



OcNOS®

Open Compute Network Operating System Version 6.5.2

Streaming Telemetry Guide

August 2024

© 2024 IP Infusion Inc. All Rights Reserved.

This documentation is subject to change without notice. The software described in this document and this documentation are furnished under a license agreement or nondisclosure agreement. The software and documentation may be used or copied only in accordance with the terms of the applicable agreement. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's internal use without the written permission of IP Infusion Inc.

IP Infusion Inc.

3965 Freedom Circle, Suite 200

Santa Clara, CA 95054

+1 408-400-1900

<http://www.ipinfusion.com/>

For support, questions, or comments via E-mail, contact:

support@ipinfusion.com

Trademarks:

IP Infusion and OcNOS are trademarks or registered trademarks of IP Infusion. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Use of certain software included in this equipment is subject to the IP Infusion, Inc. End User License Agreement at <http://www.ipinfusion.com/license>. By using the equipment, you accept the terms of the End User License Agreement.

Contents

Preface	4
Audience	4
IP Maestro Support	4
Conventions	4
Related Documentation	4
Feature Availability	4
Migration Guide	4
Support	5
Comments	5
CHAPTER 1 Streaming Telemetry	6
Overview	6
gnmic Installation	8
Streaming Telemetry Commands	8
Troubleshooting	14
CHAPTER 2 Streaming Telemetry Dial-In Mode	16
Overview	16
Prerequisites	19
Configuration	19
Implementation Examples	38
Dial-In Mode Commands	38
Abbreviations	39
Glossary	40
CHAPTER 3 Streaming Telemetry Dial-Out Mode	41
Overview	41
Prerequisites	43
Configuration	43
Implementation Examples	48
Dial-Out Commands	48
Revised CLI Commands	58
Glossary	58
CHAPTER 4 Streaming Telemetry Data Models	60
IPI Data Models	60
OpenConfig Data Models	76

Preface

This guide describes how to configure Streaming Telemetry in OcNOS.

Audience

This guide is intended for network administrators and other engineering professionals who configure OcNOS.

IP Maestro Support

Monitor devices running OcNOS Release 6.3.4-70 and above using IP Maestro software.

Conventions

The table below shows the conventions used in this guide.

Table 1: Conventions

Convention	Description
Italics	Emphasized terms; titles of books
Note:	Special instructions, suggestions, or warnings
<code>monospaced type</code>	Code elements such as commands, parameters, files, and directories

Related Documentation

For information about installing OcNOS, see the *Installation Guide* for your platform.

Feature Availability

The features described in this document that are available depend upon the OcNOS SKU that you purchased. See the <https://www.ipinfusion.com/documentation/ocnos-feature-matrix/> - *Feature Matrix* for a description of the OcNOS SKUs.

Migration Guide

Check the *Migration Guide* for configuration changes to make when migrating from one version of OcNOS to another.

Support

For support-related questions, contact support@ipinfusion.com.

Comments

If you have comments, or need to report a problem with the content, contact techpubs@ipinfusion.com.

CHAPTER 1 Streaming Telemetry

Overview

Streaming telemetry allows users to monitor network health by efficiently streaming operational data of interest from OcNOS routers. This structured data is transmitted to remote management systems for proactive network monitoring and understanding CPU and memory usage in managed devices for troubleshooting.

A machine learning (ML) database can be created with telemetry data to establish a baseline for normal network operation and predict or mitigate network issues.

Feature Characteristics

In OcNOS various gRPC Network Management Interface (gNMI) Subscription Modes, Telemetry Modes, and Encoding Types are supported, providing efficient network management capabilities.

gNMI Subscription Modes

Streaming Telemetry Dial-In Mode: In this mode, the collector initiates a connection to the target device (OcNOS) and subscribes to receive telemetry data from OcNOS devices.

Streaming Telemetry Dial-Out Mode (Persistent Subscriptions): In this mode, the target (OcNOS) initiates the gRPC tunnel connection to the collector. Once the connection is established, the collector invokes the "Publish" RPC on the target. Subscriptions configured on the target are then streamed on that connection at the specified sample interval. These subscriptions remain active on OcNOS devices as long as the corresponding configuration on the target exists. If the gRPC tunnel connection is interrupted or the target reboots, the gNMI server on the target re-establishes the connection to the gNMI collector, ensuring continued streaming.

Streaming Telemetry Modes

Stream Mode: Enables continuous and real-time transmission of telemetry data from OcNOS devices to the monitoring system. The stream mode applies to both the dial-in and dial-out gNMI subscription modes.

Poll Mode: Poll mode subscriptions allow for on-demand data retrieval through a long-lived RPC. Subscribers initiate this mode by sending a Subscribe request message, followed by sending an empty Poll message to receive the desired data.

Note: The system supports **Poll mode** only in Dial-in subscription mode.

Once Mode: In Once mode subscription, the OcNOS device responds to a subscribe request with a one-time data retrieval, similar to a get request. Upon receiving the Once mode subscribe request, the device sends back the subscribe response for all subscriptions in the list and terminates the RPC.

