Hardware Telemetry

This chapter contains the following topics:

Guidelines and Limitations

The following are the guidelines and limitations for Hardware Telemetry.

- The ASIC on the switch forwards the flows only to the default VRF. There is no support for having collectors in an area other than the default VRF.

Telemetry RPM Installation Example

The following is an example of installing the telemetry RPM.

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# feature bash-shell
switch(config)# end
switch# run bash sudo su
bash-4.3# service docker start              >> Starting docker service
Free bootflash: 86936 MB, total bootflash: 116590 MB
Carving docker bootflash storage: 2000 MB
```

```
2000+0 records in
2000+0 records out
2000000000 bytes (2.0 GB) copied, 16.5267 s, 121 MB/s
mke2fs 1.42.9 (28-Dec-2013)
fs_types for mke2fs.conf resolution: 'ext4'
Discarding device blocks: done
Discard succeeded and will return 0s - skipping inode table wipe
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
122160 inodes, 488281 blocks
4882 blocks (1.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=503316480
15 block groups
32768 blocks per group, 32768 fragments per group
8144 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

tune2fs 1.42.9 (28-Dec-2013)
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          7c841fbf-1972-4082-9ea0-37b77844be3b
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype extent flex_bg
 sparse_super large_file huge_file uninit_bg dir_nlink extra_isize
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              122160
Block count:              488281
Reserved block count:     4882
Free blocks:              471692
Free inodes:              122149
First block:              0
Block size:               4096
Fragment size:            4096
Reserved GDT blocks:      119
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         8144
Inode blocks per group:   509
Flex block group size:    16
Filesystem created:       Tue Dec 10 18:46:46 2019
Last mount time:          n/a
Last write time:          Tue Dec 10 18:46:47 2019
Mount count:              0
Maximum mount count:      -1
Last checked:             Tue Dec 10 18:46:46 2019
Check interval:           0 (<none>)
Lifetime writes:          33 MB
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
```

```
First inode:              11
Inode size:               256
Required extra isize:     28
Desired extra isize:      28
Journal inode:            8
Default directory hash:   half_md4
Directory Hash Seed:      bbea6663-7b8a-47e6-8a10-ac87ca23dfcc
Journal backup:           inode blocks
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Starting dockerd with args '--debug=true':      .

bash-4.3# cd /bootflash/
bash-4.3# ls sw_telemetry-2.0-2.x86_64.rpm
sw_telemetry-2.0-2.x86_64.rpm
bash-4.3# rpm -Uvh sw_telemetry-2.0-2.x86_64.rpm    >> Installing container RPM
Preparing...                ########################################### [100%]
   1:sw_telemetry           ########################################### [100%]
9da9fb5b1b21: Loading layer [==================================================>] 197.4
MB/197.4 MB
855a0ab0dad9: Loading layer [==================================================>] 13.54
MB/13.54 MB
0fbbc3806bd9: Loading layer [==================================================>] 539.1
kB/539.1 kB
17518e2561b2: Loading layer [==================================================>] 81.41
kB/81.41 kB
a00c7083c0c8: Loading layer [==================================================>] 469.5
kB/469.5 kB
b5b4081bee95: Loading layer [==================================================>] 4.608
kB/4.608 kB
83af9595fe45: Loading layer [==================================================>] 1.536
kB/1.536 kB
96b5c176e3d9: Loading layer [==================================================>] 25.73
MB/25.73 MB
Loaded image: sw_telemetry:latest
bash-4.3# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
sw_telemetry        latest              b8f86559a869        36 hours ago        228 MB
bash-4.3# pwd
/bootflash
bash-4.3# cat env.list      >> create env.list file with local loopback ip which is reachable
 to external collector
transport_source_ip=14.13.12.11
bash-4.3# exit
exit
switch# show running-config interface loopback 10

!Command: show running-config interface loopback10
!Running configuration last done at: Tue Dec 10 18:55:59 2019
!Time: Tue Dec 10 18:57:25 2019

version 9.3(3) Bios:version 05.39

interface loopback10
  ip address 14.13.12.11/32
  ip router ospf 10 area 0.0.0.0

switch# ping 5.1.1.1 source-interface loopback 10   >> 5.1.1.1 is my external collector
PING 5.1.1.1 (5.1.1.1): 56 data bytes
64 bytes from 5.1.1.1: icmp_seq=0 ttl=62 time=0.766 ms
64 bytes from 5.1.1.1: icmp_seq=1 ttl=62 time=0.64 ms
64 bytes from 5.1.1.1: icmp_seq=2 ttl=62 time=0.557 ms
```

```
64 bytes from 5.1.1.1: icmp_seq=3 ttl=62 time=0.553 ms
64 bytes from 5.1.1.1: icmp_seq=4 ttl=62 time=0.532 ms

