Configuring IP-aware Netflow for VRF Ingress

- Restrictions for IP-aware Netflow for VRF Ingress, on page 1
- Information About IP-aware Netflow for VRF Ingress, on page 1
- How to Configure IP-aware Netflow for VRF Ingress, on page 2
- Configuration Examples for IP-aware Netflow for VRF Ingress, on page 6
- Feature History for IP-aware Netflow for VRF Ingress, on page 8

Restrictions for IP-aware Netflow for VRF Ingress

- IP-aware VRF ingress Netflow is supported with IPv4, IPv6 and MVPNv4 as CE facing interface
- Supported only on layer 3 interface
- Supported only for ingress traffic on the VRF interface
- Supported only for MPLS L3 VPN VRF interface
- IP aware VRF ingress Netflow on MVPNv6 as CE facing interface is not supported
- Not supported on portchannel, SVI as CE facing interface
- Not supported for egress traffic on the VRF interface
- Not supported on MPLS L2VPN Attachment circuit interface

Information About IP-aware Netflow for VRF Ingress

This feature enables collecting the virtual routing and forwarding (VRF) ID from incoming packets on a router by applying an input flow monitor having a flow record that collects the VRF ID as a key or a non-key field.

Table 1: Scale Numbers

Platform	SDM Template	Max IPv4 Flows	Max IPv6 Flows
9300	Access	16K	8K
9400	Distribution	32K	16K

Platform	SDM Template	Max IPv4 Flows	Max IPv6 Flows
9500	Access	32K	16K
9600	Core	32K	32K

How to Configure IP-aware Netflow for VRF Ingress

This section provides the configuration steps for configuring IP-aware Netflow for VRF Ingress:

Creating a Flow Record

Perform the following task to create a flow record.

Step 1

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device(config)# configure terminal`	Enters global configuration mode.
Step 3	**flow record** *flow_record_name* **Example:** `Device(config)# flow record flow-record-1`	Enters flow record configuration mode.
Step 4	**description** *description* **Example:** `Device(config-flow-record)# description flow-record-1`	(Optional) Creates a description for the flow record.
Step 5	**match ipv4 version** **Example:** `Device (config-flow-record)# match ipv4 version`	Specifies a match to the IP version from the IPv4 header.
Step 6	**match ipv4** {**source** | **destination**} *address*	Specifies a match to the IPv4 source and destination address.

	Command or Action	Purpose
Step 7	**match ipv4 protocol** **Example:** `Device (config-flow-record)# match ipv4 protocol`	Specifies a match to the IPv4 protocol.
Step 8	**match transport {source-port | destination-port}**	Configures source-port or destination port as a key field for the flow record.
Step 9	**match ipv4 tos** **Example:** `Device (config-flow-record)# match ipv4 tos`	Configures IPv4 ToS as a key field for the flow record.
Step 10	**match ipv4 ttl** **Example:** `Device (config-flow-record)# match ipv4 ttl`	Configures IPv4 TTL as a key field for the flow record.
Step 11	**match flow direction** **Example:** `Device (config-flow-record)# match flow direction`	Specifies a match to the flow identifying fields.
Step 12	**collect counter packets long** **Example:** `Device (config-flow-record)# collect flow direction`	Configures the number of packets seen in a flow as a non-key field and enables collecting the total number of packets from the flow.
Step 13	**collect counter bytes long** **Example:** `Device (config-flow-record)# collect counter bytes long`	Configures the number of bytes seen in a flow as a non-key field and enables collecting the total number of bytes from the flow.
Step 14	**end** **Example:** `Device(config-flow-record)# end`	Returns to privileged EXEC mode.
Step 15	**show flow record** **Example:** `Device # show flow record`	Displays information about all the flow records.

Creating a Flow Exporter

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device(config)# configure terminal`	Enters global configuration mode.
Step 3	**flow exporter** *flow_exporter_name* **Example:** `Device(config)# flow exporter flow-exporter-1`	Enters flow exporter configuration mode.
Step 4	**description** *description* **Example:** `Device(config-flow-exporter)# description flow-exporter-1`	(Optional) Creates a description for the flow exporter.
Step 5	**destination** { *hostname* | *ipv4-address* | *ipv6-address* } **Example:** `Device (config-flow-exporter)# destination 10.10.1.1`	Specifies the hostname, IPv4 or IPv6 address of the system to which the exporter sends data.
Step 6	**source** *interface-type interface-name* **Example:** `Device (config-flow-exporter)# destination 10.10.1.1`	Specifies the local interface from which the exporter will use the IP address as the source IP address for exported datagrams.
Step 7	**end** **Example:** `Device(config-flow-record)# end`	Returns to privileged EXEC mode.
Step 8	**show flow exporter** **Example:** `Device # show flow exporter`	Displays information about all the flow exporters.

Creating a Flow Monitor

You can create a flow monitor and associate it with a flow record.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device(config)# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor** *monitor-name* **Example:** `Device (config)# flow monitor flow-monitor-1`	Creates a flow monitor and enters flow monitor configuration mode.
Step 4	**description** *description* **Example:** `Device (config-flow-monitor)# description flow-monitor-1`	(Optional) Creates a description for the flow monitor.
Step 5	**record** *record-name* **Example:** `Device (config-flow-monitor)# record flow-record-1`	Specifies the name of a record that was created previously.
Step 6	**exporter** *exporter-name* **Example:** `Device (config-flow-monitor)# exporter flow-exporter-1`	Specifies the name of an exporter that was created previously.
Step 7	**cache type normal** {**timeout** | **active** | **inactive**} | **type normal**	(Optional) Specifies to configure flow cache parameters.
Step 8	**end** **Example:** `Device(config-flow-record)# end`	Returns to privileged EXEC mode.
Step 9	**show flow monitor** **Example:** `Device # show flow monitor`	Displays information about all the flow monitors.

