enables Flexible NetFlow to monitor non-IP traffic.

   > **Note**

5. Use the **end** command to exit interface configuration mode and return to privileged EXEC mode.

Example

```
Switch(config-if)# end
```

6. (Optional) Use the **show flow interface** command to display the NetFlow on an interface information.

Example

```
Switch# show flow interface
```

Configure a bridged NetFlow on a VLAN

Learn how to configure a bridged NetFlow on a VLAN by applying a flow monitor and an optional sampler to a VLAN interface.

You can apply a flow monitor and an optional sampler to a VLAN.

Procedure

1. Use the **enable** command to enter privileged EXEC mode.

Example

```
Switch> enable
```

Enter your password if prompted.

2. Use the **configure terminal** command to enter global configuration mode.

Example

```
Switch# configure terminal
```

3. Use the **vlan** [**configuration**] *vlan-id* command to enter VLAN or VLAN configuration mode.

Example

```
Switch(config)#  vlan configuration 30
```

Enters VLAN or VLAN configuration mode.

4. Use the **ip flow monitor** *monitor name* [**sampler** *sampler name*] { **input** | **output** } command to associate a flow monitor and optional sampler with the VLAN for input traffic.

Example

```
Switch(config-vlan-config)# ip flow monitor MonitorTest input
```

5. Use the **end** command to exit flow-record configuration mode.

Example

```
Switch(config-flow-record)# end
```

Configure Layer 2 NetFlow

Learn how to configure Layer 2 NetFlow by defining Layer 2 keys in Flexible NetFlow records to capture flows on Layer 2 interfaces.

You can define Layer 2 keys in Flexible NetFlow records that you can use to capture flows in Layer 2 interfaces.

Procedure

1. Use the **enable** command to enter privileged EXEC mode.

   Example

   ```
   Switch> enable
   ```

   Enter your password if prompted.

2. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Switch# configure terminal
   ```

3. Use the **flow record** *name* command to enter flow record configuration mode.

   Example

   ```
   Switch(config)# flow record L2_record
   ```

   Enters flow record configuration mode.

4. Use the **match datalink** { **dot1q** | **ethertype** | **mac** | **vlan** } command to specify the layer 2 attribute as a key.

   Example

   ```
   Switch(config-flow-record)# match datalink ethertype
   ```

5. Use the **end** command to exit flow exporter configuration mode and return to privileged EXEC mode.

   Example

   ```
   Switch(config-flow-monitor)# end
   ```

6. (Optional) Use the **show flow record** command to monitor NetFlow on an interface.

Example

```
Switch# show flow record
```

Monitor Flexible NetFlow

This topic explains how to monitor Flexible NetFlow using various show commands to display information and statistics about flow exporters, monitors, interfaces, records, and samplers.

You can display information and statistics about flow exporters, monitors, interfaces, records, and samplers using these show commands.

Table 6: Flexible NetFlow monitor commands

The commands in the following table can be used to monitor Flexible NetFlow.

Command	Purpose
show flow exporter [**broker** | **export-ids** | **name** | *name* | **statistics** | **templates**]	Displays information about NetFlow flow exporters and statistics.
show flow interface	Displays information about NetFlow interfaces.
show flow monitor [**name** *monitor-name*]	Displays information about NetFlow flow monitors and statistics.
show flow monitor statistics [**name** *monitor-name*] **statistics**	Displays the statistics for the flow monitor.
show flow monitor [**name** *monitor-name*] **cache format** {**table** | **record** | **csv** }	Displays the contents of the cache for the flow monitor, in the format specified.
show flow record [**name** *record-name*]	Displays information about NetFlow flow records.
show sampler [**broker** | **name** | *name*]	Displays information about NetFlow samplers.

Flow configuration commands

Describes the configuration of a flow and its application to an interface using CLI commands. This process enables network traffic monitoring and data export for analysis.

This reference provides the CLI command sequence required to create a flow and apply it to a specific interface.