--- 5.1.1.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 0.532/0.609/0.766 ms
switch# run bash sudo su
bash-4.3# cd /bootflash/
bash-4.3# chkconfig --add docker           >> make docker persistent on realod
bash-4.3# docker run -v /var/run/netns:/var/run/netns:ro,rslave --network host --cap-add
SYS_ADMIN -it --env-file /bootflash/env.list -v /bootflash:/bootflash -p 50051:50051 -p
50052:50052 -d -v /tmp/nginx_local/:/tmp/nginx_local/ -v /var/run/:/var/run/ -v
/etc/localtime:/etc/localtime:ro -v /etc:/etc -v /nxos/tmp:/nxos/tmp --restart unless-stopped
 --ipc="host" sw_telemetry_grpc        >> starting container
57a65cbceb599f3750c76de1d88749054d8a396c23c1c34e2bd22743a14dbc3e
bash-4.3# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
            PORTS               NAMES
57a65cbceb59        sw_telemetry        "ip netns exec def..."   2 seconds ago       Up 1
second                                  determined_meninsky
bash-4.3#
bash-4.3# exit
```

About Streaming Statistics Export

The Streaming Statistics Export (SSX) module reads statistics from the ASIC and sends them to a remote server (collector) for analysis. SSX can read a register or (directly accessible) memory of the switch.

The contents of registers and the packet format are specified by an instruction sequence.

SSX Supported Types

Table 1: SSX Supported Types

Types	Description
egress buffer depth	Streams the buffer occupancy per slice
egress queue depth	Instant buffer utilization per output queue
egress queue drop	Tail drops per output queue
ingress queue depth	Instant buffer utilization per input queue
ingress queue drop	Overflow drops per input queue
ECN stats	ECN marked packets per output queue

SSX gRPC Filter Types

The docker container can be configured using gRPC and the filters that are supported for SSX in the following table. The gRPC message number is the enum number of the filters in the proto file.

Table 2: SSX gRPC Filter Types

Filter Types	Filter_types gRPC Message Number
FILTER_TYPE_EGRESS_BUFFER_DEPTH	6
FILTER_TYPE_EGRESS_Q_DEPTH,	3
FILTER_TYPE_EGRESS_Q_DROP,	4
FILTER_TYPE_INGRESS_Q_DEPTH,	2
FILTER_TYPE_INGRESS_Q_DROP,	4
FILTER_TYPE_ECN_STATS	7

SSX Examples

SSX Egress Queue Drop


```
"egressQueueDrop": [
 {
  "QDropBytes": "0",
  "QDropPkts": "0",
  "intf": "Ethernet1/1",
  "queueIndex": 0,
  "ts": "1562008724381260"
 },
 {
  "QDropBytes": "0",
  "QDropPkts": "0",
  "intf": "Ethernet1/1",
  "queueIndex": 1,
  "ts": "1562008724381260"
 },
 {
  "QDropBytes": "0",
  "QDropPkts": "0",
  "intf": "Ethernet1/1",
  "queueIndex": 2,
  "ts": "1562008724381260"
 },
 {
  "QDropBytes": "0",
  "QDropPkts": "0",
  "intf": "Ethernet1/1",
  "queueIndex": 3,
  "ts": "1562008724381260"
 },
 {
```

```
 "QDropBytes": "0",
 "QDropPkts": "0",
 "intf": "Ethernet1/1",
 "queueIndex": 4,
 "ts": "1562008724381260"
},
{
 "QDropBytes": "0",
 "QDropPkts": "0",
 "intf": "Ethernet1/1",
 "queueIndex": 5,
 "ts": "1562008724381260"
},
{
 "QDropBytes": "0",
 "QDropPkts": "0",
 "intf": "Ethernet1/1",
 "queueIndex": 6,
 "ts": "1562008724381260"
},
{
 "QDropBytes": "0",
 "QDropPkts": "0",
 "intf": "Ethernet1/1",
 "queueIndex": 7,
 "ts": "1562008724381260"
},
```


SSX Ingress Queue Occupancy


```
"ingressQueueOccupancy": [
  {
   "QOcc": "0",
   "intf": "Ethernet1/1",
   "queueIndex": 0,
   "ts": "1562640434373522"
  },
  {
   "QOcc": "0",
   "intf": "Ethernet1/1",
   "queueIndex": 1,
   "ts": "1562640434373522"
  },
  {
   "QOcc": "0",
   "intf": "Ethernet1/1",
   "queueIndex": 2,
   "ts": "1562640434373522"
  },
  {
   "QOcc": "0",
   "intf": "Ethernet1/1",
   "queueIndex": 3,
   "ts": "1562640434373522"
  },
  {
   "QOcc": "0",
   "intf": "Ethernet1/1",
   "queueIndex": 4,
   "ts": "1562640434373522"
  },
  {
   "QOcc": "0",
```

```
    "intf": "Ethernet1/1",
    "queueIndex": 5,
    "ts": "1562640434373522"
   },
   {
    "QOcc": "0",
    "intf": "Ethernet1/1",
    "queueIndex": 6,
    "ts": "1562640434373522"
   },
   {
    "QOcc": "0",
    "intf": "Ethernet1/1",
    "queueIndex": 7,
    "ts": "1562640434373522"
   },
```


SSX Egress Queue Occupancy


```
 "egressQueueOccupancy": [
   {
    "QOcc": "0",
    "intf": "Ethernet1/1",
    "queueIndex": 0,
    "ts": "1562640434380646"
   },
   {
    "QOcc": "0",
    "intf": "Ethernet1/1",
    "queueIndex": 1,
    "ts": "1562640434380646"
   },
   {
    "QOcc": "0",
    "intf": "Ethernet1/1",
    "queueIndex": 2,
    "ts": "1562640434380646"
   },
   {
    "QOcc": "0",
    "intf": "Ethernet1/1",
    "queueIndex": 3,
    "ts": "1562640434380646"
   },
   {
    "QOcc": "0",
    "intf": "Ethernet1/1",
    "queueIndex": 4,
    "ts": "1562640434380646"
   },
   {
    "QOcc": "0",
    "intf": "Ethernet1/1",
    "queueIndex": 5,
    "ts": "1562640434380646"
   },
   {
    "QOcc": "0",
    "intf": "Ethernet1/1",
    "queueIndex": 6,
    "ts": "1562640434380646"
   },
   {
```

```
    "QOcc": "0",
    "intf": "Ethernet1/1",
    "queueIndex": 7,
    "ts": "1562640434380646"
   },
```

SSX Ingress Pause Drops

```
"data": {
  "ingressQueuePauseDrop": [
   {
    "QDropPkts": "0",
    "intf": "Ethernet1/1",
    "queueIndex": 4,
    "ts": "1562970319568955"
   },
   {
    "QDropPkts": "0",
    "intf": "Ethernet1/1",
    "queueIndex": 3,
    "ts": "1562970319568955"
   },
```

SSX Egress Mem Occupancy (per slice)