Applying Flow Monitor to an Interface

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device(config)# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type interface-name*	Specifies an interface and enters interface configuration mode.
Step 4	**no switchport** **Example:** `Device(config-if)# description no switchport`	For physical ports only, enters Layer 3 mode.
Step 5	**vrf forwarding** *vrf-name*	Associates the VRF with the Layer 3 interface.
Step 6	{**ip** | **ipv6**} **flow-monitor** *monitor-name* **input**	Associates a flow monitor to the interface for input packets.
Step 7	**end** **Example:** `Device(config-flow-record)# end`	Returns to privileged EXEC mode.
Step 8	**show flow interface** **Example:** `Device# show flow interface`	Displays the status of NetFlow (enabled or disabled) on the specified interface.

Configuration Examples for IP-aware Netflow for VRF Ingress

The **show flow interface** command displays information about Netflow on the specified interface. :

```
Interface TenGigabitEthernet1/0/36
FNF:  monitor: v4vrfingress
 direction: Input
 traffic(ip): on
FNF: monitor: v6vrfingress
```

```
 direction: Input
 traffic(ipv6): on
```

The **show flow monitor** *flow-monitor-name* **cache** command displays the contents of the cache for the flow monitor.

```
  Cache type:                               Normal (Platform cache)
  Cache size:                                10000
  Current entries:                             100

  Flows added:                                 100
  Flows aged:                                    0

IPV4 SOURCE ADDRESS:       108.3.20.100
IPV4 DESTINATION ADDRESS:  108.2.20.100
TRNS SOURCE PORT:          0
TRNS DESTINATION PORT:     0
FLOW DIRECTION:            Input
IP VERSION:                4
IP TOS:                    0x20
IP PROTOCOL:               255
IP TTL:                    64
counter bytes long:        2956000
counter packets long:      2000
```

The **show flow exporter** command displays information about all the flow exporters. :

```
Flow Exporter v4vrfingress:
  Description:              User defined
  Export protocol:          NetFlow Version 9
  Transport Configuration:
    Destination type:       IP
    Destination IP address: 15.15.15.16
    Source IP address:      15.15.15.15
    Source Interface:       TenGigabitEthernet1/0/1
    Transport Protocol:     UDP
    Destination Port:       9995
    Source Port:            52319
    DSCP:                   0x0
    TTL:                    255
    Output Features:        Used
Flow Exporter v6vrfingress:
  Description:              User defined
  Export protocol:          NetFlow Version 9
  Transport Configuration:
    Destination type:       IP
    Destination IP address: 15.15.15.16
    Source IP address:      15.15.15.15
    Source Interface:       TenGigabitEthernet1/0/1
    Transport Protocol:     UDP
    Destination Port:       9995
    Source Port:            50881
    DSCP:                   0x0
```

```
    TTL:                    255
    Output Features:        Used
```

The **show platform software fed switch active fnf monitors-dump** displays Netflow monitors dump.

```
FNF Monitors
============
Monitor (0x7f4afc031748):
    profile_id(c461d4fe) ref_ct(1) wdavc_monitor(0)
wdavc_monitor_create_requested(False)
wdavc_remote_monitoring_remote_caching(0) flags(0x0000) is_wireless(No)
is_etta_over_fnf No ettaOrBaseProfile(00000000) etta_refcnt(0)
field(113) size(16) param(0) flags(1) offset(0)
field(114) size(16) param(0) flags(1) offset(16)
field(118) size(2) param(0) flags(1) offset(32)
field(119) size(2) param(0) flags(1) offset(34)
field(156) size(1) param(0) flags(1) offset(36)
field(181) size(8) param(0) flags(0) offset(37)
field(42) size(1) param(0) flags(1) offset(45)
field(46) size(1) param(0) flags(1) offset(46)
field(43) size(1) param(0) flags(1) offset(47)
field(47) size(1) param(0) flags(1) offset(48)
Monitor (0x7f4afc029338):
    profile_id(74c02ab0) ref_ct(1) wdavc_monitor(0)
wdavc_monitor_create_requested(False)
wdavc_remote_monitoring_remote_caching(0) flags(0x0000) is_wireless(No)
is_etta_over_fnf No ettaOrBaseProfile(00000000) etta_refcnt(0)
field(93) size(4) param(0) flags(1) offset(0)
field(94) size(4) param(0) flags(1) offset(4)
field(118) size(2) param(0) flags(1) offset(8)
field(119) size(2) param(0) flags(1) offset(10)
field(156) size(1) param(0) flags(1) offset(12)
field(177) size(8) param(0) flags(0) offset(13)
field(181) size(8) param(0) flags(0) offset(21)
field(42) size(1) param(0) flags(1) offset(29)
field(43) size(1) param(0) flags(1) offset(30)
field(46) size(1) param(0) flags(1) offset(31)
field(47) size(1) param(0) flags(1) offset(32)
```

Feature History for IP-aware Netflow for VRF Ingress

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 Fuji 16.8.1a	IP-aware Netflow for VRF Ingress	This feature was introduced.