Note: The system supports **Once mode** only in Dial-in subscription mode.

gNMI In-Band Support

gNMI In-Band support in OcNOS enables streaming telemetry data transmission across any one of the default, management, and user-defined VRFs. If no VRF is defined, streaming telemetry is automatically enabled within the default VRF. This enhancement allows network operators to utilize existing data interfaces for efficient in-band telemetry data transmission.

Encoding Types

Protocol Buffers (protobuf): Offers a compact binary serialization format for efficient encoding and transmission of structured telemetry data. Protobuf is optimized for performance and bandwidth efficiency.

JavaScript Object Notation (JSON): Provides a human-readable data interchange format commonly used for telemetry data representation. JSON encoding facilitates interoperability and ease of integration with various applications and tools. It adheres to the JSON specification outlined in RFC7159, employing relevant quoting. Consequently, string values are quoted while number values remain unquoted.

JSON-IETF: This variant of JSON encoding adheres to the IETF standards, ensuring consistency and compatibility with industry specifications. JSON_IETF encoded data conforms the rules outlined in RFC7951 for JSON serialization.

OcNOS supports the protobuf, JSON, and JSON-IETF encoding types for both the dial-in and dial-out gNMI subscription modes

Support for IPI Native Data Models and OpenConfig Data Models

[Streaming Telemetry Data Models](#): OcNOS supports IPI native data models and OpenConfig data models, providing standardized representations of network configurations and telemetry data. This support enhances interoperability and facilitates consistent management across heterogeneous network environments.

Scale and Minimum Sample Interval Supported

To limit the impact of telemetry on critical features of the OcNOS target device, certain limits have been implemented for different platform types.

High End Platforms

A system is considered high range if it has eight or more CPU cores and is not based on an "Intel Atom" processor. Users can subscribe to a maximum of 100 sensor paths (including Dial-In and Dial-Out subscriptions) at any given time. The minimum supported sample interval is 10 seconds.

Standard Range Platforms

A system is considered standard range if it has fewer than eight CPU cores or is based on an "Intel Atom" processor. Users can subscribe to a maximum of 50 sensor paths (including Dial-In and Dial-Out subscriptions) at any given time. The minimum supported sample interval is 90 seconds.

Note:

The total count of sensor paths includes the child paths of a subscribe request. For instance, if a subscribe request has four child paths, the total sensor paths count equals five (the given path plus four child paths). Use the [show streaming-telemetry](#) command to display the minimum sample interval and the maximum number of sensor paths supported for a platform.

Scale Scenarios

- New Subscribe RPC Request Makes Total Paths To Not Exceed Max Allowed:** When new paths are added to the existing paths already handled by the gNMI server, the total number does not exceed the maximum limit of 50 paths for standard platforms and 100 paths for high-end platforms. Consequently, the gNMI server accepts this subscribe request and proceeds with the processing.
- New Subscribe RPC Request Makes Total Paths To Reach Allowed Max:** With the new Subscribe RPC Request, the total paths handled would be exactly equal to 50 paths for standard platforms and 100 paths for high-end platforms. The gNMI server accepts the new subscribe request; however, a warning is logged by the gNMI server, indicating that the maximum number of paths has been reached, and it signifies that no new Subscribe RPC Stream mode requests will be handled until the number of currently handled paths drops below 100.
- New Subscribe RPC Request Makes Total Paths To Exceed Allowed Max:** With the new Subscribe RPC Request, the total paths handled exceed 50 paths for standard platforms and 100 paths for high-end platforms. The gNMI server returns an error. The RPC request is not closed but will be accepted and responded to when the total number of paths handled drops to a level that can accommodate this RPC request.

Minimum Sample Interval: The minimum supported sample interval is 10 seconds for “High-end” platform and 90 seconds for “Standard” platform type. Any sampling mode request with a sample interval of less than the minimum allowed will result in an error. However, if a sample interval is 0, it defaults to the minimum sample interval supported by the gNMI server for that platform type.

gnmic Installation

gNMI Specification can be found at: <https://github.com/openconfig/reference/blob/master/rpc/gnmi/gnmi-specification.md>.

gNMI Collector Tool

- For dial-in subscription mode, except when using “proto” encoding, use the open-source gNMI collector tool (gnmic). Install the open-source gNMI collector tool (gnmic) with the command:

```
bash -c "$(curl -sL https://get-gnmic.openconfig.net)"
```
- For dial-out subscription mode or when “proto” encoding is needed, use the gnmic tool from the gNMI collector package. It is delivered with the OcNOS installer, named `OcNOS-<SKU_NAME>-<version>-telemetry-client-bin.tar`, and includes the gNMI Client collector application (gnmic) and the IPI-OC.proto files.

Streaming Telemetry Commands

This section lists the telemetry commands.

- [debug cml](#)
- [debug telemetry gnmi](#)
- [destination-group](#)
- [destination-group GRPC](#)
- [encoding](#)
- [feature streaming-telemetry](#)
- [grpc-tunnel-server retry-interval](#)
- [sensor-group](#)
- [sensor-group sample-interval](#)
- [sensor-path](#)
- [show streaming-telemetry](#)
- [show running-config streaming-telemetry](#)
- [show streaming-telemetry dynamic-subscriptions](#)
- [show streaming-telemetry persistent-subscriptions](#)
- [subscription-name](#)
- [tunnel-server](#)

debug cml

Use this command to enable or disable debugging information for CML streaming telemetry.

