Core Components of Model-driven Telemetry Streaming

The core components used in streaming model-driven telemetry data are described in this chapter.

- Session, on page 1
- Sensor Path, on page 2
- Sensor Paths Supported for EDT in NCS 1001, on page 2
- OpenConfig Sensor Paths Supported for MDT in NCS 1001, on page 3
- Sensor Paths Supported in NCS 1004, on page 3
- Subscription, on page 5
- Transport and Encoding, on page 6

Session

A telemetry session can be initiated using:

Dial-in Mode

In a dial-in mode, an MDT receiver dials in to the router, and subscribes dynamically to one or more sensor paths or subscriptions. The router acts as the server and the receiver is the client. The router streams telemetry data through the same session. The dial-in mode of subscriptions is dynamic. This dynamic subscription terminates when the receiver cancels the subscription or when the session terminates.

There are two methods to request sensor-paths in a dynamic subscription:

- **OpenConfig RPC model:** The `subscribe RPC` defined in the model is used to specify sensor-paths and frequency. In this method, the subscription is not associated with an existing configured subscription. A subsequent `cancel` RPC defined in the model removes an existing dynamic subscription.

- **IOS XR MDT RPC:** IOS XR defines RPCs to subscribe and to cancel one or more configured subscriptions. The sensor-paths and frequency are part of the telemetry configuration on the router. A subscription is identified by its configured subscription name in the RPCs.

Dial-out Mode

In a dial-out mode, the router dials out to the receiver. This is the default mode of operation. The router acts as a client and receiver acts as a server. In this mode, sensor-paths and destinations are configured and bound together into one or more subscriptions. The router continually attempts to establish a session with each destination in the subscription, and streams data to the receiver. The dial-out mode of subscriptions is persistent. When a session terminates, the router continually attempts to re-establish a new session with the receiver every 30 seconds.

Sensor Path

The sensor path describes a YANG path or a subset of data definitions in a YANG model with a container. In a YANG model, the sensor path can be specified to end at any level in the container hierarchy.

An MDT-capable device, such as a router, associates the sensor path to the nearest container path in the model. The router encodes and streams the container path within a single telemetry message. A receiver receives data about all the containers and leaf nodes at and below this container path.

The router streams telemetry data for one or more sensor-paths, at the configured frequency (cadence-based streaming) or when the sensor-path content changes (event-based streaming), to one or more receivers through subscribed sessions.

Sensor Paths Supported for EDT in NCS 1001

The following sensor paths are supported for Event-based telemetry in NCS 1001.

EDT Sensor Path	Description
`Cisco-IOS-XR-controller-optics-oper:optics-oper/optics-ports/optics-port/optics-info`	This event is triggered when the configuration changes for optics/ots controller (say shutdown / no shutdown) or when the configuration changes for Transport Admin State (say sec-admin-state maintenance).
`Cisco-IOS-XR-pmengine-oper:performance-management-history/global/periodic/optics-history/optics-port-histories/optics-port-history/optics-second30-history`	This event is triggered when the 30 seconds historical PM is completed. It returns latest bucket for all optics/ots controllers.
`Cisco-IOS-XR-pmengine-oper:performance-management-history/global/periodic/optics-history/optics-port-histories/optics-port-history/optics-minute15-history`	This event is triggered when the 15 minutes historical PM is completed. It returns latest bucket for all optics/ots controllers.
`Cisco-IOS-XR-pmengine-oper:performance-management-history/global/periodic/optics-history/optics-port-histories/optics-port-history/optics-hour24-history`	This event is triggered when the 24 hours historical PM is completed. It returns latest bucket for all optics/ots controllers.

OpenConfig Sensor Paths Supported for MDT in NCS 1001

The following OpenConfig sensor paths are supported for Model-based telemetry in NCS 1001.

MDT Sensor Path	Description
`openconfig-optical-amplifier:optical-amplifier/amplifiers/amplifier`	Sensor path related to EDFA objects (ots controllers)
`openconfig-transport-line-protectionaps/aps-modules/aps-module`	Sensor path related to PSM objects (ots controllers)
`openconfig-channel-monitor:channel-monitors/channel-monitor/channels`	Sensor path related to EDFA objects (ots-och controllers and spectrum information)

Sensor Paths Supported in NCS 1004

The following sensor paths are supported in NCS 1004.

Table 1:

Model Type	Sensor Path	Description
Native	Cisco-IOS-XR-show-fpd-loc-ng-oper:show-fpd/hw-module-fpd	Provides the details of the FPGA versions of various hardware components and the packaged FPGAs with the ISO such as, BP_FPGA, XGE_FLASH.
Native	Cisco-IOS-XR-mediasvr-linux-oper:media-svr/all	Provides details of available space and occupied space in the various directory structures.
Native	Cisco-IOS-XR-alarmgr-server-oper:alarms/brief/brief-system/active	Provides the list of all active system alarms on the node.
Native	Cisco-IOS-XR-alarmgr-server-oper:alarms/brief/brief-system/suppressed	Provides the list of all suppressed system alarms on the node.
Native	Cisco-IOS-XR-alarmgr-server-oper:alarms/brief/brief-system/conditions	Provides the list of all conditional system alarms on the node.