```
 "egressMemOccupancy": [
   {
    "EgBuf": "0",
    "slice": "0",
    "ts": "1573329731594964"
  },
   {
    "EgBuf": "8066",
    "slice": "1",
    "ts": "1573329731594964"
   },
   {
    "EgBuf": "0",
    "slice": "2",
    "ts": "1573329731594964"
   },
   {
    "EgBuf": "0",
    "slice": "3",
    "ts": "1573329731594964"
   }
  ],
```

SSX ECN

```
"ecnStats": [
   {
    "ECNBytes": "0",
    "ECNPkts": "0",
    "intf": "Ethernet1/1",
    "queueIndex": 0,
    "ts": "1562018638423871"
   },
   {
    "ECNBytes": "0",
```

```
 "ECNPkts": "0",
 "intf": "Ethernet1/1",
 "queueIndex": 1,
 "ts": "1562018638423871"
},
{
 "ECNBytes": "0",
 "ECNPkts": "0",
 "intf": "Ethernet1/1",
 "queueIndex": 2,
 "ts": "1562018638423871"
},
{
 "ECNBytes": "0",
 "ECNPkts": "0",
 "intf": "Ethernet1/1",
 "queueIndex": 3,
 "ts": "1562018638423871"
},
{
 "ECNBytes": "0",
 "ECNPkts": "0",
 "intf": "Ethernet1/1",
 "queueIndex": 4,
 "ts": "1562018638423871"
},
{
 "ECNBytes": "0",
 "ECNPkts": "0",
 "intf": "Ethernet1/1",
 "queueIndex": 5,
 "ts": "1562018638423871"
},
{
 "ECNBytes": "0",
 "ECNPkts": "0",
 "intf": "Ethernet1/1",
 "queueIndex": 6,
 "ts": "1562018638423871"
},
{
 "ECNBytes": "0",
 "ECNPkts": "0",
 "intf": "Ethernet1/1",
 "queueIndex": 7,
 "ts": "1562018638423871"
},
```


About Port Counters

The following port counters are supported:

Note This feature provides 32 interface level counter statistics. An ingress/egress tcam region needs to be carved as IPv4 and IPv6 statistics are provided. It is mandatory to carve ingress and egress CNTACL regions.

```
CF-QP2(config)# hardware access-list tcam region ing-cntacl 512
CF-QP2(config)# hardware access-list tcam region egr-cntacl 512
```

Table 3: Supported Port Counters

Port Counter Exports	Description
Port-in/out stats	Port input and output stats in pkts/bytes
Port-in/out Discard stats	Port input and output discard stats in pkts/bytes
Port-in/out Error stats	Port input and output error stats in pkts/bytes
Port-in/out IPv4 stats	Port input and output IPV4 stats in pkts/bytes
Port-in/out IPv6 stats	Port input and output IPV6 stats in pkts/bytes
Port-QoS Group stats	Per port, per queue egress queue stats in pkts/bytes
Port-PFC stats (TX/RX)	Per port, per queue PFC stats (TX/RX)

Port Counter Exports

Table 4: Port Counter gRPC Filter Types

Filter Type	Filter_types gRPC Message Number
FILTER_TYPE_PORT_COUNTERS	1

Port Counter Sample Output

The following is a sample of port counter output.


```
"counters": [
  {
   "interfaceName": "Ethernet1/1",
   "portIn": "21900548",
   "portInDiscard": "0",
   "portInError": "0",
   "portInIpv4": "5468000",
   "portInIpv4Rate": "0",
   "portInIpv6": "5468094",
   "portInIpv6Rate": "0",
   "portOut": "22919963",
   "portOutDiscard": "0",
   "portOutError": "0",
   "portOutIpv4": "5468000",
   "portOutIpv4Rate": "0",
   "portOutIpv6": "5468000",
   "portOutIpv6Rate": "0",
   "portPacketIn": "109722",
   "portPacketInIpv4": "27340",
   "portPacketInIpv4Rate": "0",
   "portPacketInIpv6": "27341",
   "portPacketInIpv6Rate": "0",
   "portPacketOut": "112917",
   "portPacketOutIpv4": "27340",
```