Command Syntax

```
debug cml enable telemetry
debug cml disable telemetry
```

Parameters

None

Default

By default, debugging information is disabled.

Command Mode

Exec Mode

Applicability

This command was introduced in OcNOS version 6.4.1.

Examples

The following example illustrates how to enable and disable the telemetry debugging information.

```
OcNOS#debug cml enable telemetry
OcNOS#debug cml disable telemetry
```

debug telemetry gnmi

Use this command to enable or disable gNMI server debugging logs with severity levels.

Command Syntax

```
debug telemetry gnmi (enable) (severity (debug|info|warning|error|fatal|panic|d-panic)) (vrf (management|NAME))
debug telemetry gnmi (disable) (severity (debug|info|warning|error|fatal|panic|d-panic)) (vrf (management|NAME))
```

Parameters

debug	Logs a message at debug level
info	Logs a message at info level
warning	Logs a message at warning level
error	Logs a message at error level
fatal	Logs a message and causes the program to exit with return code 1.
panic	Logs a message and triggers the program to generate a traceback.
d-panic	Logs at the Panic level
d-panic	Logs at the Panic level
d-panic	Logs at the Panic level
vrf management	(Optional) Enables gNMI server debugging logs in the management VRF.
vrf NAME	(Optional) Disables gNMI server debugging logs in a user-defined VRF.

Default

By default, this command is disabled, and the gNMI server debugging level in the disabled state is set to the Error level.

Command Mode

Configure Mode

Applicability

Introduced in OcNOS version 6.4.1 and added the `vrf (NAME|management)` parameter in the OcNOS version 6.5.2.

Examples

The following example illustrates how to enable and disable the telemetry debug logs and their corresponding show output.

```
OcNOS(config)#feature streaming-telemetry
OcNOS(config)#debug telemetry gnmi enable severity warning
OcNOS(config)#commit
OcNOS(config)#show running-config streaming-telemetry
!
feature streaming-telemetry
debug telemetry gnmi enable severity warning
!
OcNOS(config)#debug telemetry gnmi disable severity warning
OcNOS(config)#commit
OcNOS(config)#show running-config streaming-telemetry
!
feature streaming-telemetry
!
```

feature streaming-telemetry

Use this command to enable the streaming telemetry and, upon configuration, to start the gNMI server. The gNMI server initiates listening for incoming gRPC connections on port 9339.

Note:

- Users can configure streaming telemetry on any one of the default, management, and user-defined VRFs. If no VRF is defined, streaming telemetry is automatically enabled in the default VRF.
- If streaming telemetry is already configured in any VRF (default, management, or user-defined), attempting to configure it for another VRF will result in an error, as gNMI can only be enabled on one VRF at a time. To configure the streaming feature on a new VRF, disable the streaming telemetry on the current VRF first, commit the change, and then configure the feature on the required VRF.

Use the `no` parameter of this command to disable the streaming telemetry, It will stop the gNMI server.

Command Syntax

```
feature streaming-telemetry (vrf (NAME|management))
no feature streaming-telemetry (vrf (NAME|management))
```

Parameters

<code>vrf NAME</code>	(Optional) Enables streaming telemetry in a user-defined VRF.
<code>vrf management</code>	(Optional) Enables streaming telemetry in the management VRF.

Default

By default, the streaming-telemetry feature is disabled.

Command Mode

Configure mode

Applicability

Introduced in OcNOS version 6.4.1 and added the `vrf (NAME|management)` parameter in the OcNOS version 6.5.2.

Examples

The following example illustrates how to enable the streaming telemetry on the default, management, and user-defined VRFs.

Default VRF

```
OcNOS#configure terminal
OcNOS (config)#feature streaming-telemetry
OcNOS (config)#commit
```

Management VRF

```
OcNOS#configure terminal
OcNOS (config)#feature streaming-telemetry vrf management
OcNOS (config)#commit
```

User-defined VRF

```
OcNOS#configure terminal
OcNOS (config)#feature streaming-telemetry vrf VRF1
OcNOS (config)#commit
```

show streaming-telemetry

Use this command to display the streaming-telemetry details. of persistent (dial-out) and dynamic (dial-in) subscription connection details, including the POLL mode subscriptions.

The “show streaming-telemetry and all its sub-commands” also shows the max sensor-paths and minimum sample-interval for that platform.

Command Syntax

```
show streaming-telemetry
```

Parameters

None

Command Mode

Exec mode

Applicability

This command was introduced in OcNOS version 6.5.2.

