



Inband Network Telemetry

This chapter contains the following topics:

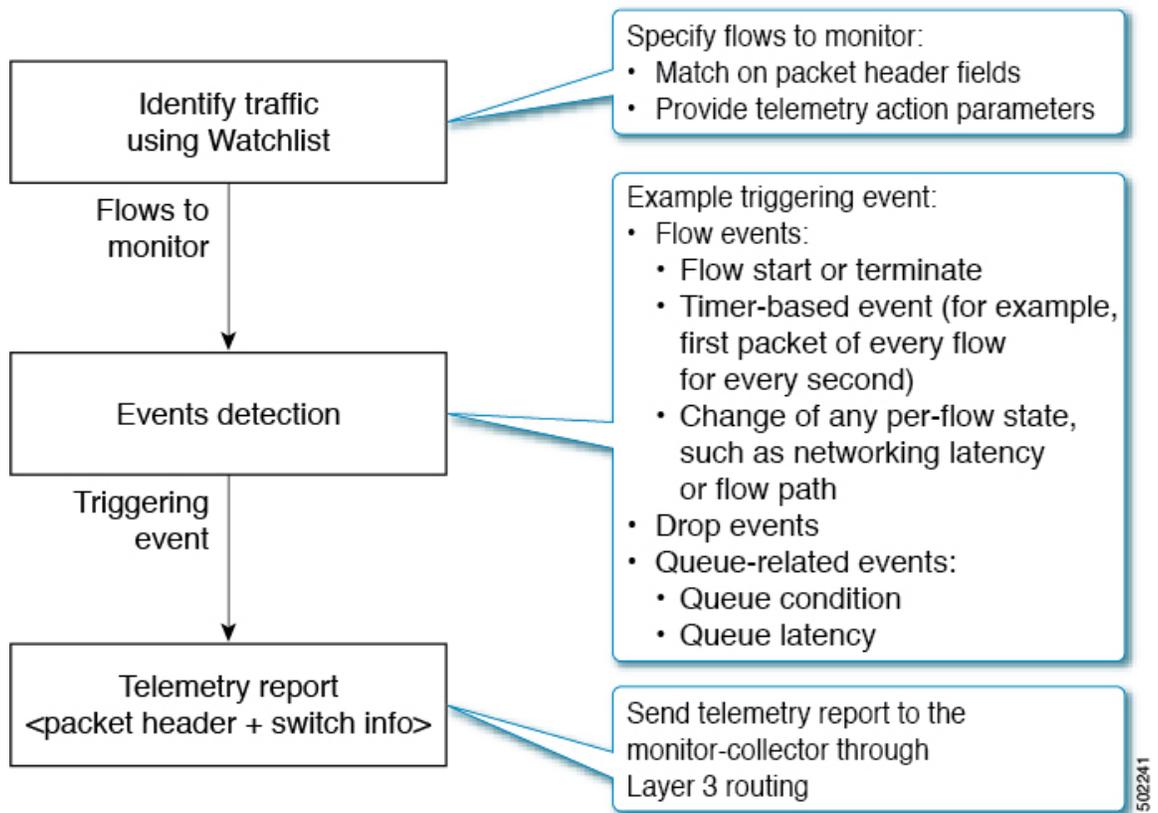
- [About Inband Network Telemetry, on page 1](#)
- [Licensing Requirements for Inband and Postcard Telemetry, on page 6](#)
- [INT Collection Parameters, on page 7](#)
- [Example Inband Network Telemetry Headers, on page 7](#)
- [Telemetry Report Formats, on page 9](#)
- [Guidelines and Restrictions, on page 10](#)
- [Configuring Inband Network Telemetry, on page 11](#)
- [Flow and Queue Profile Arguments for Report Types, on page 17](#)
- [Configuring Packet Postcards, on page 17](#)

About Inband Network Telemetry

Inband Network Telemetry (INT) is a framework that is designed to monitor, collect, and report flows and network states, by the data plane, without requiring intervention or work by the control plane. INT sources (applications, end-host networking stacks, hypervisors, NICs, send-side TORs, and so on) embed the monitored INT flow information in normal data packets, and all the downstream devices add the same information with their details that are based on the source information received.

Inband Network Telemetry is used to achieve per-packet network visibility with low overheads. The following figure shows the typical workflow for Inband Network Telemetry.

Figure 1: Example of Inband Network Telemetry Workflow



As part of the Inband Network Telemetry process, every data packet is inspected without interfering with other packet-processing logic in the switch data plane. Not all the components that are shown in the preceding figure are necessarily located in each INT switch. For example, the watchlist component can be in one switch, and the event detection and telemetry report components can be located in another switch.

A telemetry watchlist table specifies the flows to monitor. It performs a match on the packet headers and switch ports, and provides telemetry action parameters. Packets that match the watchlist entries with telemetry actions are processed by the event detection logic. If a triggering event is detected, the switch generates a telemetry report to the monitor-collector. The report message includes packet header and switch metadata associated with the packet (for example, timestamp, ingress or egress ports, and queue depth or latency).

Terminology

Following are terms used for this feature:

- **INT Header** — A packet header that carries INT information.
- **INT Packet** — A packet containing an INT Header.
- **INT Instruction** — Instructions that are embedded in the INT header, indicating which INT Metadata (defined below) to collect at each INT switch. The collected data is written into the INT Header.
- **INT Source** — A trusted entity that creates and inserts INT Headers into the packets it sends.

- **INT Sink** — A trusted entity that extracts the INT Headers and collects the path state that is contained in the INT Headers. The INT Sink is responsible for removing INT Headers so that INT is transparent to upper layers.
- **INT Metadata** — Information that an INT Source inserts into the INT Header. Examples of metadata are described in [INT Collection Parameters, on page 7](#).
- **INT Domain** — A group of interconnected INT switches under the same administration.

INT Packet Flow

At INT, source flows to be monitored are filtered using the flow watchlist. Various pieces of information on the filtered flows are extracted, based on the collect parameters enabled, such as ingress or egress port ID, ingress or egress timestamp, switch ID, and queue occupancy. The data packets are encapsulated with the INT information and sent toward the destination. The telemetry INT reports are then sent to the monitor-collector, based on whether the events are flow events, packet-drop events, or queue-congestion events.

