Establish a Model-Driven Telemetry Session from a Destination to a Router

Streaming telemetry is a new paradigm in monitoring the health of a network. It provides a mechanism to efficiently stream configuration and operational data of interest from Cisco IOS XR routers. This streamed data is transmitted in a structured format to remote management stations for monitoring and troubleshooting purposes.

With telemetry data, you create a data lake. Analyzing this data, you proactively monitor your network, monitor utilization of CPU and memory, identify patterns, troubleshoot your network in a predictive manner, and devise strategies to create a resilient network using automation.

Telemetry works on a subscription model where you subscribe to the data of interest in the form of sensor paths. The sensor paths describe OpenConfig data models or native Cisco data models. You can access the OpenConfig and Native data models for telemetry from Github, a software development platform that provides hosting services for version control. You choose who initiates the subscription by establishing a telemetry session between the router and the receiver. The session is established using either a dial-out mode or dial-in mode, described in the Scale-Up Your Network Monitoring Strategy Using Telemetry article.

Watch this video to discover the power of real-time network management using model-driven telemetry.

This article describes the dial-in mode where a receiver dials in to the router to establish a telemetry session. In this mode, the receiver dials in to the router, and subscribes dynamically to one or more sensor paths specified in a subscription. The router streams telemetry data through the same session that is established by the receiver. The dial-in mode of subscriptions is dynamic. This dynamic subscription terminates when the receiver cancels the subscription or when the session terminates.

The following image shows a high-level overview of the dial-in mode:
This article describes, with a use case that illustrates the simultaneous monitoring of various parameters in the network, how streaming telemetry data helps you gain better visibility of the network, and make informed decisions to stabilize it.

• Monitor Network Parameters Using Telemetry Data for Proactive Analysis, on page 2

Monitor Network Parameters Using Telemetry Data for Proactive Analysis

The use case illustrates how, with the dial-in mode, you can use telemetry data to stream various parameters about your network. You use this data for predictive analysis where you monitor patterns, and proactively troubleshoot issues. This use case describes the tools used in the open-sourced collection stack to store and analyse telemetry data.

Note
Watch this video to see how you configure model-driven telemetry to take advantage of data models, open source collectors, encodings and integrate into monitoring tools.

Telemetry involves the following workflow:

• Define: You define a subscription to stream data from the router to the receiver. To define a subscription, you create a sensor-group.

• Deploy: The receiver initiates a session with the router and establishes a subscription-based telemetry session. The router streams data to the receiver. You verify subscription deployment on the router.

• Operate: You consume and analyse telemetry data using open-source tools, and take necessary actions based on the analysis.
Before you begin

Ensure you meet these dependencies:

- Make sure you have L3 connectivity between the router and the receiver.
- Enable gRPC server on the router to accept incoming connections from the receiver.

```
Router#configure
Router(config)#grpc
Router(config-grpc)#port <port-number>
Router(config-grpc)#commit
```

The port-number ranges from 57344 to 57999. If a port number is unavailable, an error is displayed.

The following example shows the output of the gRPC configuration with TLS enabled on the router.

```
Router#show grpc
Address family : ipv4
Port : 57300
VRF : global-vrf
TLS : enabled
TLS mutual : disabled
Trustpoint : none
Maximum requests : 128
Maximum requests per user : 10
Maximum streams : 32
Maximum streams per user : 32
TLS cipher suites
Default : none
Enable : none
Disable : none
Operational enable : ecdhe-rsa-chacha20-poly1305
: ecdhe-ecdsa-chacha20-poly1305
: ecdhe-rsa-aes128-gcm-sha256
: ecdhe-ecdsa-aes128-gcm-sha256
: ecdhe-rsa-aes128-sha
Operational disable : none
```

Define a Subscription to Stream Data from Router to Receiver

Create a subscription to define the data of interest to be streamed from the router to the destination.

**Step 1**

Specify the subset of the data that you want to stream from the router using sensor paths. The sensor path represents the path in the hierarchy of a YANG data model. This example uses the native data model `Cisco-IOS-XR-um-telemetry-model-driven-cfg.yang`. Create a sensor-group to contain the sensor paths.

