



Hardware Timestamp

Table 1: Feature History Table

Whenever periodic stats are streamed, the collector reads the stats from its internal cache, instead of fetching the stats from the hardware. When the data is read from the sensor paths of Stats manager cache, the rate calculation shows spikes. This behavior is due to the timestamp from the collector that is off by several seconds. Therefore, timestamp of some other collector takes precedence because timestamps of collectors are not in synchronization with the current timestamp. This is observed when there are multiple collectors providing stats updates for the same interface.

The YANG data model for Stats manager `Cisco-IOS-XR-infra-statsd-oper.yang` is enhanced to enable the collector to read periodic stats data from the router using hardware timestamp.

The hardware timestamp is taken into account when a primary collector (for generic or proto stats) provides stats updates from the hardware to the Stats manager. With hardware timestamping in rate computation while streaming periodic stats, the spikes due to the timestamp issue is resolved.

The hardware timestamp is updated only when the collector attempts to read the counters from hardware. Else, the value remains 0. The latest stats can be streamed at a minimum cadence of 10 seconds and periodic stats at a cadence of 30 seconds. The support is available only for physical interfaces and subinterfaces, and bundle interface and subinterfaces.

When there is no traffic flow on protocols for an interface, the hardware timestamp for the protocols is published as 0. This is due to non-synchronized timestamps sent by the collector for protocols in traffic as compared to non-traffic scenarios.

A non-zero value is published for protocols that have stats published by a primary collector for both traffic and non-traffic scenarios.



Note The hardware timestamp is supported only for primary collectors. When the hardware has no update, the timestamp will be same. However generic counters are computed for primary and non-primary collectors. The non-primary collectors show the latest stats, but not the timestamp.

When the counters are cleared for an interface using **clear counters interface** command, all counter-related data including the timestamps for the interface is cleared. After all counter values are cleared and set to 0, the last data time is updated only when there is a request for it from a collector. For example, last data time gets updated from a collector:

```
Router# :Aug 7 09:01:08.471 UTC: statsd_manager_1[168]: Updated last data time for ifhandle
0x02000408,
```

```
stats type 2 from collector with node 0x100, JID 250, last data time 1596790868.
INPUT: last 4294967295 updated 1596469986. OUTPUT: last 4294967295 updated 1596469986
```

All other counter values and hardware timestamp are updated when the counters are fetched from the hardware. In this case, all counters including the hardware timestamp is 0:

```
{ "node_id_str": "MGBL_MTB_5504", "subscription_id_str": "app_TEST_200000001",
  "encoding_path": "Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/cache/generic-counters",
  "collection_id": "7848",
  "collection_start_time": "1596790879567",
  "msg_timestamp": "1596790879571", "data_json":
  [{"timestamp": "1596790879570", "keys": [{"interface-name": "FortyGigE0/1/0/11"}],
    "content": {"packets-received": "0", "bytes-received": "0", "packets-sent": "0",
      "bytes-sent": "0", "multicast-packets-received": "0", "broadcast-packets-received": "0",
      "multicast-packets-sent": "0", "broadcast-packets-sent": "0", "output-drops": "0", "output-queue-drops": "0",
      "input-drops": "0", "input-queue-drops": "0", "runt-packets-received": "0", "giant-packets-received": "0",
      "throttled-packets-received": "0", "parity-packets-received": "0", "unknown-protocol-packets-received": "0",
      "input-errors": "0", "crc-errors": "0", "input-overruns": "0", "framing-errors-received": "0", "input-ignored-packets": "0",
      "input-aborts": "0", "output-errors": "0", "output-underruns": "0", "output-buffer-failures": "0", "output-buffers-swapped-out": "0",
      "applique": "0", "resets": "0", "carrier-transitions": "0", "availability-flag": "0",
      "last-data-time": "1596790868", "hardware-timestamp": "0",
      "seconds-since-last-clear-counters": "15", "last-discontinuity-time": "1596469946", "seconds-since-packet-received": "0",
      "seconds-since-packet-sent": "0"}]}, "collection_end_time": "1596790879571" }
```

- [Target-Defined Mode for Cached Generic Counters Data, on page 2](#)

Target-Defined Mode for Cached Generic Counters Data

Table 2: Feature History Table

Feature Name	Release Information	Description
Target-Defined Mode for Cached Generic Counters Data	Release 7.3.3	<p>This feature streams telemetry data for cached generic counters using a TARGET_DEFINED subscription. This subscription ensures that any change to the cache streams the latest data to the collector as an event-driven telemetry notification.</p> <p>This feature introduces support for the following sensor path:</p> <pre>Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/cache/generic-counters</pre>

Streaming telemetry pushes the subscribed data from the router to one or more collectors. The telemetry infrastructure retrieves the data from the system database when you send a subscription request. Based on the subscription request or the telemetry configuration the cached generic counters data can be retrieved periodically based on the sample-interval. Data, such as interface statistics, is cached and refreshed at certain intervals. The TARGET_DEFINED subscription mode can be used to retrieve data when the cache gets updated, and is not based on a timer.