```
 "portPacketOutIpv4Rate": "0",
 "portPacketOutIpv6": "27340",
 "portPacketOutIpv6Rate": "0",
 "portQosGroup0MulticastTxBytes": "0",
 "portQosGroup0MulticastTxPkts": "0",
 "portQosGroup0UnicastTxBytes": "440800",
 "portQosGroup0UnicastTxPkts": "2204",
 "portQosGroup1MulticastTxBytes": "0",
 "portQosGroup1MulticastTxPkts": "0",
 "portQosGroup1UnicastTxBytes": "21231200",
 "portQosGroup1UnicastTxPkts": "106156",
 "portQosGroup2MulticastTxBytes": "0",
 "portQosGroup2MulticastTxPkts": "0",
 "portQosGroup2UnicastTxBytes": "0",
 "portQosGroup2UnicastTxPkts": "0",
 "portQosGroup3MulticastTxBytes": "0",
 "portQosGroup3MulticastTxPkts": "0",
 "portQosGroup3UnicastTxBytes": "0",
 "portQosGroup3UnicastTxPkts": "0",
 "portQosGroup4MulticastTxBytes": "0",
 "portQosGroup4MulticastTxPkts": "0",
 "portQosGroup4UnicastTxBytes": "0",
 "portQosGroup4UnicastTxPkts": "0",
 "portQosGroup5MulticastTxBytes": "0",
 "portQosGroup5MulticastTxPkts": "0",
 "portQosGroup5UnicastTxBytes": "0",
 "portQosGroup5UnicastTxPkts": "0",
 "portQosGroup6MulticastTxBytes": "0",
 "portQosGroup6MulticastTxPkts": "0",
 "portQosGroup6UnicastTxBytes": "0",
 "portQosGroup6UnicastTxPkts": "0",
 "portQosGroup7MulticastTxBytes": "0",
 "portQosGroup7MulticastTxPkts": "0",
 "portQosGroup7UnicastTxBytes": "0",
 "portRx0Pfc": "0",
 "portRx1Pfc": "0",
 "portRx2Pfc": "0",
 "portRx3Pfc": "0",
 "portRx4Pfc": "0",
 "portRx5Pfc": "0",
 "portRx6Pfc": "0",
 "portRx7Pfc": "0",
 "portTx0Pfc": "0",
 "portTx1Pfc": "0",
 "portTx2Pfc": "0",
 "portTx3Pfc": "0",
 "portTx4Pfc": "0",
 "portTx5Pfc": "0",
 "portTx6Pfc": "0",
 "portTx7Pfc": "0",
 "timestamp": "1561055719695692554"
},
```


Guidelines and Limitations for Buffer Drop Capture and Buffer Latency Capture

The following are the Guidelines and Limitations for Buffer Drop Capture (BDC) and Buffer Latency Capture (BLC).

- These features are supported on the following platforms:
  - Cisco Nexus 3408-S
  - Cisco Nexus 3432D-S
- Flow of Interest (FoI): You must apply IPv4 and IPv6 ACLs. If you add only an IPv4 ACL, an explicit deny for IPv6 is configured.
- FoI: Use the same ACL for BDC, BLC, and IPT.
- BDC does not work on Inno Block 4 and Inno Block 5.
- BDC does not work on no-drop queue.
- ODM merge does not work for BDC/BLC FoI. By default, statistic-per-entry is present for these ACLs.
- The hw-telemetry table does not have enough match bits available for ether type (16 bits) to differentiate between IPv4 and IPv6.
- If the original data traffic and SPAN copy of that traffic is going out on the same IB and if the original data traffic is dropped on a queue where BDC is enabled, you will not see any BDC packets for this traffic.
- If the BLC port of interest is also configured as SPAN TX source (egress mirror enable), only SPAN destination will receive a copy and the BLC copy is not created. Hence, BLC does not work on this port.
- We support 5000 PPS for BDC/BLC together.
- You need to carve the TCAM before using FoI. An error is not displayed when applying FoI without TCAM carving.

Configuring Software Telemetry

Configure software telemetry to support streaming data to the docker container running on the switch. Software telemetry must run on same VRF as the container. For example, if the docker container is running on the default VRF, software telemetry must be configured with the **use-vrf default** command for streaming the data.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**feature telemetry** **Example:** `switch(config)# feature telemetry`	Enter configuration mode for streaming telemetry.

	Command or Action	Purpose
Step 3	**feature hardware-telemetry** **Example:** `switch(config)# feature hardware-telemetry`	Enter configuration mode for streaming telemetry.
Step 4	**telemetry** **Example:** `switch(config)# telemetry`	Enter configuration mode for streaming telemetry.
Step 5	**destination-profile** **Example:** `switch(config-telemetry)# destination-profile`	Enter the **destination-profile** command to specify the default destination profile.
Step 6	**use-vrf default** **Example:** `switch(config-tm-dest-profile)# use-vrf default`	To specify the destination VRF.

FTE drop and Latency Capture

Starting with the Cisco NX-OS 10.1(2) release, the flow table (FT) from the ASIC can generate notifications or events when certain conditions are detected in the packets of a flow. These detected events are first collected in a first in, first out (FIFO) method within the flow table. Once a predetermined threshold is reached, these events are exported to the container.

About Flow Table Events (FTE)

The flow table (FT) from the ASIC can generate notifications or events when certain conditions are detected in the packets of a flow. These detected events are first collected in a first in, first out (FIFO) method within the flow table. Once a predetermined threshold is reached, these events are exported as a UDP payload to the analyzer or collector.

An event packet typically includes information such as the following about the flow:

- Flow tuple
- Event reason
- Drop info (for example, acl_drop, policer_drop, policy_drop, buffer_drop, ids_drop, forward_drop)
- TCP-related fields (for example, flags, seq/ack num, and so on)
- Interface info (for example, srcid, oport, oclass, and so on)
- L2 or L4 payload length

Events are asynchronous and are packet-driven. The event export can therefore overwhelm the collectors if they are not throttled appropriately. The Cisco NX-OS 9.2(2t) release supports a per-event counter to limit the event-generated traffic, where the supported event types include the following:

- ACL drops — Generate an event if ACL drops are seen in the system.
- Buffer drops — Generate an event if buffer drops are seen in the system.
- Forward drops — Generate an event if forwarding drops are seen in the system.
- Latency — Generate an event if the latency of a packet exceeds the configured threshold value (in milli/micro sec) in the system. This is a system-wide event.

Guidelines and Limitations

- The flow table events (FTE) feature is supported on the following Cisco Nexus 3400-S Series switches:
  - Cisco Nexus 93108TC-FX
  - Cisco Nexus 93180YC-FX
  - Cisco Nexus 9336C-FX2
  - Cisco Nexus 93240YC-FX2
  - Cisco Nexus 93240YC-FX2Z
  - Cisco Nexus 9348GC-FXP
- Feature NetFlow and FTE cannot coexist on the same switch.

Configuring FTE

Configuring flow table events (FTE) on a switch requires:

- Enabling the FTE feature
- Configuring the FTE exporter
- Configuring the FTE record
- Configuring the FTE event
- Configuring the FTE monitor
- Creating the global configuration for FTE for the attached monitor profile

Enabling the FTE Feature

```
switch# conf
switch(config)# feature hardware-telemetry
switch(config)#
switch(config)# hardware-telemetry fte
switch(config-fte)#
```

Configuring the FTE Exporter

