Packet Timestamping

This chapter contains the following sections:

• About Packet Timestamping, on page 1
• Guidelines and Limitations, on page 3

About Packet Timestamping

Packet timestamping enables precise, scalable traffic monitoring. It helps to detect congestion spots on routers or devices in the network.

Every participating switch can add one or more timestamp shims, and the decision is based on local configuration.

Timestamping consists of:

• Per-port or Per-flow timestamping

• Insert up to two timestamps at the end of the frame (pre-enqueue and post-dequeue)

• Convey a notion of source identifier that accompany every timestamp record (path topology)

The following figure provides a graphical representation of packet timestamping.
An advantage of per-port timestamping is that you can save IFP entries and all packets get timestamped. Each port can be configured to enable timestamping in this way:

- Packets entering ports with timestamping enabled get an ingress timestamp.
- Packets that leave timestamp-enabled ports get an egress timestamp.

To enable per-port timestamping on ingress and egress of port ethernet1/1 using the CLI in NX-OS:

```
configure terminal
interface ethernet1/1
timestamp ingress id source_id egress id source_id
```

To disable per-port timestamping on port e1/1 using the CLI in NX-OS:

```
no timestamp
```

**Per-Flow Timestamping**

Per-flow granularity is achieved in timestamping by defining new action fields for the IFP policy table.

**Captured Data**

Following is an example of captured data without packet timestamping:

```
0000 0000 01 00 00 01 00 10 94 00 00 02 08 00 45 00
0010 00 52 d2 ee 00 00 ff fd 66 67 c0 55 01 02 c0 00
```
Following is an example of captured date with packet timestamping in ingress and egress enabled:

\[\text{0000} \quad 00 \quad 01 \quad 00 \quad 01 \quad 00 \quad 01 \quad 01 \quad 00 \quad 01 \quad 00 \quad 10 \quad 94 \quad 00 \quad 02 \quad 08 \quad 00 \quad 45 \quad 00\]
\[\text{0010} \quad 00 \quad 52 \quad 81 \quad ef \quad 00 \quad 00 \quad ff \quad fd \quad b7 \quad 66 \quad c0 \quad 55 \quad 01 \quad 02 \quad c0 \quad 00\]
\[\text{0020} \quad 00 \quad 01 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00\]
\[\text{0030} \quad 00 \quad 01 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00\]
\[\text{0040} \quad 00 \quad 01 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00 \quad 00\]
\[\text{0050} \quad 54 \quad df \quad 93 \quad d4 \quad cd \quad 46 \quad 97 \quad 80 \quad 2c \quad 89 \quad f1 \quad 0e \quad 50 \quad 33 \quad 0c \quad d9\]
\[\text{0060} \quad 9b \quad e9 \quad fc \quad 50 \quad 00 \quad 00 \quad b1 \quad cc \quad e0 \quad 0e \quad 00 \quad 01 \quad 04 \quad 00 \quad 00 \quad 00\]
\[\text{0070} \quad 00 \quad 00 \quad 00 \quad 00 \quad b1 \quad cc \quad e2 \quad 24 \quad 00 \quad 00 \quad 01 \quad 77\]

**Guidelines and Limitations**

Following are the guidelines and limitations for the timestamping feature:

- For information about supported platforms, see [Supported Platforms for Programmability Features](#).

- Timestamp is not part of the L3 packet. Any checks that assume that the L3+ packet length field represents the total frame length will not be accurate. Systems that need to subject packets to such checks must disable timestamping for the corresponding system or port of flow.

- Header length fields or checksum fields (for example, UDP checksum) will not be updated with the insertion of the timestamp.

- IEEE 802.3 frames (for example, SNAP LLC) are not supported.

- Features that rely on the I2E_CLASSID and HG_CLASSID extended header will not co-exist with packet timestamping.

- No switches across the timestamping path should do pad-stripping or otherwise adjust frame content based on the IP header `payload_len/total_len` field for Ethernet II frames.

- Timestamping is not available for:
  - Mirrored copy
  - SOBMH packets
  - Truncated packets
  - Ingress of HiGig port
  - RCPU