Report Types

Following are the different types of reports used by INT:

- **Local flow reports** — A general telemetry report, which is generated from flow events. Sent from the source or sink for host-to-host data flows matching the watchlist.
- **Drop reports** — A general telemetry report, which is generated from drop events. Sent for certain supported drops. Every INT-enabled switch sends these reports to the monitor-collector.
- **Queue Congestion reports** — A general telemetry report, which is generated from queue-related events. Sent for packets exceeding the queue depth or latency. Every INT-enabled switch sends these reports to the monitor-collector.
- **INT reports** — A report that is specific to INT. Sent by the sink. When INT-encapsulated data packets are received on the sink fabric port, two reports are generated by the sink:
 - Local report for traffic arriving on a fabric port
 - INT report for data that is received from the source

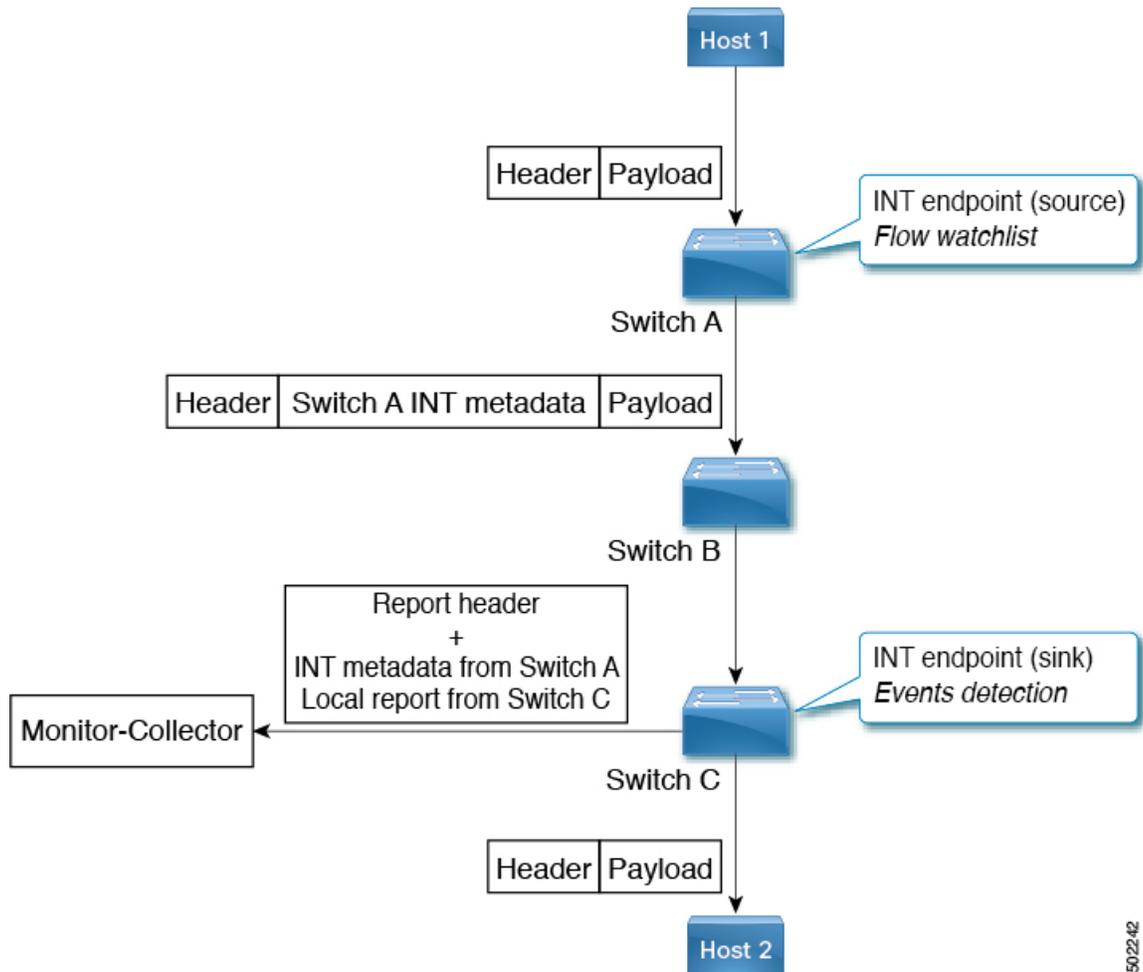
About Flow Events

Each switch can play the role of *endpoint* for INT-enabled packets. The endpoint acts both as *source* and *sink*, where:

- *Source* initiates INT operations by inserting a telemetry header into a packet and then instructing downstream network devices along the routing path to add desired telemetry information into the packet.
- *Sink* retrieves the embedded or collected INT information from the received data packets, extracts the telemetry information from the incoming packets, and sends telemetry reports to the monitor-collector if triggering flow events are detected.

The following figure shows an example flow event for Inband Network Telemetry. Switches along the route path add switch metadata into the packet header based on the telemetry instructions that are carried in the telemetry header.