```
switch(config-fte)# fte exporter exp
```

Set the destination IP address of the collector:

```
switch(config-fte-exporter)# destination 100.1.1.2
```

Set a UDP source-port number:

```
switch(config-fte-exporter)# transport udp source-port 555
```

Set a source IP address to be used on the exporter FTE packet:

```
switch(config-fte-exporter)# source 1.1.1.1
```

Set the exporter-id:

```
switch(config-fte-exporter)# exporter-id 1
switch(config-fte-exporter)#
switch(config-fte-exporter)# exit
switch(config-fte)#
```

Configuring the FTE Record

```
switch(config-fte)# fte record rec
```

You must configure at least one of the following match parameters for FTE records:

Command	Purpose
match ipv4 {destination address | source address} Example: `switch(config-fte-record)# match ipv4 source address`	Specifies the IPv4 source or destination address as a key.
match ipv4 {protocol | tos} Example: `switch(config-fte-record)# match ipv4 protocol`	Specifies the IPv4 protocol or ToS fields as keys. **Note** The **match transport destination-port** and **match ip protocol** commands are required to export Layer-4 port data. The data is collected and displayed in the output of the **show hardware flow ip** command but is not collected and exported until you configure both commands.
match ipv4 transport {destination-port | source-port}	Specifies the IPv4 transport source or destination port as a key.

Command	Purpose
Example: `switch(config-fte-record)# match ipv4 transport source-port`	**Note** The **match transport destination-port** and **match ip protocol** commands are required to export Layer-4 port data. The data is collected and displayed in the output of the **show hardware flow ip** command but is not collected and exported until you configure both commands.
match ipv6 {destination address | source address | flow-label} Example: `switch(config-fte-record)# match ipv6 source address`	Specifies the IPv6 key.
match ipv6 {protocol | tos} Example: `switch(config-fte-record)# match ipv6 protocol`	Specifies the IPv6 protocol or ToS fields as keys. **Note** The **match transport destination-port** and **match ip protocol** commands are required to export Layer-4 port data. The data is collected and displayed in the output of the **show hardware flow ip** command but is not collected and exported until you configure both commands.
match ipv6 transport {destination-port | source-port} Example: `switch(config-fte-record)# match ipv6 transport source-port`	Specifies the IPv6 transport source or destination port as a key. **Note** The **match transport destination-port** and **match ip protocol** commands are required to export Layer-4 port data. The data is collected and displayed in the output of the **show hardware flow ip** command but is not collected and exported until you configure both commands.
match datalink {mac source-address | mac destination-address | ethertype} Example: `switch(config-fte-record)# match datalink ethertype`	Specifies the Layer-2 attribute as a key.

Configuring the FTE Event

```
switch(config-fte-event)# group drop-events
switch(config-fte-event-drop-events)#
```

Capture buffer drop events:

```
switch(config-fte-event-drop-events)# capture buffer-drops
```

Capture ACL drop events:

```
switch(config-fte-event-drop-events)# capture acl-drops
```

Capture forwarding drop events:

```
switch(config-fte-event-drop-events)# capture fwd-drops
```

Set the number of events per FTE packet:

```
switch(config-fte-event-drop-events)# flow-count 300
```

Capture latency exceeding a threshold value in micro/milli sec. The latency here is system-wide. TTAG must be enabled for correct functionality.

```
switch(config-fte-event-drop-events)# group latency-events
switch(config-fte-event-latency-events)# capture latency exceeding-threshold 100 micro-sec
```

Set the number of events per FTE packet:

```
switch(config-fte-event-latency-events)# flow-count 300
switch(config-fte-event-latency-events)#
switch(config-fte-event-latency-events)# exit
```

Configuring the FTE Monitor

```
switch(config-fte-event)# fte monitor mon
```

Attach a record profile to a monitor:

```
switch(config-fte-monitor)# record rec
```

Attach an exporter profile to a monitor:

```
switch(config-fte-monitor)# exporter exp
```

Attach an event profile to a monitor:

```
switch(config-fte-monitor)# event eve1
switch(config-fte-monitor)#
```

Configuring the System Command

Configure the system for FTE for the selected Monitor Profile:

```
switch(config-fte)# fte system monitor mon
```

FTE show Commands

show fte record

```
switch# show fte record
```

Output similar to the following appears, where the **No. of users** field displays the number of monitors referencing to this record:

```
FTE record rec_1:
No. of users: 1
Fields:
match ipv4 source address
match ipv4 destination address
match ipv4 protocol
match ipv4 tos
match ipv4 transport source-port
match ipv4 transport destination-port
match ipv6 protocol
match ipv6 tos
match ipv6 transport source-port
match ipv6 transport destination-port
match ipv6 source address
match ipv6 destination address
match datalink mac source-address
match datalink mac destination-address
match datalink ethertype
```

show fte exporter

```
switch# show fte exporter
```

Output similar to the following appears:

```
FTE exporter exp_1:
Destination: 8.1.1.1
VRF: default (0)
Source UDP Port 777
Source IP 20.21.22.23
Exporter-id 666
```

show fte event

```
switch# show fte event
```

Output similar to the following appears, where the **No. of users** field displays the number of monitors referencing to this event:

```
FTE event eve_1:
No. of users: 1
Fields:
capture buffer-drops
capture fwd-drops
capture acl-drops
capture latency exceeding-threshold 1 micro-sec
```

show fte monitor

```
switch# show fte monitor
```

Output similar to the following appears, where the **No. of users** field displays the number of monitors referencing to this event:

```
FTE Monitor mon_1:
Use count: 1
FTE Record: rec_1
FTE Event: eve_1
FTE Exporter: exp_1
```

show flow table (FT) entries

```
switch# show system internal flow nf records
```

Output similar to the following appears:

```
Asic: 1 Slice: 1
=====================
-----------------------------------------------------------------IPV4 FLOW CACHE
-----------------------------------------------------------------
Index STEP DTEP PktDispTenant SIP DIP SPORT DPORT PROTO COS Bytes PktsF-TstmpL-Tstmp
fds pdid d r c p
-------------------------------------------------------------------------------------------------------------------
0326 00000 00109 0 0 0 0 0 0 1 0 004129 0x0e010113 0x0d010102 0000 0000 0061 000 000018685500
 000000012457 0x000000000000 0x00000000008d
0352 00000 00045 0 0 0 0 0 0 1 0 004127 0x0d010110 0x0c010102 0099 0100 0006 000 000007984600
 000000079847 0x000000000000 0x00000000008f
0397 00000 00109 0 0 0 0 0 0 1 0 004129 0x0e010114 0x0d010102 0000 0000 0061 000 000019311000
 000000012875 0x000000000000 0x000000000092
0588 00000 00045 0 0 0 0 0 0 1 0 004127 0x0d01010c 0x0c010102 0099 0100 0006 000 000008846500
 000000088467 0x000000000000 0x00000000009e
0618 00000 00109 0 0 0 0 0 0 1 0 004129 0x0e01010f 0x0d010102 0000 0000 0061 000 000021210000
 000000014141 0x000000000000 0x0000000000a0
0647 00000 00045 0 0 0 0 0 0 1 0 004127 0x0d01010b 0x0c010102 0099 0100 0006 000 000009071600
 000000090718 0x000000000000 0x0000000000a3
0673 00000 00109 0 0 0 0 0 0 1 0 004129 0x0e010108 0x0d010102 0000 0000 0061 000 000021712500
 000000014476 0x000000000000 0x0000000000a4
-------------------------------------------------------------------------------------------------------------------
```

show run hardware-telemetry

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

Output similar to the following appears:

```
hardware-telemetry fte
fte exporter exp
  destination 100.1.1.2
  transport udp source-port 555
  source 1.1.1.1
  exporter-id 1
fte record rec
  match ipv4 source address
  match ipv4 destination address
  match ipv4 protocol
  match ipv4 tos
```