Examples

The following example displays the streaming telemetry details.

```
OcNOS#show streaming-telemetry

Feature streaming telemetry   : Enabled

VRF                           : management
Platform type                 : Standard range
Maximum sensor-paths         : 50
Minimum sample-interval      : 90
Number of active sensor-paths : 6 (Dial-In : 2, Dial-out : 4)
Tunnel-server Retry-interval  : Default-60 (seconds)

SI          : Sampling Interval in seconds
Enc-Type    : Encoding type
OriginPath  : Sensor Path

Dial-In STREAM Mode Subscription Details:
~~~~~
ClientIP:Port      ID      SI      Enc-Type      Origin:Path
-----
10.12.43.175:41592  4234   95      JSON_IETF     ipi:/interfaces/interface[name=eth0]/state
                                                           ipi:/interfaces/interface[name=eth0]/state/counters

Dial-In POLL Mode Subscription Details:
~~~~~
ClientIP:Port      ID      Enc-Type      Origin:Path
-----
10.12.43.175:41592  11448   JSON_IETF     ipi:/components/component[name=PSU-1]/state/temperature
                                                           ipi:/components/component[name=PSU-1]/state/board-fru
                                                           ipi:/components/component[name=PSU-1]/state
                                                           ipi:/components/component[name=PSU-1]/state/memory

Dial-Out Subscription Details:
~~~~~
1. Subscription-name      : SUB-1
   Status                  : ACTIVE
   Enc-Type                : JSON
   Tunnel-server details:
   ~~~~~
   Destination-group      Status      Tunnel-IP:Port
   -----
   tunnel-1               IN-ACTIVE  10.12.66.160:11161

   Sensor-group details:
   ~~~~~
   Sensor-group           SI          Origin:Path
   -----
   Platform               100        ipi:/components/component[name=CHASSIS]/state
                                                           [*] ipi:/components/component[name=CHASSIS]/state/memory
                                                           [*] ipi:/components/component[name=CHASSIS]/state/board-fru
                                                           [*] ipi:/components/component[name=CHASSIS]/state/temperature
```

The below table explains the output fields.

show streaming-telemetry output details

Field	Description
Feature streaming telemetry	Shows if the streaming telemetry feature is enabled or disabled.
VRF	Specifies the VRF type.
Platform type	Displays the platform type is standard or high range.

show streaming-telemetry output details

Field	Description
Maximum sensor-paths	Shows the maximum number of sensor paths allowed. For more details, refer to Scale Scenarios section.
Minimum sample-interval	Indicates the minimum sampling interval in seconds. For more details, refer to Scale Scenarios section.
Number of active sensor-paths	Shows the total number of active sensor paths for Dial-In and Dial-Out subscriptions (Stream mode subscriptions).
Tunnel-server Retry-interval	Displays the duration between retry attempts in seconds.
Enc-Type	Indicates the encoding type used for each subscription.
SI	Denotes the sampling interval in seconds.
Origin:Path	Displays the origin and path of the data being monitored.
ClientIP:Port	Displays the IP address and port of the client in Dial-In Mode subscriptions.
ID	Shows the unique identifier for each subscription.
Subscription-name	Shows the name of the Dial-Out subscription.
Status	Indicates if the subscription is active or inactive.
Tunnel-server details	Provides details about the tunnel server, including destination group, status, and IP:Port.
Sensor-group details	Show the details about the sensor group, including the sampling interval and origin:path.

show running-config streaming-telemetry

Use this command to display streaming telemetry status in the running configuration.

Command Syntax

```
show running-config streaming-telemetry
```

Parameters

None

Command Mode

Exec mode and Configuration Mode

Applicability

This command was introduced in OcNOS version 6.4.1.

Examples

The following example shows the streaming telemetry status in the `show running-config` output.

```
OcNOS#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
OcNOS(config)#feature streaming-telemetry
OcNOS(config)#commit
OcNOS(config)#show running-config streaming-telemetry
!
feature streaming-telemetry
!
OcNOS(config)#exit
OcNOS#show running-config streaming-telemetry
!
feature streaming-telemetry
!
```

Troubleshooting

Follow the below troubleshooting steps, to debug telemetry related issues:

Verify Collector (gnmic) Command Options: Verify the input parameters, such as the sensor path, prefix and origin `"ipi:"`.

Check the Encoding Method Compatibility: Check that the request conforms to the supported encoding methods.

Ensure Proper Connectivity: Validate the connectivity between the router and the remote management system. This involves verifying network settings, ports, firewalls, and any potential disruptions in communication.

Collector: If `gnmic` does not receive a response or not receiving expected response, restart the request using the `--log` option. If more verbose debug output is needed, consider adding the `--debug` option as well. The `gnmic` tool displays the possible cause for any error, which helps in debugging the issue.

gNMI Server: If the issue is on server side, follow the steps below to troubleshoot telemetry issues on the OcNOS target. Enable debug and verify the logs in `/var/log/messages` file.

1. In configure mode, enable debug with a specific severity level either "info" or "debug" level, using the following command:

```
debug telemetry gnmi (enable) (severity
(debug|info|warning|error|fatal|panic|d-panic) |) (vrf (management|NAME) |)
```

Note: To disable the debug telemetry, configure `debug telemetry gnmi (disable)` command.

2. In Exec mode, enable telemetry related debugs, using the following command:

```
debug cml enable telemetry
```

Note: To disable telemetry related debugs, configure `"debug cml disable telemetry"` command.

3. To check the state of streaming telemetry, collect the output of the following commands based on the telemetry mode:

- For Dial-in mode, use: `show streaming-telemetry dynamic-subscriptions`.
- For Dial-out mode, use: `show streaming-telemetry persistent-subscriptions`.

Note:

For Dial-out mode, Subscription status could become inactive for the following reasons:

- Sensor group(s) and destination group(s) are not configured
- Destination group(s) are not configured
- Sensor group(s) are not configured

- Sensor-group(s) doesn't have any sensor-path(s) configured, and destination-group(s) doesn't have any tunnel-server(s) configured.
- Destination-group(s) doesn't have any tunnel-server(s) configured
- Sensor-group(s) doesn't have any sensor-path(s) configured

Note: If telemetry is in “disabled” state, then telemetry feature need to enabled.

4. Collect the output of the following command to gather diagnostic information and the logs in `/var/log/messages` file, to triage further.