Figure 2: Example Flow Event, Source, and Sink



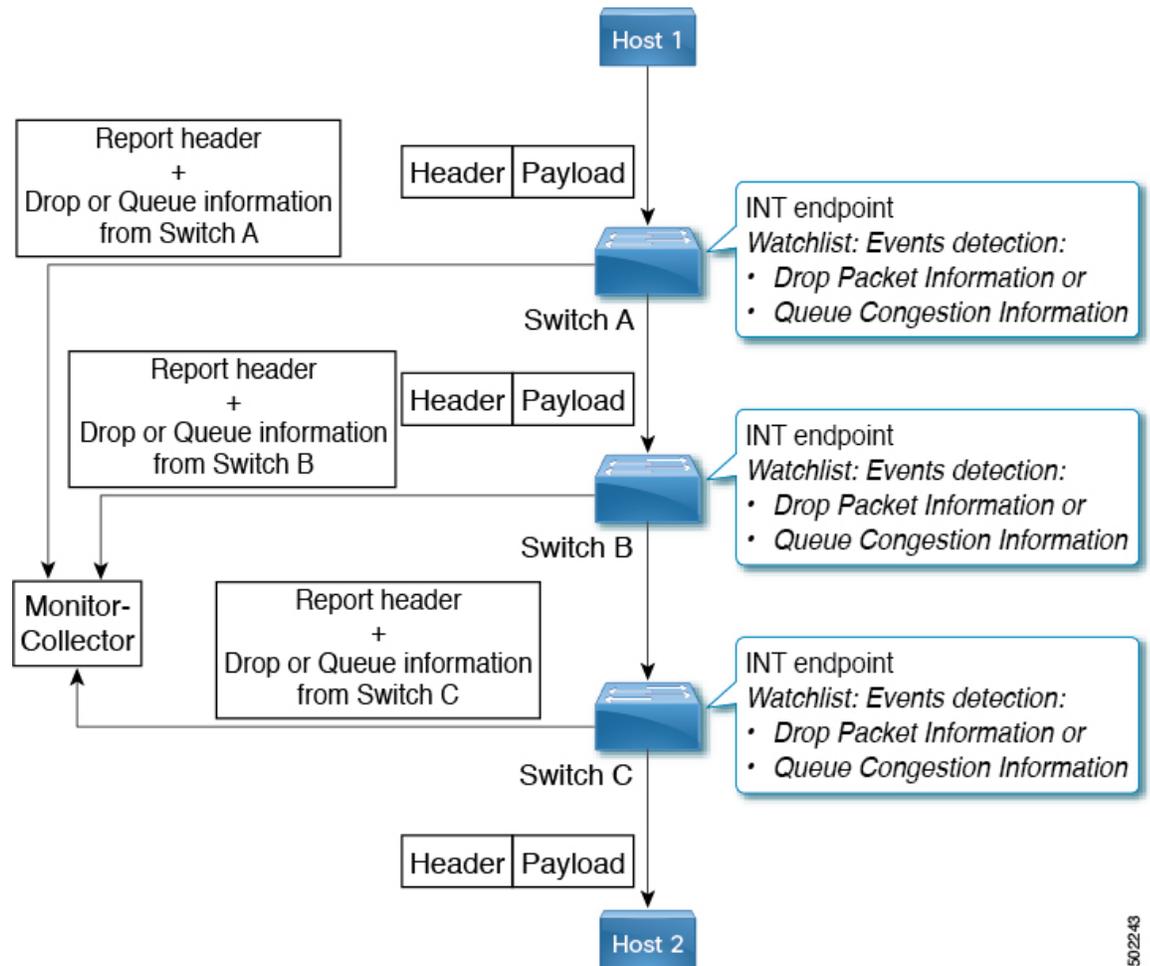
About Drop Packet and Queue Congestion Events

Each switch provides a report to the monitor-collector, based on the type of triggering event that occurred:

- **Dropped Packets** — Drop reports are generated for most dropped packets, except for STP-blocked ports and ACL deny intentional drop packets. This provides visibility into the impact of packet drops on user traffic.
- **Congested Queues** — Queue congestion reports are generated for traffic entering a specific queue during a period of queue congestion. This provides visibility into the traffic causing and prolonging queue congestion.

The following figure shows an example of drop and queue congestion reports.

Figure 3: Example Drop and Queue Congestion Reports



502243

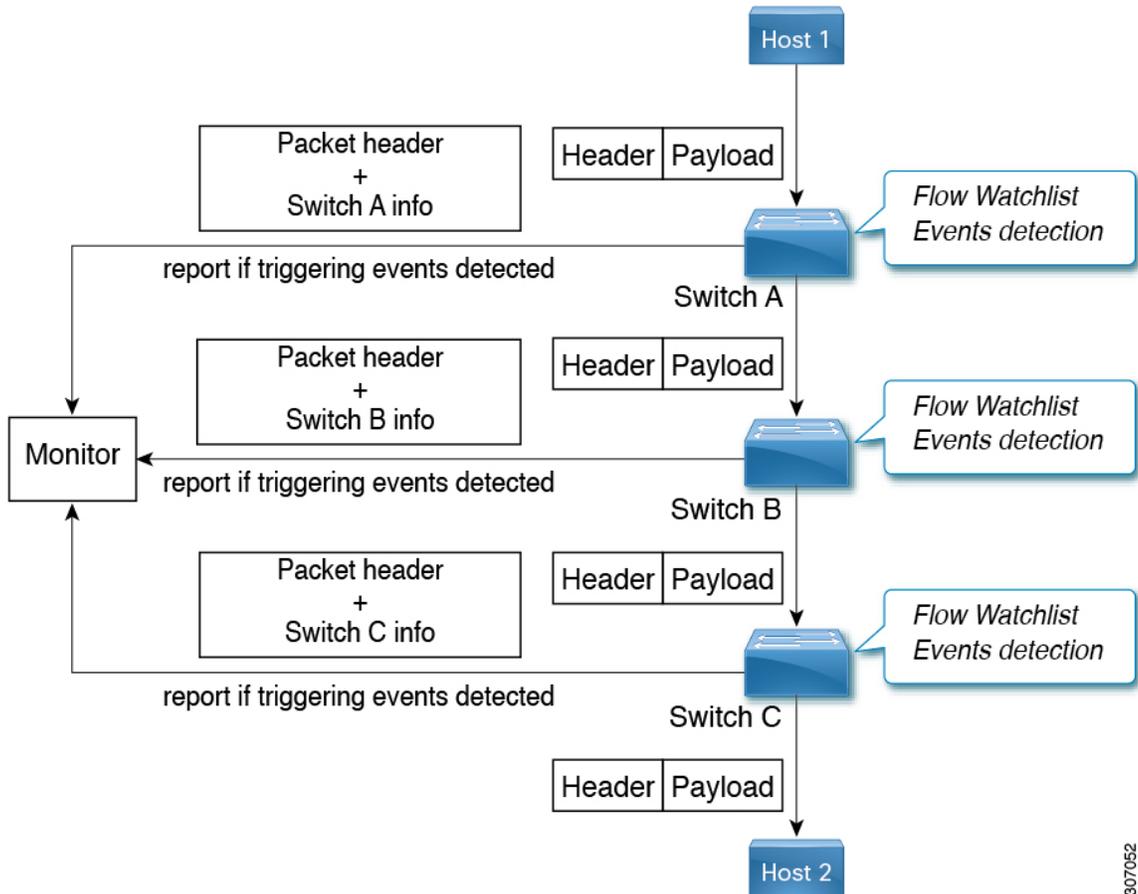
About Packet Postcards

Beginning in NX-OS release 9.2.(3), packet postcards are a different way of transmitting the telemetry data to a monitor. With packet postcards, the host's data packet is sent to another host, just like with the traditional model of INT, which uses source and sink switches as INT endpoints. However, the way that telemetry data is processed is different.

- In the traditional model, a host sends a flow, an INT source switch adds telemetry data to the host's packet, then the sink switch extracts all telemetry from each hop in the flow. The INT sink switch sends all telemetry data for the flow at once, which results in a larger amount of data that is transmitted to the monitor at one time.
- With packet postcards, each switch in the INT domain can extract telemetry data and send it to the collector. The result is faster convergence, distributed packet processing, and less overhead than with a traditional model.