```
  match ipv4 transport source-port
  match ipv4 transport destination-port
  match ipv6 protocol
  match ipv6 tos
  match ipv6 transport source-port
  match ipv6 transport destination-port
  match ipv6 source address
  match ipv6 destination address
  match ipv6 flow-label
  match datalink mac source-address
  match datalink mac destination-address
  match datalink ether type
fte event eve1
  group drop-events
    capture buffer-drops
    capture acl-drops
    capture fwd-drops
    flow-count 32768
  group latency-events
    capture latency exceeding-threshold 100 micro-sec
    flow-count 32768
fte monitor mon
  record rec
  exporter exp
  event eve1
fte system monitor mon
```

FTE Drop and Latency Filter Types

Table 5: FTE Drop and Latency Filter Types

Filter Types	Filter_types gRPC Message Number
FTE_FILTER_TYPES	10

Innovium Path Telemetry

About Innovium Path Telemetry

Innovium Path Telemetry (IPT) is a telemetry feature which makes a truncated copy of the original packet and adds IPT metadata at each node that has IPT enabled.

- Source: At this node a copy of the original packet is made and IPT probe-marker, base header and hop information are added after the original packets L2-L4 headers.
- Transit (1 or more): At this node, IPT packets are identified based on the probe-marker value and another hop information is added after the last node hop information. There can be multiple transit nodes for an IPT packet.
- Sink: This is the last node, and here hop information is added to all IPT packets based on the probe-marker value. Collector headers are added at the beginning of the packet and sent to the collector.

Guidelines and Limitations for Innovium Path Telemetry

Innovium Path Telemetry (IPT) has the following configuration guidelines and limitations:

- On the sink node, IPT packets are accounted for on the sink interface queue besides being accounted for on the collector interface queue.
- Egress remarking of Innovium Path Telemetry (IPT) packets is not supported. Egress remarking of original packets should not be used with IPT since IPT data cannot be guaranteed in this case.
- SPAN and ERSPAN cannot monitor IPT packets.
- Innovium Path Telemetry is supported on Cisco Nexus 3408-S and 3432D-S switches.
- For Innovium Path Telemetry to work correctly, TCAM carving for ingress and egress is required.
- Only unicast, non-fragmented TCP and UDP packets are supported.
- Flow of Interest (FoI) is supported.
- The FoI ACL is shared with buffer latency and buffer drop features.
- Deny ACEs within the ACL are not supported.
- A maximum of one monitor can be applied.
- IPT/buffer drop and buffer latency share same ACL. If buffer drop and/or buffer latency have FoI with action telemetry_path, even though FoI is not applied to IPT, filter action happens for IPT as well.

Configuring Flow of Interest ACL for IPT

This procedure describes how to configure a flow of interest.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**ip access-list** *list-name* **Example:** `switch(config)# ip access-list ipt-foi-v4`	Configure the access list.
Step 3	*seq-number* **{permit | deny}** *protocol {source-ip-prefix | source-ip-mask} {destination-ip-prefix | destination-ip-mask}* **telemetry_path** **Example:** `switch(config-acl)# 10 permit ip 10.1.1.1/32 11.1.1.1/32 telemetry_path`	Specify packets to forward. IPT enabled. The value of *seq-number* is from 1 to 4294967295.

	Command or Action	Purpose
	`switch(config-acl)# 10 permit ipv6 10:1:1::/64 20:1:1::/64 telemetry_path`	

Step 4	*seq-number* **{permit | deny}** *protocol {source-ip-prefix | source-ip-mask} {destination-ip-prefix | destination-ip-mask}* **telemetry_path**
	Example:
	`switch(config-acl)# 20 permit ip 30.1.1.2/32 30.1.1.2/32 telemetry_path`
	`switch(config-acl)# 20 permit ipv6 30:1:1::/64 30:1:1::/64 telemetry_path`
		Specify packets to forward. IPT enabled. The value of *seq-number* is from 1 to 4294967295.

Configuring TCAM for Innovium Path Telemetry

Configuring Ingress TCAM Carving

This procedure configures a TCAM region in support of the Flow of Interest (FOI) option.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**hardware access-list tcam region hw-telemetry** *size* **Example:** `switch(config)# hardware access-list tcam region hw-telemetry 128`	Configure the TCAM region for the hardware telemetry size. The values of *size* are 128 or 256.
Step 3	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the running configuration to the startup configuration.
Step 4	**reload** **Example:** `switch(config)# reload`	Reboots the switch.

Configuring Egress TCAM Carving

This procedure describes how to configure egress TCAM carving.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**hardware access-list tcam region e-hw-telemetry** *size* **Example:** `switch(config)# hardware access-list tcam region e-hw-telemetry 128`	Configure TCAM region egress hardware telemetry size. The values of *size* is 128.
Step 3	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the running configuration to the startup configuration.
Step 4	**reload** **Example:** `switch(config)# reload`	Reboots the switch.

Configuring the Source Node

This procedure configures the IPT probe marker, base header, and hop information.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**feature hardware-telemetry** **Example:** `switch(config)# feature hardware-telemetry`	Enable hardware telemetry.
Step 3	**hardware-telemetry int-clone-md** **Example:** `switch(config)# hardware-telemetry int-clone-md`	Enable INT Clone MD configuration.
Step 4	**int-clone-md probe-marker** *pm_value*int-clone-md probe-marker **Example:**	Configure a probe marker value. For *pm_value*, the range is from 1 - 281474976710655.

	Command or Action	Purpose
	`switch(config-int-clone-md)#` **`int-clone-md probe-marker 0x4d2`**	
Step 5	**int-clone-md source record** *src_rec* **Example:** `switch(config-int-clone-md)#` **`int-clone-md source record src_rec1`**	Define a source record.
Step 6	**interface ethernet** *slot/chassis* **Example:** `switch(config-int-clone-md-source-record)#` **`interface ethernet 1/23`**	Configure interfaces.
Step 7	**exit** **Example:** `switch(config-int-clone-md-source-record)#` **`exit`**	Exit current configuration mode.
Step 8	**int-clone-md source monitor** *src_mon* **Example:** `switch(config-int-clone-md)#` **`int-clone-md source monitor src_mon1`**	Define a source record.
Step 9	**record** *src_rec* **Example:** `switch(config-int-clone-md-source-monitor)#` **`record src_rec1`**	Add a record.
Step 10	**filter ip access-list** *ipt* **Example:** `switch(config-int-clone-md-source-monitor)#` **`filter ip access-list ipt`**	Applies the previously configured FoI ACL.
Step 11	**sampling rate** sr-value **Example:** `switch(config-int-clone-md-source-monitor)#` **`sampling rate 1`**	Specify sampling rate for INT Clone MD.
Step 12	**filter ip access-list** *access-list-name* **Example:** `switch(config-buffer-drop-monitor)#` **`filter ip access-list ipt`**	Configure the IPv4 filter.
Step 13	**filter ipv6 access-list** *access-list-name* **Example:** `switch(config-buffer-drop-monitor)#` **`filter ipv6 access-list acl2`**	Configure the IPv6 filter.