```
show techsupport gnmi
```

CHAPTER 2 Streaming Telemetry Dial-In Mode

Overview

Dial-in mode in streaming telemetry enables collectors to initiate connections with OcNOS routers to receive operational data of interest. In this mode, the collector sends a Subscribe Remote Procedure Call (RPC) request to the gNMI server (OcNOS target device), specifying the data paths to monitor. The server then streams the requested telemetry data to the collector, facilitating proactive network monitoring and troubleshooting.

Feature Characteristics

The gNMI-based Dial-in mode telemetry for the management plane supports three types of telemetry subscriptions: **Poll, Once, and Stream**. It enables the "STREAM" type subscription with "SAMPLING" mode for the Subscribe RPC. The gNMI-based collector connects to the OcNOS target device and invokes the Subscribe RPC, specifying the set of path(s) of interest. Two key components are involved in this process:

- **gNMI Server (OcNOS Target):** The gNMI server operates within the OcNOS device, serving as the source of telemetry data. It supports the gNMI protocol, allowing gNMI-based clients (collectors) to request and receive streaming data. The server streams the requested data to the client according to the specified parameters.
- **gNMI Client (Collector):** The gNMI client, also known as the collector, runs outside the OcNOS target device and is responsible for receiving and gathering telemetry data. In this context, it is the entity that connects to the OcNOS target device to collect data using the gNMI protocol. The collector initiates the Subscribe RPC to specify the data of interest.

Figure 2-1 illustrates the gNMI client's (Collector) Subscribe request and response (RPC) interaction with the gNMI server (OcNOS Target).



Figure 2-1: Sample Subscribe Request

Example Message Flow: Subscribe Request and Response

Figure 2-2 illustrates a sample gnmic Subscribe Request and Subscribe Response between the collector and the OcNOS target device.

Step 1: Subscription Request Initiation

- The gnmic collector server initiates a Subscribe Request by sending a Subscribe RPC in Stream type.