The following figure shows an example of packet postcards on a flow.

Figure 4: Example of Packet Postcards



307052

With packet postcards, the defined roles of source and sink switches become irrelevant because telemetry data is communicated at each switch.

- If the packet or flow matches the watchlist conditions, the comparison is true, and event-detection logic generates the report and sends it to the monitor. Triggering events are detected based on the switch local information such as ingress or egress ports and queuing latency for the monitored flow. Unlike standard Inband Network Telemetry, a postcard switch never modifies the original data packets by adding INT metadata. After telemetry data is sent to the collector, the packet is then forwarded onto the network to either its next hop or destination.
- If the packet or flow does not match the watchlist conditions, the comparison is false, and the event-detection logic does not process the telemetry data. The packet is forwarded by the switch, continuing along the route to its next hop. If it ingresses another switch in the INT domain, the packet is compared against that switch's watchlist, if necessary a packet postcard is sent to the monitor-collector, and the packet is forwarded until it reaches its eventual destination.

Licensing Requirements for Inband and Postcard Telemetry

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	Inband Telemetry or Packet Postcards requires a N3K-STR1K9 license. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the Cisco NX-OS Licensing Guide .

INT Collection Parameters

The following information can be collected for a given flow for INT encapsulation.

- **Switch ID** — The unique ID of a switch. Switch IDs must be unique within an INT domain. The switch ID is generated automatically from bytes [0][1][4][5] of the VDC MAC address. The switch ID is collected by default.

For example, using the **show vdc** command to get the VDC MAC address on this device:

```
switch(config-int)# show vdc

vdc_id  vdc_name  state  mac                type  lc
-----  -
1       bod-int2  active b0:12:80:8f:e3:45 Ethernet None
```

In this case, the switch ID for this switch is 0xB012E345.

- **Ingress port identifier** — The port on which the INT packet was received.
- **Ingress timestamp** — The device local time when the INT packet was received on the ingress physical or logical port.
- **Egress port identifier** — The port on which the INT packet was sent out.
- **Egress timestamp** — The device local time when the physical or logical port processed the INT packet.
- **Queue occupancy** — The build-up of traffic in the queue (in bytes, cells, or packets) that the INT packet observes in the device while being forwarded.

Example Inband Network Telemetry Headers

This section shows example INT headers with two hosts (Host 1 and Host 2), communicating over a network path composed of three switches (Switch 1, Switch 2 and Switch 3) as shown in the following example.

```
==> packet P travels from Host 1 to Host 2 ==>

Host 1 -----> Switch 1 -----> Switch 2 -----> Switch 3 -----> Host 2
```

In this scenario, Host 1 sends a TCP packet to Host 2. The ToR switch of Host 1 (Switch 1) acts as the INT source. It adds INT headers and its own metadata in the packet.

Finally, the ToR switch of Host 2 (Switch 3) acts as the INT sink and removes INT headers before forwarding the packet to Host 2.

The following is the packet that INT sink Switch 3 receives, starting from the IPv4 header.

IP Header

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Ver=4 | IHL=5 | DSCP=0x17 |ECN|           Length           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Identification           |Flags|   Fragment Offset   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Time to Live | Proto = 6 |           Header Checksum       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Source Address           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Destination Address     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

TCP Header

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Source Port           | Destination Port = 22 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Sequence Number       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Acknowledgment Number |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Data |           |U|A|P|R|S|F|           | |
| Offset| Reserved |R|C|S|S|Y|I|           | Window           |
|           |           |G|K|H|T|N|N|           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Checksum           |           Urgent Pointer   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

INT Shim Header for TCP/UDP

A shim header is a header that is inserted to allow for space in different packet protocol headers. In a TCP/UDP packet, following a TCP/UDP header, the shim header creates an encapsulation into which the INT headers are inserted. One boundary of the encapsulation is the TCP/UDP payload, while the other boundary of the INT header is the shim header.

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Type=1 | Reserved | Length = 8 | RESERVED |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

INT Metadata Header and Metadata Stack

The following shows a basic example of one set of instruction bit values. These instruction bit values correspond to one record consisting of one collect queue occupancy command that is configured through **inband telemetry record** command. No other collect commands are configured in the record.

The instruction set values in the INT Metadata header will differ depending on how a record is configured.

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Ver  |Rep|C|E| R | InsCnt=2| MaxHopCnt=16 | TotalHopCnt=2 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0|           Reserved           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
|                               sw id of switch A                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
|                               queue occupancy of switch A                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

TCP Payload

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
|                               TCP payload                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Telemetry Report Formats

This section specifies the packet formats for telemetry reports.

Outer Encapsulation

There are two types of reports:

- **Upstream reports** — The sink generates these reports only if there is a violation of a threshold for the flows being monitored. These reports contain the whole INT stack of embedded metadata up until the sink.
- **Local reports** — Each device generates these reports in cases of queue and drop violations. These reports contain information and metadata of the current device only.

Telemetry reports are defined using a UDP-based encapsulation. The source IP address identifies the switch that generates the telemetry report. The Destination IP address identifies a location in the distributed telemetry monitoring system that receives the telemetry report.

UDP header (8 Octets)

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
|           Source Port           |           Destination Port           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
|           Length           |           Checksum           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The Source Port can be used to carry flow entropy, for example based on a hash of the inner 5-tuple. Otherwise, it should be set to 0.

The Destination Port is user-configurable. Use the same Destination Port value for all telemetry reports in a particular deployment.

Telemetry Report Header (16+ Octets)

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Ver  |NProto |D|Q|F|           Reserved           | hw_id  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
|                               Sequence Number      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
|                               Ingress Timestamp     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

- **Ver** — Version 0.5.
- **NProto** — Next Protocol. Indicates:
 - 0: Ethernet
 - 1: Telemetry Drop Header, followed by Ethernet
 - 2: Telemetry Switch Local Header, followed by Ethernet
- **D|Q|F** — An enumeration that indicates the reason why a packet was dropped:
 - **D: Dropped** — Indicates that at least one packet matching a drop watchlist was dropped.
 - **Q: Congested Queue Association** — Indicates the presence of congestion on a monitored queue.
 - **F: Tracked Flow Association** — Indicates that this telemetry report is for a tracked flow (for example, if the packet matched a flow watchlist somewhere, for INT, or locally, for postcard). The report can include INT metadata beyond the inner Ethernet header. Other telemetry reports are likely to be received for the same tracked flow, from the same switch, and from other switches, for drop reports, postcards, or path changes.
- **hw_id** — Identifies the hardware subsystem within the switch that generated this report. For example, in a chassis with multiple line cards, the `hw_id` could identify a specific line card, or a subsystem within a line card.
- **Sequence Number** — Reflects the sequence of reports from a specific `hw_id` to a particular telemetry report destination. The sequence number can be used to detect loss of telemetry reports before they reach their intended destination.
- **Ingress Timestamp** — The device local time when the packet was first received on the ingress physical or logical port, in nanoseconds.