Model Type	Sensor Path	Description
Native	Cisco-IOS-XR-controller-optics-oper:optics-oper/optics-ports	Provides the details of all the trunk or client ports of optics controller such as Baud rate, TX-RX power admin state, and LED state
Native	Cisco-IOS-XR-pmengine-oper:performance-management/otu/otu-ports/otu-port/otu-current/otu-second30/otu-second30fecs	Provides the details of OTU FEC PM counters for 30 second bucket such as, OSNR, PDL, PSR.
Native	Cisco-IOS-XR-pmengine-oper:performance-management/otu/otu-ports/otu-port/otu-current/otu-second30/otu-second30otns	Provides the details of OTU OTN PM counters for 30 second bucket such as, BBER-FE, FC-FE.
Native	Cisco-IOS-XR-pmengine-oper:performance-management/optics/optics-ports/optics-port/optics-current/optics-second30/optics-second30-optics	Provides the details of Optics PM counters for 30 second bucket such as, LB+E4C, OPT, OPR.
Native	Cisco-IOS-XR-pmengine-oper:performance-management/ethernet/ethernet-ports/ethernet-port/ethernet-current/ethernet-second30/second30-ethers	Provides the details of Ethernet PM counters for 30 second bucket such as, STAT-PKT, TX-PKT.
Native	Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization	Provides the snapshot of current CPU utilization of the node.
Native	Cisco-IOS-XR-nto-misc-oper:memory-summary/nodes/node/summary	Provides the snapshot of memory utilization of the node.
Native	Cisco-IOS-XR-osa-oper:osa/node-ids/node-id/mxponder-slices	Provides the details of muxponder slices configured on the node.
Native	Cisco-IOS-XR-spirit-install-instmgr-oper:software-install/active	Provides the details of the current active ISO and RPMs on the node.

Model Type	Sensor Path	Description
OpenConfig	openconfig-system:system	Checks for the alarms, host name, SSH configuration, and gRPC configuration.
OpenConfig	openconfig-platform:components/component	Checks for the inventory of the node such as subcomponents and field replaceable units such as, QSFP-100G-LR4-S, QSFP-100G-CWDM4-S.
OpenConfig	openconfig-terminal-device:terminal-device	Provides the supported operational modes of the OC terminal device configuration.
OpenConfig	openconfig-terminal-device:terminal-device/logical-channels/channel/otn/state	Provides the details of PM counters for 10 second history bucket for OTN/ODU logical channels trunk ports such as, ES-NE, ESR-NE, BBE-FE.
OpenConfig	openconfig-terminal-device:terminal-device/logical-channels/channel/ethernet/state	Provides the details of PCS counters for 10 second history bucket and all the other packet counters as cumulative for the ethernet logical channel client ports such as, STAT-MULTICAST-PKT, TX-PKT, IN-MCAST.

Subscription

A subscription binds one or more sensor paths and destinations. An MDT-capable device streams data for each sensor path at the configured frequency (cadence-based streaming) or when the sensor-path content changes (event-based streaming) to the destination.

The following example shows subscription `SUB1` that associates a sensor-group, sample interval and destination group.

```
Router(config)#telemetry model-driven
Router(config-model-driven)#subscription SUB1
Router(config-model-driven-subs)#sensor-group-id SGROUP1 sample-interval 10000
Router(config-model-driven-subs)#strict-timer
```

Note With a `strict-timer` configured for the sample interval, the data collection starts exactly at the configured time interval allowing a more deterministic behavior to stream data.

In 32-bit platforms, `strict-timer` can be configured only under the subscription. Whereas, 64-bit platforms support configurtion at global level in addition to the subscription level. However, configuring at the global level will affect all configured subscriptions.

```
Router(config)#telemetry model-driven
Router(config-model-driven)#strict-timer
```

Transport and Encoding

The router streams telemetry data using a transport mechanism. The generated data is encapsulated into the desired format using encoders.

Model-Driven Telemetry (MDT) data is streamed through these supported transport mechanisms:

- **Google Protocol RPC (gRPC):** used for both dial-in and dial-out modes.
- **Transmission Control Protocol (TCP):** used for only dial-out mode.
- **User Datagram Protocol (UDP):** used for only dial-out mode.

The data to be streamed can be encoded into Google Protocol Buffers (GPB) or JavaScript Object Notation (JSON) encoding. In GPB, the encoding can either be compact GPB (for optimising the network bandwidth usage) or self-describing GPB. The encodings supported are:

- **GPB encoding:** configuring for GPB encoding requires metadata in the form of compiled .proto files. A .proto file describes the GPB message format, which is used to stream data. The .proto files are available in the Github repository.
    - **Compact GPB encoding:** data is streamed in compressed and non self-describing format. A .proto file corresponding to each sensor-path must be used by the receiver to decode the streamed data.
    - **Key-value (KV-GPB) encoding:** data of each sensor path streamed is in a self-describing formatted ASCII text. A single .proto file `telemetry.proto` is used by the receiver to decode any sensor path data. Because the key names are included in the streamed data, the data on the wire is much larger as compared to compact GPB encoding.
- **JSON encoding**