- This subscription request aims explicitly to gather data related to interface state counters and CPU state.
- A fixed 30/45-second sampling interval is set for data collection.

Step 2: Data Collection and Processing

- The gNMI server, within the OcNOS router, is responsible for data collection.

- At regular 30/45-second intervals, it retrieves data from the sensor path, focusing on interface state counters and CPU State.
- The received data undergoes a validation process, and the data is transformed into the required encoding type.

Step 3: Continuous Subscription Response Streaming

- The gNMI Server responds to the subscription request by continuously streaming Subscribe Response data.
- This streaming process maintains the same 30/45-second interval as the data collection.
- The collected data is streamed in real-time to the gnmic collector server.

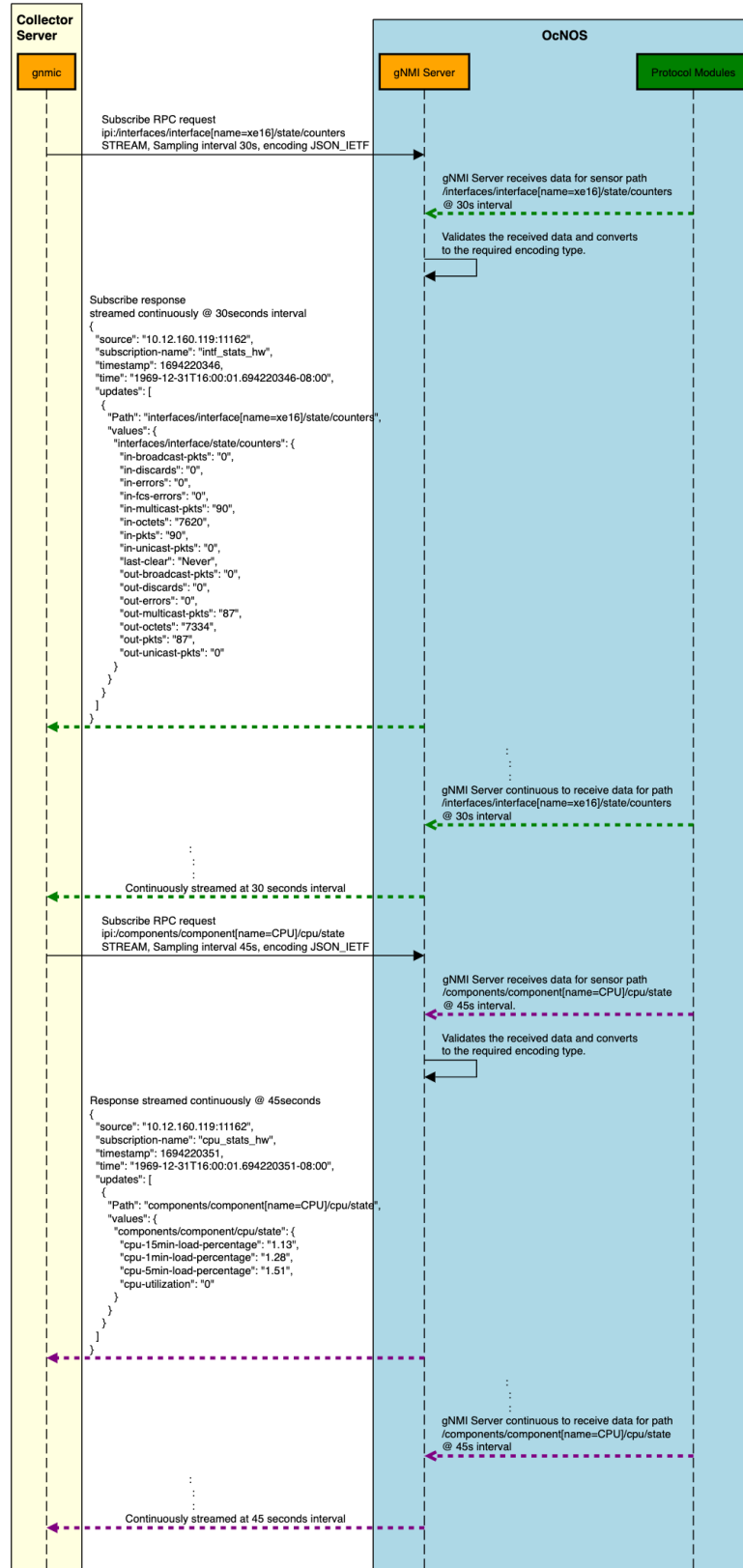


Figure 2-2: Message Flow: Subscribe Request and Response

Benefits

Proactive Network Monitoring: Obtain real-time insights into network health and performance, and how to enable quicker response to issues.

Resource Utilization Monitoring: Monitor CPU and memory utilization to optimize resource allocation and performance.

Predictive Troubleshooting: Identify patterns and potential issues before they impact the network, reducing downtime.

Automation and Resilience: Use telemetry data to automate network management tasks and design a more resilient network.

Prerequisites

Before configuring Dial-In mode, ensure that:

- A supported OcNOS router running a compatible release.
- Access to the management interface of the router.
- Any gNMI client that complies with gNMI specifications can be used as a client.
- Refer to the [gnmic Installation](#) to download the gNMI collector package.

Configuration

In this example, streaming telemetry with OcNOS is demonstrated, using 'gnmic' as the gNMI Client.

Note: To install the gnmic tool for managing network devices using gNMI, refer to the [gnmic Installation](#).

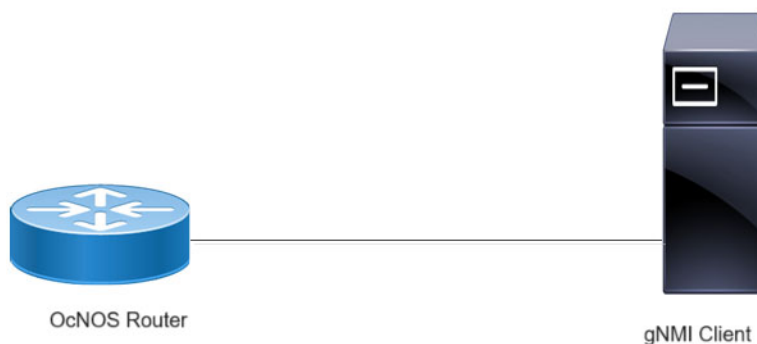


Figure 2-3: Dial-In Streaming Telemetry Topology

Note: Before configuring Dial-In, meet all [Prerequisites](#).

Enable streaming telemetry on OcNOS:

```
OcNOS#configure terminal
OcNOS (config)#feature streaming-telemetry
OcNOS (config)#commit
```

Telemetry Subscription Request via gnmic Command and YAML Input

Use the gnmic command with a YAML file input to request telemetry subscriptions with multiple paths.

```
gnmic -a <ipaddress:port> -u <UserName> -p <Password> --insecure --config <path to
config file> subscribe
```

This command establishes a telemetry subscription with the specified paths defined in the YAML file.

Telemetry Subscription Request via gnmic Command with a Single Path Option

Use the gnmic command with a single path option to request a telemetry subscription for a specific data path.

```
gnmic -a <ipaddress:port> -u <UserName> -p <Password> --encoding [json, json_ietf]
--insecure --mode STREAM --stream-mode sample --sample-interval sample-interval-
value sub --path <path>
```

```
gnmic -a <ipaddress:port> -u <UserName> -p <Password> --encoding [json, json_ietf]
--insecure --mode [POLL, ONCE] sub --path <path>
```

```
gnmic -a <ipaddress:port> -u <UserName> -p <Password> --encoding proto --insecure -
-mode STREAM --stream-mode sample --sample-interval sample-interval-value sub --
path <path> --proto-file protofile.proto
```

This command creates a telemetry subscription for the specified path with the chosen sample interval and encoding format.

Supported gnmic Options

The following table explains the option fields.

gnmic Option Details

Option	Description
--encoding	Specifies the encoding format as either <code>json</code> , <code>proto</code> , or <code>json_ietf</code> . The default encoding format is <code>json</code> .
--mode	Sets the mode of operation (<code>STREAM</code> , <code>POLL</code> , and <code>ONCE</code>).
--insecure	Allows insecure connections.
--stream-mode	Sets the stream mode (<code>Sample</code>).
--sample-interval	Sets the sample interval (10s). Note: Interval should be 10s or more.
--config	Specifies the YAML configuration file path (Example: <code>input_path.yaml</code>).
--path	Sets the path to subscribe to specific data (Example: <code>'ipi:/interfaces/interface[name]/state'</code>). Note: For multiple paths specify each path with <code>--path</code> option.
--prefix	Defines a common prefix for all specified paths (Example: <code>'ipi:/interfaces'</code>).
--proto-file	Path to Proto file (Example: <code>generated.proto</code>).

Invoking Subscribe RPC with gnmic

Use Case 1: Monitoring Interface State with Single Path Option

In this use case, gnmic subscribes to a specific path using the Subscribe RPC, monitoring the state of an interface with the path `'ipi:/interfaces/interface[name=ce51]/state'`.

```
#gnmic -a 10.12.91.111:9339 -u ocnos -p ocnos --encoding json ietf --insecure
--mode STREAM --stream-mode sample --sample-interval 10s sub --path 'ipi:/
interfaces/interface[name=ce51]/state'
```

```
{
  "source": "10.12.91.111:9339",
  "subscription-name": "default-1695368813",
  "timestamp": 1551956933,
  "time": "1970-01-01T05:30:01.551956933+05:30",
  "updates": [
    {
      "Path": "ipi:interfaces/interface[name=ce51]/state",
      "values": {
        "interfaces/interface/state": {
          "admin-status": "up",
          "counters": {
            "in-broadcast-pkts": "0",
            "in-discards": "0",
            "in-errors": "0",
            "in-fcs-errors": "0",
            "in-multicast-pkts": "23",
            "in-octets": "2126",
            "in-pkts": "23",
            "in-unicast-pkts": "0",
            "last-clear": "Never",
            "out-broadcast-pkts": "0",
            "out-discards": "0",
            "out-errors": "0",
            "out-multicast-pkts": "28",
            "out-octets": "2552",
            "out-pkts": "28",
            "out-unicast-pkts": "0"
          },
          "ifindex": 10051,
          "last-change": 15500,
          "logical": false,
          "oper-status": "up"
        }
      }
    }
  ]
}
```

The output of the Subscribe RPC includes the following information:

Subscribe RPC Output details

Option	Description
source	The source IP address and port of the gNMI server.
subscription-name	The name of the subscription.
timestamp	The timestamp of the response.
time	The timestamp in a human-readable format.
updates	An array of updates, each containing Path and Values.

Option	Description
Path	The path to the subscribed data.
values	The values of the subscribed data.

Validation

The below show command provides details about the subscriptions that have been established, including the client ID, sampling interval, encoding type, and the sensor paths that are being monitored.

```
OcNOS#show streaming-telemetry dynamic-subscriptions
```

```
Feature streaming telemetry : Enabled
```

```
VRF : default
```

```
Platform type : Standard range
```

```
Maximum sensor-paths : 50
```

```
Minimum sample-interval : 90
```

```
Number of active sensor-paths : 2 (Dial-In : 2, Dial-out : 0)
```

```
SI : Sampling Interval in seconds
```

```
Enc-Type : Encoding type
```

```
Origin:Path : Sensor Path
```

```
Dial-In STREAM Mode Subscription Details:
```

```
~~~~~
```

ClientIP:Port	ID	SI	Enc-Type	Origin:Path
-----	----	----	-----	-----
10.12.43.165:59304	4148	10	JSON_IETF	ipi:interfaces/interface[name=ce51]/state/counters ipi:interfaces/interface[name=ce51]/state