	Command or Action	Purpose
Step 14	**exit** **Example:** `switch(config-int-clone-md-source-monitor)# exit`	Exit current configuration mode.
Step 15	**int-clone-md system source monitor** *src_mon* **Example:** `switch(config-int-clone-md)# int-clone-md system source monitor src_mon1`	Specify source monitor to be applied.

Configuring the Transit Node

This procedure configures the **probe-marker** value.

Before you begin

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**feature hardware-telemetry** **Example:** `switch(config)# feature hardware-telemetry`	Enable hardware telemetry.
Step 3	**hardware-telemetry int-clone-md** **Example:** `switch(config)# hardware-telemetry int-clone-md`	Enable INT Clone MD configuration.
Step 4	**int-clone-md probe-marker** *pm_value* **Example:** `switch(config-int-clone-md)# int-clone-md probe-marker 0x4d2`	Configure a probe marker value. The range for *pm_value* is from 1 - 281474976710655.

Configuring the Sink Node

This procedure configures the hop information to all IPT packets based on the **probe-marker** value.

Before you begin

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**feature hardware-telemetry** **Example:** `switch(config)# feature hardware-telemetry`	Enable hardware telemetry.
Step 3	**hardware-telemetry int-clone-md** **Example:** `switch(config)# hardware-telemetry int-clone-md`	Enable INT Clone MD configuration.
Step 4	**int-clone-md probe-marker** *pm_value* **Example:** `switch(config-int-clone-md)# int-clone-md probe-marker 0x4d2`	Configure a probe marker value. The range of *pm_value* is from 1 - 281474976710655.
Step 5	**int-clone-md sink collector** *sink_col* **Example:** `switch(config-int-clone-md)# int-clone-md sink collector sink_col`	Configure sink collector.
Step 6	**source ipv4** *ipaddr* **Example:** `switch(config-int-clone-md-sink-collector)# source ipv4 192.0.2.1`	Configure IP address.
Step 7	**dscp** *dscp-val* **Example:** `switch(config-int-clone-md-sink-collector)# dscp 33`	Configure DSCP value. The range of *dscp-val* is 0-2147483647.
Step 8	**ttl** *ttl-val* **Example:** `switch(config-int-clone-md-sink-collector)# ttl 60`	Configure TTL value. The range of *ttl-val* is 1-255.
Step 9	**exit** **Example:** `switch(config-int-clone-md-sink-collector)# exit`	Exit current configuration mode.

	Command or Action	Purpose
Step 10	**int-clone-md sink record** *sink_rec* **Example:** `switch(config-int-clone-md)# int-clone-md sink record sink_rec1`	Define a sink record.
Step 11	**interface ethernet** *slot/chassis* **Example:** `switch(config-int-clone-md-sink-record)# interface Ethernet1/23`	Configure interface.
Step 12	**exit** **Example:** `switch(config-int-clone-md-sink-collector)# exit`	Exit current configuration mode.
Step 13	**int-clone-md sink monitor** *sink_mon* **Example:** `switch(config-int-clone-md)# int-clone-md sink monitor sink_mon1`	Define a sink monitor.
Step 14	**collector** *sink_col* **Example:** `switch(config-int-clone-md-sink-monitor)# collector sink_col1`	Define a sink monitor.
Step 15	**record** *sink_rec* **Example:** `switch(config-int-clone-md-sink-monitor)# record sink_rec1`	Add a record.
Step 16	**exit** **Example:** `switch(config-int-clone-md-sink-collector)# exit`	Exit current configuration mode.
Step 17	**int-clone-md system sink monitor** *sink_mon* **Example:** `switch(config-int-clone-md)# int-clone-md system sink monitor sink_mon1`	Sink monitor to be applied.

Verifying Innovium Path Telemetry

To display the Innovium Path Telemetry configuration information enter the following command:

Command	Purpose

show ipt details
`---------------------------------`
`  IPT Enabled ***** System Details`
`---------------------------------`
`Probe Marker   : 1234`
 	
`-------------------------------------`
`Source Monitor : src_mon(1) details`
`-------------------------------------`
`In use (applied to system): YES`
`V4 acl: ipt`
`V6 acl: iptv6`
`Sampling Rate: 1`
`Record Attached: src_rec`
 	
`-------------------------------------`
`Source Record : src_rec(1) details`
`-------------------------------------`
`Total interfaces under record:  1`
`Ethernet1/23`
 	
`-------------------------------------`
	Displays Innovium Path Telemetry details.

Command	Purpose

show hardware access-list tcam region
`IPV4 PACL [ifacl] size = 512`
`IPV6 PACL [ipv6-ifacl] size = 0`
`MAC PACL [mac-ifacl] size = 0`
`IPV4 VACL [vacl] size = 0`
`IPV6 VACL [ipv6-vacl] size = 0`
`MAC VACL [mac-vacl] size = 0`
`IPV4 RACL [racl] size = 256`
`IPV6 RACL [ipv6-racl] size = 0`
`Egress IPV4 RACL [e-racl] size = 0`
`Egress IPV6 RACL [e-ipv6-racl] size = 0`
`SPAN [span] size = 0`
`Ingress L2 QOS [ing-l2-qos] size = 0`
`Ingress L3/VLAN QOS [ing-l3-vlan-qos] size = 128`
`Ingress SUP [ing-sup] size = 256`
`Egress L2 QOS [egr-l2-qos] size = 0`
`Egress L3/VLAN QOS [egr-l3-vlan-qos] size = 128`
`Ingress PACL v4 & v6 [ifacl-all] size = 0`
`Ingress RACL v4 & v6 [racl-all] size = 0`
`Ingress QoS L2/L3 [ing-l2-l3-qos] size = 0`
`HW Telemetry [hw-telemetry] size = 128`
`Egress PACL v4 v6 [e-ifacl-all] size = 0`
`Egress HW Telemetry [e-hw-telemetry] size = 128`
	Displays TCAM carving.

Command	Purpose

show hardware access-list tcam region
`show hardware access-list tcam region`
`                                IPV4 PACL`
`[ifacl] size =  512`
`                           IPV6 PACL`
`[ipv6-ifacl] size =    0`
`                            MAC PACL`
`[mac-ifacl] size =    0`
`                                 IPV4 VACL`
`[vacl] size =    0`
`                           IPV6 VACL`
`[ipv6-vacl] size =    0`
`                             MAC VACL`
`[mac-vacl] size =    0`
`                                 IPV4 RACL`
`[racl] size =  256`
`                           IPV6 RACL`
`[ipv6-racl] size =    0`
`                         Egress IPV4 RACL`
`[e-racl] size =    0`
`                      Egress IPV6 RACL`
`[e-ipv6-racl] size =    0`
`                                       SPAN`
`[span] size =    0`
`                        Ingress L2 QOS`
`[ing-l2-qos] size =    0`
`              Ingress L3/VLAN QOS`
`[ing-l3-vlan-qos] size =  128`
`                           Ingress SUP`
`[ing-sup] size =  256`
`                         Egress L2 QOS`
`[egr-l2-qos] size =    0`
`                Egress L3/VLAN QOS`
`[egr-l3-vlan-qos] size =  128`
`                   Ingress PACL v4 & v6`
`[ifacl-all] size =    0`
`                    Ingress RACL v4 & v6`
`[racl-all] size =    0`
`                  Ingress QoS L2/L3`
`[ing-l2-l3-qos] size =    0`
`                       HW Telemetry`
`[hw-telemetry] size =  128`
`                    Egress PACL v4 v6`
`[e-ifacl-all] size =    0`
	Display egress TCAM carving.

show ip access lists
`show ip access-lists`
 	
`IP access list ipt`
`        10 permit ip 1.1.1.1/32 2.2.2.2/32 telemetry_path`
`        20 permit ip 3.3.3.3/32 4.4.4.4/32 telemetry_queue`
	Displays IPv4 access lists.

Command	Purpose

show ipv6 access-lists
`show ipv6 access-lists`
 	
`IPv6 access list iptv6`
`10 permit ipv6 1:1::1:1/128 2:2::2:2/128 telemetry_path`
`20 permit ipv6 3:3::3:3/128 4:4::4:4/128 telemetry_queue`
	Displays IPv6 access lists.