The application can register as a data producer with the telemetry library and the SysdB paths it supports. One of the data producers, Statsd, uses the library with a TARGET_DEFINED subscription mode. As part of this mode, the producer registers the sensor paths. The statistics infrastructure streams the incremental updates for statsd cache sensor path

```
Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/cache/generic-counters.
```

With this path in the subscription, whenever cache is updated, the statsd application pushes the updates to the

telemetry daemon. The daemon sends these incremental updates to the collector. The cache updates are pushed for physical interfaces, physical subinterfaces, bundle interfaces, and bundle subinterfaces. You can subscribe to the sensor path for the cached generic counters with `TARGET_DEFINED` mode instead of the sensor path for the latest generic counters

(Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/latest/generic-counters) to reduce the system load.

Configure the router to stream telemetry data from cache for generic counters using the following instructions:

Create a `TARGET_DEFINED` subscription mode for cached generic counters using one of the two options:

- **Option 1:** gRPC Network Management Interface (gNMI) subscribe request

```
{
  "name": "SubscribeRequest",
  "subscribe": {
    "prefix": {"origin":
      "Cisco-IOS-XR-infra-statsd-oper"
    },
    "mode": "STREAM", "encoding": "PROTO", "updates_only": "false",
    "subscription": [
      { "path": {"elem": [ {"name": "infra-statistics"},
        {"name": "interfaces"},
        {"name": "interface"},
        {"name": "cache"},
        {"name": "generic-counters"}
      ]}
    ]
  },
  "mode": "TARGET_DEFINED"
}
]
```

- **Option 2:** Model-driven telemetry configuration for non-gNMI requests

```
Router(config)#telemetry model-driven
Router(config-model-driven)#subscription sub1
Router(config-model-driven-subs)#sensor-group-id grpl mode target-defined
Router(config-model-driven-subs)#source-interface Interface1
Router(config-model-driven-subs)#commit
```

After the subscription is triggered, updates to the stats cache are monitored. The statsd application pushes the cached generic counters to the client (collector).

View the number of incremental updates for the sensor path.

```
Router#show telemetry model-driven subscription .*
Fri Nov 12 23:36:27.212 UTC
Subscription: GNMI__16489080148754121540
-----
Collection Groups:
-----
  Id: 1
  Sample Interval:      0 ms    (Incremental Updates)
  Heartbeat Interval:  NA
  Heartbeat always:    False
  Encoding:             gnmi-proto
  Num of collection:   1
  Incremental updates: 12
  Collection time:     Min:      5 ms Max:      5 ms
```

```

Total time:           Min:      6 ms Avg:      6 ms Max:      6 ms
Total Deferred:      1
Total Send Errors:   0
Total Send Drops:    0
Total Other Errors:  0
No data Instances:  0
Last Collection Start:2021-11-12
                    23:34:27.1362538876 +0000
Last Collection End: 2021-11-12   23:34:27.1362545589
                    +0000
Sensor Path:         Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/
                    interface/cache/generic-counters

```

In this example, the incremental updates of 12 indicates that the cache is updated 12 times.

You can also retrieve the detailed operational data about the subscription using the following command. In this example, statsd-target is the subscription name.

```

Router#show telemetry model-driven subscription statsd-target internal
Fri Nov 12 08:51:16.728 UTC
Subscription: statsd-target
-----
State: ACTIVE
Sensor groups:
Id: statsd
Sample Interval: 0 ms (Incremental Updates)
Heartbeat Interval: NA
Sensor Path: Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/cache/
            generic-counters
Sensor Path State: Resolved

Destination Groups:
Group Id: statsd-target
Destination IP: 192.0.2.1
Destination Port: 56000
Encoding: json
Transport: grpc
State: Active
TLS : False
Total bytes sent: 623656
Total packets sent: 13
Last Sent time: 2021-08-16 08:51:15.1304821089 +0000

Collection Groups:
-----
Id: 2
Sample Interval: 0 ms (Incremental Updates)
Heartbeat Interval: NA
Heartbeat always: False
Encoding: json
Num of collection: 1
Incremental updates: 3
Collection time: Min: 94 ms Max: 94 ms
Total time: Min: 100 ms Avg: 100 ms Max: 100 ms
Total Deferred: 0
Total Send Errors: 0
Total Send Drops: 0
Total Other Errors: 0
No data Instances: 0
Last Collection Start:2021-08-16 08:51:04.1293895665 +0000
Last Collection End: 2021-08-16 08:51:04.1293996284 +0000

```

The sample interval of 0 indicates that the data is streamed whenever an event occurs. Here, the event represents the updates to the cache state.

Related Commands:

- **show tech telemetry model-driven**
- **show running-config telemetry model-driven**
- **show telemetry producers trace *producer name* info**
- **show telemetry producers trace *producer name* err**