```
Switch# configure terminal
Switch(config)# flow export export1
Switch(config-flow-exporter)# destination 10.0.101.254
Switch(config-flow-exporter)# transport udp 2055
Switch(config-flow-exporter)# exit
Switch(config)# flow record record1
Switch(config-flow-record)# match ipv4 source address
Switch(config-flow-record)# match ipv4 destination address
Switch(config-flow-record)# match ipv4 protocol
```

```
Switch(config-flow-record)# match transport source-port
Switch(config-flow-record)# match transport destination-port
Switch(config-flow-record)# match flow cts source group-tag
Switch(config-flow-record)# match flow cts destination group-tag
Switch(config-flow-record)# collect counter byte long
Switch(config-flow-record)# collect counter packet long
Switch(config-flow-record)# collect timestamp absolute first
Switch(config-flow-record)# collect timestamp absolute last
Switch(config-flow-record)# exit
Switch(config)# flow monitor monitor1
Switch(config-flow-monitor)# record record1
Switch(config-flow-monitor)# exporter export1
Switch(config-flow-monitor)# exit
Switch(config)# interface tenGigabitEthernet 1/0/1
Switch(config-if)# ip flow monitor monitor1 input
Switch(config-if)# end
```

Flexible NetFlow flow monitor configuration example

Describes the configuration of flow records, exporters, and monitors to track IPv4 ingress traffic between network interfaces. This setup enables network traffic analysis using NetFlow.

This reference provides the configuration steps to monitor IPv4 ingress traffic flow between network interfaces using NetFlow.

Configuration example

This example shows how to monitor IPv4 ingress traffic where interface g1/0/11 sends traffic to interfaces g1/0/36 and g3/0/11.

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# flow record fr-1
Switch(config-flow-record)# match ipv4 source address
Switch(config-flow-record)# match ipv4 destination address
Switch(config-flow-record)# match interface input
Switch(config-flow-record)# collect counter bytes long
Switch(config-flow-record)# collect counter packets long
Switch(config-flow-record)# collect timestamp absolute first
Switch(config-flow-record)# collect timestamp absolute last
Switch(config-flow-record)# collect counter bytes layer2 long
Switch(config-flow-record)# exit

Switch(config)# flow exporter fe-ipfix6
Switch(config-flow-exporter)# destination 2001:0:0:24::10
Switch(config-flow-exporter)# source Vlan106
Switch(config-flow-exporter)# transport udp 4739
Switch(config-flow-exporter)# export-protocol ipfix
Switch(config-flow-exporter)# template data timeout 240
Switch(config-flow-exporter)# exit
Switch(config)# flow exporter fe-ipfix
Switch(config-flow-exporter)# description IPFIX format collector 100.0.0.80
Switch(config-flow-exporter)# destination 100.0.0.80
Switch(config-flow-exporter)# dscp 30
Switch(config-flow-exporter)# ttl 210
Switch(config-flow-exporter)# transport udp 4739
Switch(config-flow-exporter)# export-protocol ipfix
Switch(config-flow-exporter)# template data timeout 240
Switch(config-flow-exporter)# exit
```

```
Switch(config)# flow exporter fe-1
Switch(config-flow-exporter)# destination 10.5.120.16
Switch(config-flow-exporter)# source Vlan105
Switch(config-flow-exporter)# dscp 32
Switch(config-flow-exporter)# ttl 200
Switch(config-flow-exporter)# transport udp 2055
Switch(config-flow-exporter)# template data timeout 240
Switch(config-flow-exporter)# exit

Switch(config)# flow monitor fm-1
Switch(config-flow-monitor)# exporter fe-ipfix6
Switch(config-flow-monitor)# exporter fe-ipfix
Switch(config-flow-monitor)# exporter fe-1
Switch(config-flow-monitor)# cache timeout inactive 60
Switch(config-flow-monitor)# cache timeout active 180
Switch(config-flow-monitor)# record fr-1
Switch(config-flow-monitor)# end

Switch# show running-config interface g1/0/11
Switch# show running-config interface g1/0/36
Switch# show running-config interface g3/0/11
Switch# show flow monitor fm-1 cache format table
```

NetFlow monitoring parameters for IPv4 egress traffic

Describes the configuration of flow records, exporters, and monitors to track IPv4 egress traffic using NetFlow. This setup enables the collection and export of traffic statistics to network collectors for analysis.

This reference provides the configuration parameters for NetFlow monitoring to track IPv4 egress traffic by setting up flow records, exporters, and monitors that collect and export traffic data to network collectors.

Flow record configuration

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# flow record  fr-1 out
Switch(config-flow-record)# match ipv4 source address
Switch(config-flow-record)# match ipv4 destination address
Switch(config-flow-record)# match interface output
Switch(config-flow-record)# collect counter bytes long
Switch(config-flow-record)# collect counter packets long
Switch(config-flow-record)# collect timestamp absolute first
Switch(config-flow-record)# collect timestamp absolute last
Switch(config-flow-record)# exit
```

Flow exporter configurations

```
Switch(config)# flow exporter fe-1
Switch(config-flow-exporter)# destination 10.5.120.16
Switch(config-flow-exporter)# source Vlan105
Switch(config-flow-exporter)# dscp 32
Switch(config-flow-exporter)# ttl 200
Switch(config-flow-exporter)# transport udp 2055
```