Guidelines and Restrictions



Note Beginning with Cisco NX-OS Release 9.3(3), this feature is no longer supported.

Following are the guidelines and restrictions for the INT feature:

- For information about supported platforms, see [Supported Platforms for Programmability Features](#).
Ensure the data that you are using for the DSCP value is different from the DSCP value that the INT flow-profile configured.
- Only flows with the Layer 4 header are exported by INT. Layer 2-only or IP-only flows with no Layer 4 header are not supported. Support exists only for IP TCP, UDP, and ICMP flows.
- INT is supported only for IPv4. IPv6 and multicast is not supported.
- When INT is enabled, it adds bytes to the data packet. Verify that the required MTU is configured end-to-end throughout the network to accommodate for the increased overhead for the INT-encapsulated packet.
- Only Default VRF is supported for exporter reachability.
- Only IPv4 Destination is supported under the exporter.
- Upstream reports are not exported on Sink when SPAN is enabled. Only local reports are allowed.
- Drop reports that are exported are not throttled.
- By default, drop reporting is suppressed for both user ACL deny drops and drops on spanning-tree (STP) blocked ports.
- Ingress MTU failure drops are not reported, as the drops happen before the reporting occurs.
- The current scale limit is 20,000 flows per pipeline.
- As a best practice, Cisco recommends using specific flow watchlists to filter flows for reporting.
- INT reports timestamp is 32 bit, which will not reflect the exact time when the flow comes to the switch. You will need a mechanism on the monitor-collector to correlate and derive the time for when the flow is received.
- Only one monitor-collector is currently supported.
- The programming of the watchlist hardware entries may take a minute or two if the number of watchlist entries is greater than 345.

Configuring Inband Network Telemetry

Telemetry flow reports are generated for flows matching the watchlist. For drop and queue reports, the watchlist is not required.

Before you begin

Make sure that standard packet postcard mode is not enabled. Traditional INT and packet postcards are mutually exclusive. If standard packet postcards are configured, and you try to configure INT, the following error is displayed.

```
Error: Cannot configure inband-telemetry when postcard is configured.
```

If you see this error, you must disable standard INT:

```
switch(config)# no feature postcard-telemetry
```

Also, INT and packet postcard mode are licensed features. Make sure that you have enabled the correct license. See [Licensing Requirements for Inband and Postcard Telemetry, on page 6](#).

Procedure

Step 1 For all three types of INT reports: Enable the **hardware-telemetry** feature on all switches in the INT domain.

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# feature hardware-telemetry
HW Telemetry enabled
switch(config)#
switch(config)# hardware-telemetry inband-telemetry
```

Step 2 For flow reports: For each switch in the INT domain, configure an INT record.

```
switch(config)# inband-telemetry record record-ID
switch(config-int-record)# collect port-id
switch(config-int-record)# collect queue-occupancy
switch(config-int-record)# collect ingress-timestamp
switch(config-int-record)# collect egress-timestamp
switch(config-int-record)# exit
```

Where *record-ID* is the name of the record. The default switch-id option is present when a new record is created.

This command affects only INT reports. Drop reports, queue congestion reports, and local flow reports are not affected.

For example:

```
switch(config)# inband-telemetry record r1
switch(config-int-record)# collect port-id
switch(config-int-record)# collect queue-occupancy
switch(config-int-record)# collect ingress-timestamp
switch(config-int-record)# collect egress-timestamp
switch(config-int-record)# exit
```

Step 3 For all three types of INT reports: For each switch in the INT domain, configure an INT exporter.

```
switch(config)# inband-telemetry exporter exporter-ID
switch(config-int-exporter)# destination destination-IP-address
switch(config-int-exporter)# source loopback loopback-interface-number
switch(config-int-exporter)# transport udp udp-port
switch(config-int-exporter)# exit
```

Where:

- *exporter-ID* is the name that you want to give to the exporter on this particular switch.
- *destination-IP-address* is a destination reachable through the default VRF.
- *loopback-interface-number* is the number for the loopback interface.

- *udp-port* is the destination UDP port. Acceptable values are from 1 to 65535. The default UDP port where exports are sent is 31337.

For example:

```
switch(config)# inband-telemetry exporter e1
switch(config-int-exporter)# destination 17.18.19.20
switch(config-int-exporter)# source loopback 36
switch(config-int-exporter)# transport udp 1000
switch(config-int-exporter)# exit
```

Step 4 For all three types of INT reports: For each switch in the INT domain, configure an INT watchlist.

A watchlist is mandatory when the monitor is applied at the system level. A watchlist is applicable to flow monitoring only. A watchlist is not effective for drop or queue congestion reports.

Be careful when configuring watchlists. Try to avoid configuring watchlists that match on control traffic going to the local CPU and generated from the local CPU.

```
switch(config)# inband-telemetry watchlist ip watchlist-ID
switch(config-int-watchlist)# 20 permit ip permit-IP-address-subnet-1
permit-IP-address-subnet-2
switch(config-int-watchlist)# 10 deny ip deny-IP-address-subnet-1 deny-IP-address-subnet-2
switch(config-int-watchlist)# exit
```

Where:

- *watchlist-ID* is the name of the watchlist. A maximum of 512 entries are supported under one watchlist. Watchlist entries are needed on the source to match specific flows that must be exported.
- *permit-IP-address-subnet-#* are the permit IP addresses and subnets.
- *deny-IP-address-subnet-#* are the deny IP addresses and subnets.

For example:

```
switch(config)# inband-telemetry watchlist ip w11
switch(config-int-watchlist)# 20 permit ip 1.2.3.4/24 5.6.7.8/32
switch(config-int-watchlist)# 10 deny ip 20.30.40.50/24 60.70.80.90/16
switch(config-int-watchlist)# exit
```

Step 5 For all three types of INT reports: For each switch in the INT domain, configure an INT monitor.