**Example:**

```
sensor-group health
  sensor-path Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization
  sensor-path Cisco-IOS-XR-nto-misc-oper:memory-summary/nodes/node/summary
  sensor-path Cisco-IOS-XR-IPSYSOS-oper:system-time/uptime

sensor-group interfaces
  sensor-path Cisco-IOS-XR-pfi-im-cmd-oper:interfaces/interface/latest/generic-counters
  sensor-path Cisco-IOS-XR-IPSYSOS-oper:interfaces/interface-summary

sensor-group optics
```

Define a Model-Driven Telemetry Session from a Destination to a Router
Step 2
Subscribe to telemetry data that is streamed from a router. A subscription binds the sensor-group, and sets the streaming method. The streaming method can be cadence-driven or event-driven. Separating the sensor-paths into different subscriptions enhances the efficiency of the router to retrieve operational data at scale.

Example:

```
subscription health
  sensor-group-id health strict-timer
  sensor-group-id health sample-interval 30000
!
subscription interfaces
  sensor-group-id interfaces strict-timer
  sensor-group-id interfaces sample-interval 30000
!
subscription optics
  sensor-group-id optics strict-timer
  sensor-group-id optics sample-interval 30000
!
subscription routing
  sensor-group-id routing strict-timer
  sensor-group-id routing sample-interval 30000
!
```

Verify Deployment of the Subscription

The receiver dials into the receiver to establish a dynamic session based on the subscription. After the session is established, the router streams data to the receiver to create a data lake.

You can verify the deployment of the subscription on the router.
Verify the state of the subscription. An *Active* state indicates that the router is ready to stream data to the receiver based on the subscription.

**Example:**

```
Router#show telemetry model-driven subscription
```

The router streams data to the receiver using the subscription-based telemetry session and creates a data lake in the receiver.

---

**Operate on Telemetry Data for In-depth Analysis of the Network**

You can start consuming and analyzing telemetry data from the data lake using an open-sourced collection stack. This use case uses the following tools from the collection stack:

- **Pipeline** is a lightweight tool used to collect data. You can download Network Telemetry Pipeline from Github. You define how you want the collector to interact with routers, and where you want to send the processed data using *pipeline.conf* file.

- **Telegraph or InfluxDB** is a time series database (TSDB) that stores telemetry data, which is retrieved by visualization tools. You can download InfluxDB from Github. You define what data you want to include into your TSDB using the *metrics.json* file.

- **Grafana** is a visualization tool that displays graphs and counters for data streamed from the router.

In summary, Pipeline accepts TCP and gRPC telemetry streams, converts data and pushes data to the InfluxDB database. Grafana uses the data from InfluxDB database to build dashboards and graphs. Pipeline and InfluxDB may run on the same server or on different servers.

Consider that the router is monitored for the following parameters:

- Memory and CPU utilization
- Interface counters and interface summary
- Transmitter and receiver power levels from optic controllers
- ISIS route counts and ISIS interfaces
- BGP neighbours, path count, and prefix count
- MPLS-TE tunnel summary
- RSVP control messages and bandwidth allocation for each interface

---

**Step 1**

Start Pipeline, and enter your router credentials.

**Example:**

```
$ bin/pipeline -config pipeline.conf
```

Startup pipeline
Load config from [pipeline.conf], logging in [pipeline.log]

CRYPT Client [grpc_in_mymdtrouter], [http://172.0.0.0:5432]
Enter username: <username>
Enter password: <password>
Wait for `^C` to shutdown

The streamed telemetry data is stored in InfluxDB.

**Step 2**

Use Grafana to create a dashboard and visualize the streamed data.

*Figure 2: Visual Analysis of Network Health using Telemetry Data*

*Figure 3: Visual Analysis of System Monitoring using Telemetry Data*
In conclusion, telemetry data shows that various parameters of the network can be monitored simultaneously. This data is streamed in near real-time without affecting the performance of the network. With this data, you gain better visibility into your network.
Operate on Telemetry Data for In-depth Analysis of the Network