```
Switch(config-flow-exporter)# template data timeout 240
Switch(config-flow-exporter)# exit

Switch(config)# flow exporter fe-ipfix6
Switch(config-flow-exporter)# destination 2001:0:0:24::10
Switch(config-flow-exporter)# source Vlan106
Switch(config-flow-exporter)# transport udp 4739
Switch(config-flow-exporter)# export-protocol ipfix
Switch(config-flow-exporter)# template data timeout 240
Switch(config-flow-exporter)# exit

Switch(config)# flow exporter fe-ipfix
Switch(config-flow-exporter)# description IPFIX format collector 100.0.0.80
Switch(config-flow-exporter)# destination 100.0.0.80
Switch(config-flow-exporter)# dscp 30
Switch(config-flow-exporter)# ttl 210
Switch(config-flow-exporter)# transport udp 4739
Switch(config-flow-exporter)# export-protocol ipfix
Switch(config-flow-exporter)# template data timeout 240
Switch(config-flow-exporter)# exit
```

Flow monitor configuration

```
Switch(config)# flow monitor fm-1-output
Switch(config-flow-monitor)# exporter fe-1
Switch(config-flow-monitor)# exporter fe-ipfix6
Switch(config-flow-monitor)# exporter fe-ipfix
Switch(config-flow-monitor)# cache timeout inactive 50
Switch(config-flow-monitor)#  cache timeout active 120
Switch(config-flow-monitor)# record fr-1-out
Switch(config-flow-monitor)# end
```

Verification

```
Switch# show flow monitor fm-1-output cache format table
```

Flexible NetFlow for ingress VRF monitoring

Provides an example to show how to configure Flexible NetFlow for Ingress VRF Support.

This example configures the collection of the VRF ID from incoming packets on a switch by applying an input flow monitor having a flow record that collects the VRF ID as a key field.

```
Switch> enable
Switch# configure terminal
Switch(config)# flow record rm_1
Switch(config-flow-record)# match routing vrf input
Switch(config-flow-record)# match ipv4 source address
Switch(config-flow-record)# match ipv4 destination address
Switch(config-flow-record)# collect interface input
Switch(config-flow-record)# collect interface output
Switch(config-flow-record)# collect counter packets
Switch(config-flow-record)# exit
```

```
Switch(config)# flow monitor mm_1
Switch(config-flow-monitor)# record rm_1
Switch(config-flow-monitor)# exit

Switch(config)# interface GigabitEthernet 1/0/1
Switch(config-if)# ip vrf forwarding green
Switch(config-if)# ip address 172.16.2.2 255.255.255.252
Switch(config-if)# ip flow monitor mm_1 input
Switch(config-if)# end
```

Flexible NetFlow for egress VRF configuration

Describes the configuration of VRF ID collection from outgoing packets on a switch. This process enables network monitoring and analysis based on VRF routing context for egress traffic.

This reference provides the configuration steps and commands required to enable Flexible NetFlow for collecting VRF ID information from egress traffic on a switch interface.

```
Switch> enable
Switch# configure terminal
Switch(config)# flow record rm_1
Switch(config-flow-record)# match routing vrf intput
Switch(config-flow-record)# match ipv4 source address
Switch(config-flow-record)# match ipv4 destination address
Switch(config-flow-record)# collect interface input
Switch(config-flow-record)# collect interface output
Switch(config-flow-record)# collect counter packets
Switch(config-flow-record)# exit

Switch(config)# flow monitor mm_1
Switch(config-flow-monitor)# record rm_1
Switch(config-flow-monitor)# exit

Switch(config)# interface GigabitEthernet 1/0/1
Switch(config-if)# ip vrf forwarding green
Switch(config-if)# ip address 172.16.2.2 255.255.255.252
Switch(config-if)# ip flow monitor mm_1 output
Switch(config-if)# end
```

Feature history for Flexible NetFlow

Provides release and related information for Flexible NetFlow features explained in this module.

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE 16.5.1a	Flexible NetFlow	Flexible NetFlow uses flows to provide statistics for accounting, network monitoring, and network planning.

Release	Feature	Feature Information
Cisco IOS XE 17.1.1	Source Group Tag (SGT), Destination Group Tag (DGT) over Flexible NetFlow for IPv6 traffic	Allows capture of IP flow information for packets undergoing Multiprotocol Label Switching (MPLS) label imposition when entering an MPLS network. These packets arrive on a device as IP packets and are transmitted as MPLS packets.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/.