```
switch(config)# inband-telemetry monitor monitor-ID
switch(config-int-monitor)# record record-ID
switch(config-int-monitor)# exporter exporter-ID
switch(config-int-monitor)# watchlist watchlist-ID
switch(config-int-monitor)# exit
```

Where:

- *monitor-ID* is the name of this INT monitor.
- *record-ID* is the name of the record that you entered in [Step 2, on page 12](#).
- *exporter-ID* is the name of the exporter that you entered in [Step 3, on page 12](#).
- *watchlist-ID* is the name of the watchlist that you entered in [Step 4, on page 13](#).

For example:

```
switch(config)# inband-telemetry monitor m1
switch(config-int-monitor)# record r1
switch(config-int-monitor)# exporter e1
switch(config-int-monitor)# watchlist w1
switch(config-int-monitor)# exit
```

Step 6 For flow reports: Configure a port-type fabric to encapsulate the INT domain.

The port-type fabric must be enabled on all links connecting leaf and spine.

```
switch(config)# interface ethernet interface1/interface2
switch(config-if)# port-type fabric
switch(config-if)# exit
```

Where *interface#* is the Ethernet interface that you are configuring.

For example:

```
switch(config)# interface ethernet 1/1
switch(config-if)# port-type fabric
switch(config-if)# exit
```

Step 7 For all three types of INT reports: Apply an INT monitor at both the source and the sink.

```
switch(config-int-record)# inband-telemetry system monitor monitor-ID
```

Where *monitor-ID* is the name of the INT monitor.

For example:

```
switch(config-int-record)# inband-telemetry system monitor m1
```

Step 8 For flow reports (optional): Configure a flow profile and queue profile.

```
switch(config-int-record)# inband-telemetry flow-profile
switch(config-int-flow-prof)# dscp dscp-value
switch(config-int-flow-prof)# age age-value
switch(config-int-flow-prof)# latency quantization latency-quant-value
exit

switch(config-int-record)# inband-telemetry queue-profile queue-profile-default
switch(config-int-q-prof)# depth depth-value
switch(config-int-q-prof)# latency latency-value
exit
```

Where the following arguments are applicable to different types of reports. See [Flow and Queue Profile Arguments for Report Types](#), on page 17.

- *dscp-value* is the flow-profile DSCP value that is used for INT. The value must match on both the source and the sink for INT encapsulation and decapsulation to happen. This value must not overlap with the DSCP values that are used in the data traffic. Allowed values: 1, 2, 4, 8, 16, 32, 63. Default is 1.
- *age-value* is the flow-profile age, in seconds, when the exports are sent out. Used to determine the frequency at which a given flow is reported. Allowed values: 0 to 7200. Default is 30.
- *latency-quant-value* is the flow-profile latency quantization value. Flow reports are generated when the flow is exceeding the configured latency value. Only flows that match the watchlist are generated. This

value is converted to a power of 2, in nanoseconds. For example, a value of 11 is converted to 2 power 11, or 2048. This calculation means that if the difference between the ingress timestamp and the egress timestamp is more than 2048 nanoseconds at any point in the path, one upstream or local report is generated per flow. Allowed values are 0 to 28. Default is 11.

- *depth-value* is the queue-profile depth value, in number of cells. Congestion queue reports are generated for flows exceeding the configured queue depth. Allowed values: 1 to 204800. Default is 300.

Adjust the depth value as needed for the number of reports that are generated. For example, if your deployment is generating too many reports, increase the depth value to reduce the number of reports. Conversely, if you need to generate more reports, lower the depth value.

- *latency-value* is the queue-profile latency, in nanoseconds. The latency value is the time from when the packet is put into the queue to the time when it is dequeued from the queue. Allowed values: 30 to 8192000. Default is 2048. If a violation happens, a queue congestion report is generated from the local switch when the congestion occurs. If needed, adjust the latency value to increase or decrease the number of reports you want in your deployment.

Note The *depth-value* and *latency-value* arguments are compared, and the lowest value is used, so you do not need to configure both. If you configure one argument, the default value for the other argument is used for the comparison.

Note The *depth-value* and *latency-value* are used to derive the queue quota which controls creation of queue congestion reports. Queue congestion reports are sent when either the depth or latency that is defined for the queue profile is hit. The quota controls the number of congestion queue reports sent. You can display the queue quota value through **show inband-telemetry queue-profile**.

For example:

```
switch(config-int-record)# inband-telemetry flow-profile
switch(config-int-flow-prof)# dscp 1
switch(config-int-flow-prof)# age 10
switch(config-int-flow-prof)# latency quantization 28
exit

switch(config-int-record)# inband-telemetry queue-profile queue-profile-default
switch(config-int-q-prof)# depth 300
switch(config-int-q-prof)# latency 8192000
exit
```

Step 9 Verify that the Inband Network Telemetry was configured correctly.

- To view the INT exporter configuration information:

```
switch(config-int-watchlist)# show inband-telemetry exporter exporter-ID
```

For example:

```
switch(config-int-watchlist)# show inband-telemetry exporter e1
INT exporter e1:
  Destination: 10.1.1.2
  VRF: default
  Destination UDP Port 1000
  Source Interface mgmt0 (209.165.202.129)
```

- To view the INT watchlist configuration information:

```
switch(config-int-watchlist)# show inband-telemetry watchlist watchlist-ID
```

For example:

```
switch(config-int-watchlist)# show inband-telemetry watchlist w11
INT watchlist w11:
  No. of users: 1
  No. of aces: 1
  20 permit ip 1.2.3.4/24 5.6.7.8/32
```

The **No. of users:** field displays the number of ACE entries that are used in the watchlist when the monitor is applied.

- To view the INT monitor configuration information:

```
switch(config-int-monitor)# show inband-telemetry monitor monitor-ID
```

For example:

```
switch(config-int-monitor)# show inband-telemetry monitor m1
INT Monitor m1:
  INT Record: r1
  INT Exporter: e1
  INT Watchlist: w11
switch(config-int-monitor)#
```

- To display the monitor session that is active at the system level:

```
switch(config-int-watchlist)# show inband-telemetry sessions
INT Sessions:
  system monitor:m1
```

- To view the running configuration of the Inband Network Telemetry:

```
switch(config)# show running-config hardware-telemetry
!Running configuration last done at: Tue Oct 30 15:24:08 2018
!Time: Tue Oct 30 15:24:13 2018

version 9.2(2) Bios:version 05.25
feature hardware-telemetry

hardware-telemetry inband-telemetry
  inband-telemetry exporter e1
    destination 10.1.1.2
    transport udp 1000
  inband-telemetry record r1
    collect switch-id
    collect port-id
    collect queue-occupancy
    collect ingress-timestamp
    collect egress-timestamp
  inband-telemetry watchlist ip w11
    20 permit ip 1.2.3.4/24 5.6.7.8/32
  inband-telemetry monitor m1
    record r1
    exporter e1
    watchlist w11
  inband-telemetry queue-profile queue-profile-default
    depth 300
    latency 8192000
  inband-telemetry flow-profile
```

```
dscp 1
age 10
latency quantization 28
inband-telemetry system monitor m1
```

Flow and Queue Profile Arguments for Report Types

The following table shows the arguments for the **inband-telemetry flow-profile** and **inband-telemetry queue-profile** commands and how they apply to the different report types. Unless documented, the arguments apply to both INT and Packet Postcards.

Table 1: Flow Reports for Specific Arguments

Report Type	Command and Arguments
Only flow reports Note The DSCP value is applicable to INT, but not Packet Postcard mode.	inband-telemetry flow-profile dscp <i>dscp-value</i> inband-telemetry flow-profile age <i>age-value</i>
Both flow and queue congestion reports	inband-telemetry flow-profile latency-quantization <i>latency-quant-value</i>
Only queue congestion reports	inband-telemetry queue-profile depth <i>depth-value</i> inband-telemetry queue-profile latency <i>latency-value</i>

Configuring Packet Postcards

With packet postcards, each switch in the INT domain performs a watchlist comparison and events detection. When an event is triggered, each switch sends its own telemetry data to the monitor-collector independent of the other switches. Each switch requires a watchlist and events detection logic.

Before you begin

Make sure that standard INT is not enabled. Packet postcard mode and the traditional INT model are mutually exclusive. If standard INT is configured, and you try to configure packet-postcard mode, the following error is displayed.

```
Error: Cannot configure postcard when inband-telemetry is configured.
```

If you see this error, you must disable standard INT:

```
switch(config)# no feature hardware-telemetry
```

Also, INT and packet postcard mode are licensed features. Make sure that you have enabled the correct license. See [Licensing Requirements for Inband and Postcard Telemetry](#), on page 6.

Procedure

Step 1 Enable the **postcard-telemetry** feature on all switches in the INT domain.

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# feature hardware-telemetry
HW Telemetry enabled
switch(config)#
switch(config)# hardware-telemetry postcard-telemetry
switch(config-postcard)#
```

Step 2 On each switch in the INT domain, configure the appropriate profile:

a) **For a flow profile:** Configure the flow profile parameters.

```
switch(config-postcard)# postcard-telemetry flow-profile
switch(config-postcard-flow-prof)# age age-value
switch(config-postcard-flow-prof)# latency quantization latency-quant-value
exit
```

Where the following arguments are applicable to different types of reports. See [Flow and Queue Profile Arguments for Report Types, on page 17](#).

- *age-value* is the flow-profile age, in seconds, when the exports are sent out. Used to determine the frequency at which a given flow is reported. Allowed values: 0 to 7200. Default is 30.
- *latency-quant-value* is the flow-profile latency quantization value. Flow reports are generated when the flow exceeds the configured latency value. Only flows that match the watchlist are generated. This value is converted to a power of 2, in nanoseconds. For example, a value of 11 is converted to 2 power 11, or 2048. This calculation creates the difference between the ingress timestamp and the egress timestamp. If the difference is more than 2048 nanoseconds at any point in the path, one upstream or local report is generated per flow. When the difference between the ingress and egress timestamp goes below the configured value, a recovery report is sent. Allowed values are 0 to 28. Default is 11.

For example:

```
switch(config-postcard)# postcard-telemetry flow-profile
switch(config-postcard-flow-prof)# age 45
switch(config-postcard-flow-prof)# latency quantization 4
switch(config-postcard-flow-prof)# exit
switch(config-postcard)#
```

b) **For a queue congestion profile:** Configure the watchlist parameters for the default queue profile.

```
switch(config-postcard)# postcard-telemetry queue-profile queue-profile-default
switch(config-postcard-q-prof)# depth depth-value
switch(config-postcard-q-prof)# latency latency-value
exit
```

Where the following arguments are applicable to different types of reports. See [Flow and Queue Profile Arguments for Report Types, on page 17](#).

- *depth-value* is the queue-profile depth value, in number of cells. Congestion queue reports are generated for flows exceeding the configured queue depth. Allowed values: 1 to 204800. Default is 300.

Adjust the depth value as needed for the number of reports that are generated. For example, if your deployment is generating too many reports, increase the depth value to reduce the number of reports. Conversely, if you need to generate more reports, lower the depth value.

- *latency-value* is the queue-profile latency, in nanoseconds. The latency value is the time from when the packet is put into the queue to the time when it is dequeued from the queue. Allowed values: 30 to 8192000. Default is 2048. If a violation happens, a queue congestion report is generated from the local switch when the congestion occurs. If needed, adjust the latency value to increase or decrease the number of reports you want in your deployment.

Note The *depth-value* and *latency-value* arguments are compared, and the lowest value is used, so you do not need to configure both. If you configure one argument, the default value for the other argument is used for the comparison.

Note The *depth-value* and *latency-value* are used to derive the queue quota which controls creation of queue congestion reports. Queue congestion reports are sent when either the depth or latency that is defined for the queue profile is hit. The quota controls the number of congestion queue reports that are sent. You can display the queue quota value through **show postcard-telemetry queue-profile**.

For example:

```
switch(config-postcard)# postcard-telemetry queue-profile queue-profile-default
switch(config-postcard-q-prof)# depth 200
switch(config-postcard-q-prof)# latency 2048
exit
```

Step 3 On each switch, configure a postcard exporter.

```
switch(config)# postcard-telemetry exporter exporter-ID
switch(config-postcard-exporter)# destination destination-IP-address
switch(config-postcard-exporter)# source interface-type
switch(config-postcard-exporter)# transport udp udp-port
switch(config-postcard-exporter)# exit
```

Where:

- *exporter-ID* is the name that you want to give to the exporter on this particular switch.
- *destination-IP-address* is a destination of the monitor-collector reachable through the default VRF. The destination can be an IPv4 address.
- *source* is any of the following interface types:
 - *ethernet* plus an *ethernet-interface* specifies an Ethernet IEEE 802.3z interface.
 - *loopback* plus a *loopback-interface* specifies a loopback interface.
 - *mgmt* plus a *management-interface* specifies a management interface.
 - *port-channel* plus a *port-channel-interface* specifies a port-channel interface.
 - *vlan* plus a *vlan-id* specifies a VLAN interface.
- *udp-port* is the destination UDP port. Acceptable values are from 1 to 65535. The default UDP port where exports are sent is 31337.

For example:

```
switch(config)# postcard-telemetry exporter ex4
switch(config-postcard-exporter)# destination 192.0.2.12
switch(config-postcard-exporter)# source loopback 40
switch(config-postcard-exporter)# transport udp 999
switch(config-postcard-exporter)# exit
```

Step 4 On each switch in the INT domain, configure a watchlist.

A watchlist is mandatory when the monitor is applied at the system level. However, a watchlist is not effective for drop or queue congestion reports.

Be careful when configuring a watchlist. Try to avoid configuring watchlists that match on control traffic going to the local CPU and generated from the local CPU.

```
switch(config)# postcard-telemetry watchlist ip watchlist-ID
switch(config-postcard-watchlist)# sequence-number permit ip permit-IP-address-subnet-1
permit-IP-address-subnet-2
switch(config-postcard-watchlist)# sequence-number deny ip deny-IP-address-subnet-1
deny-IP-address-subnet-2
switch(config-postcard-watchlist)# exit
```

Where:

- *watchlist-ID* is the name of the watchlist. A maximum of 512 entries are supported under one watchlist. Watchlist entries are needed on the source to match specific flows that must be exported.
- *permit-IP-address-subnet-#* are the permit IP addresses and subnets. When you set permit, flows on the specified IP addresses and subnets are monitored.
- *deny-IP-address-subnet-#* are the deny IP addresses and subnets. When you set deny, flows on the specified IP addresses and subnets are not monitored.

For example:

```
switch(config)# postcard-telemetry watchlist ip watch1
switch(config-postcard-watchlist)# 25 permit ip 10.10.20.30/24 192.0.2.1/32
switch(config-postcard-watchlist)# 30 deny ip 192.0.2.50/24 192.168.70.80/24
switch(config-postcard-watchlist)# exit
```

Step 5 For each switch in the INT domain, configure a monitor.

```
switch(config)# postcard-telemetry monitor monitor-ID
switch(config-postcard-monitor)# exporter exporter-ID
switch(config-postcard-monitor)# watchlist watchlist-ID
switch(config-postcard-monitor)# exit
```

Where:

- *monitor-ID* is the name of this INT monitor.
- *exporter-ID* is the name of the exporter that you entered in [Step 3, on page 19](#).
- *watchlist-ID* is the name of the watchlist that you entered in [Step 4, on page 20](#).

For example:

```
switch(config)# postcard-telemetry monitor mon10
switch(config-postcard-monitor)# exporter ex4
switch(config-postcard-monitor)# watchlist watch10
switch(config-postcard-monitor)# exit
```

Step 6 Apply a monitor at all switches in the INT domain.

```
switch(config-postcard)# postcard-telemetry system monitor monitor-ID
```

Where *monitor-ID* is the name of the INT monitor.

For example:

```
switch(config-postcard)# postcard-telemetry system monitor mon10
```

Step 7 Verify that packet postcard telemetry is configured correctly.

- To view the postcard exporter configuration information:

```
switch(config-postcard-exporter)# show postcard-telemetry exporter exporter-ID
```

For example:

```
switch(config-postcard-exporter)# show postcard-telemetry exporter name ex4
POSTCARD exporter ex4:
  Destination: 192.0.2.12
  VRF: default
  Destination UDP Port 999
  Source Interface loopback40
switch(config-postcard-exporter)#
```

- To view the postcard telemetry-watchlist configuration information:

```
switch(config-postcard-watchlist)# show postcard-telemetry watchlist watchlist-ID
```

For example:

```
switch(config-postcard-watchlist)# show postcard-telemetry watchlist 27
No. of users: 1
  No. of aces: 2
    30 permit ip 192.0.2.50/24 192.168.70.80/24
    100 deny ip 10.10.10.10/32 1.2.3.4/24

switch(config-postcard-watchlist)#
```

The **No. of users:** field displays the number of ACE entries in the watchlist that are used when the monitor is applied.

- To view the postcard monitor configuration information for packet postcards:

```
switch(config-postcard-monitor)# show postcard-telemetry monitor monitor-ID
```

For example:

```
switch(config-postcard-monitor)# show postcard-telemetry monitor mon10
POSTCARD Monitor mon10:
  POSTCARD Exporter: ex4
  POSTCARD Watchlist: 27
switch(config-postcard-monitor)#
```

- To display the monitor session that is active at the system level:

```
switch(config-postcard-monitor)# show postcard-telemetry sessions
POSTCARD Sessions:
    system monitor:mon10
switch(config-postcard-monitor)#
```

- To display the queue profile configuration:

```
switch(config-postcard-q-prof)# show postcard-telemetry queue-profile
POSTCARD Queue Profile queue-profile-default:
    Fields:
        Depth: 200
        Latency: 444
        Quota configured: 34
```

- To display the flow profile configuration:

```
switch(config-postcard-flow-prof)# # show postcard-telemetry flow-profile
POSTCARD flow Profile flow-profile-default:
    Fields:
        AGE: 45
        Latency Quantization: 16 nanosec
        (User configured quantization value is 4)
```

- To view the running configuration of telemetry for packet postcards:

```
switch(config)# show running-config hardware-telemetry

!Command: show running-config hardware-telemetry
!Running configuration last done at: Fri Nov 30 18:59:24 2018
!Time: Fri Nov 30 18:59:47 2018

version 9.2(3) Bios:version 05.25
feature hardware-telemetry

hardware-telemetry postcard-telemetry
  postcard-telemetry exporter ex4
    destination 192.02.12
    transport udp 999
    source loopback40
  postcard-telemetry watchlist ip 27
    30 permit ip 192.0.2.50/24 192.168.70.80/24
    100 deny ip 10.10.10.10/32 1.2.3.4/24
  postcard-telemetry monitor mon10
    exporter ex4
    watchlist 27
  postcard-telemetry queue-profile queue-profile-default
    depth 200
    latency 444
  postcard-telemetry flow-profile
    age 45
    latency quantization 4
  postcard-telemetry system monitor mon10
```