Use Case 2: Monitoring Interface State with Multiple Path Option

In this use case, gnmic subscribes to a specific path using the Subscribe RPC, monitoring the state of an interface with the multiple path 'ipi:/interfaces/interface[name=ce51]/state' and 'ipi:/interfaces/interface[name=ce52]/state'.

```
#gnmic -a 10.12.91.111:9339 -u ocnos -p ocnos --encoding json ietf --insecure --mode STREAM --stream-mode sample --sample-interval 11s sub --path 'ipi:/interfaces/interface[name=ce51]/state' --path 'ipi:/interfaces/interface[name=ce52]/state'
```

```
{
  "source": "10.12.91.111:9339",
  "subscription-name": "default-1695377304",
  "timestamp": 1551965423,
  "time": "1970-01-01T05:30:01.551965423+05:30",
  "updates": [
    {
      "Path": "ipi:interfaces/interface[name=ce51]/state",
      "values": {
        "interfaces/interface/state": {
          "admin-status": "up",
          "counters": {
            "in-broadcast-pkts": "0",
            "in-discards": "0",
            "in-errors": "0",
            "in-fcs-errors": "0",
            "in-multicast-pkts": "10",
            "in-octets": "1060",
```

```
        "in-pkts": "10",
        "in-unicast-pkts": "0",
        "last-clear": "Never",
        "out-broadcast-pkts": "0",
        "out-discards": "0",
        "out-errors": "0",
        "out-multicast-pkts": "10",
        "out-octets": "1020",
        "out-pkts": "10",
        "out-unicast-pkts": "0"
    },
    "ifindex": 10051,
    "last-change": 22500,
    "logical": false,
    "oper-status": "up"
}
}
]
}
{
    "source": "10.12.91.111:9339",
    "subscription-name": "default-1695377304",
    "timestamp": 1551965423,
    "time": "1970-01-01T05:30:01.551965423+05:30",
    "updates": [
        {
            "Path": "ipi:interfaces/interface[name=ce52]/state",
            "values": {
                "interfaces/interface/state": {
                    "admin-status": "up",
                    "counters": {
                        "in-broadcast-pkts": "0",
                        "in-discards": "0",
                        "in-errors": "0",
                        "in-fcs-errors": "0",
                        "in-multicast-pkts": "13",
                        "in-octets": "1664",
                        "in-pkts": "13",
                        "in-unicast-pkts": "0",
                        "last-clear": "Never",
                        "out-broadcast-pkts": "0",
                        "out-discards": "0",
                        "out-errors": "0",
                        "out-multicast-pkts": "10",
                        "out-octets": "1020",
                        "out-pkts": "10",
                        "out-unicast-pkts": "0"
                    },
                    "ifindex": 10052,
                    "last-change": 22500,
                    "logical": false,
                    "oper-status": "up"
                }
            }
        }
    ]
}
```

```
]
}
```

Validation

The below show command provides details about the subscriptions that have been established, including the client ID, sampling interval, encoding type, and the sensor paths that are being monitored.

```
OcNOS#show streaming-telemetry dynamic-subscriptions
```

```
Feature streaming telemetry : Enabled
```

```
VRF : default
Platform type : Standard range
Maximum sensor-paths : 50
Minimum sample-interval : 90
Number of active sensor-paths : 4 (Dial-In : 4, Dial-out : 0)
```

```
SI : Sampling Interval in seconds
Enc-Type : Encoding type
Origin:Path : Sensor Path
```

```
Dial-In STREAM Mode Subscription Details:
```

```
~~~~~
ClientIP:Port          ID      SI      Enc-Type      Origin:Path
-----
10.12.43.145:59334    42000  11      JSON_IETF     ipi:interfaces/interface[name=ce52]/state/counters
                                     ipi:interfaces/interface[name=ce52]/state
                                     ipi:interfaces/interface[name=ce51]/state/counters
                                     ipi:interfaces/interface[name=ce51]/state
```

Use Case 3: Monitoring Interface State Using Proto Encoding for IPI Xpath

In this use case, gnmic subscribes to the specified path using the Subscribe RPC, monitoring the state of an interface 'ipi:/interfaces/interface[name=eth0]/state' using the proto encoding.

```
./gnmic -a 10.12.160.119:9339 -u admin -p admin --insecure --mode STREAM --
stream-mode sample --sample-interval 10s subscribe --path 'ipi:/interfaces/
interface[name=eth0]/state' --encoding proto --proto-file generated.proto
```

```
{
  "source": "10.12.160.119:9339",
  "subscription-name": "default-1713864599",
  "timestamp": 1550833401388910124,
  "time": "2019-02-22T11:03:21.388910124Z",
  "updates": [
    {
      "Path": "ipi:interfaces/interface[name='eth0']/state",
      "values": {
        "interfaces/interface/state": {
          "adminStatus": "up",
          "counters": {
            "inMulticastPkts": "159435",
            "inOctets": "902863924",
            "inPkts": "732970",
            "lastClear": "Never",
            "outOctets": "8931839",
            "outPkts": "105457"
          },
          "ifindex": 3,
          "lastChange": 8000,
          "operStatus": "up"
        }
      }
    }
  ]
}
```



```

    }
  }
]
}

```

Validation

The below show command provides details about the subscriptions that have been established, including the client ID, sampling interval, encoding type, and the sensor paths that are being monitored.

```
OcNOS#show streaming-telemetry dynamic-subscriptions
```

```
Feature streaming telemetry : Enabled
```

```
VRF : default
Platform type : Standard range
Maximum sensor-paths : 50
Minimum sample-interval : 90
Number of active sensor-paths : 2 (Dial-In : 2, Dial-out : 0)
```

```
SI : Sampling Interval in seconds
Enc-Type : Encoding type
Origin:Path : Sensor Path
```

```
Dial-In STREAM Mode Subscription Details:
```

```

-----
ClientIP:Port      ID      SI      Enc-Type      Origin:Path
-----
10.12.66.120:53246  14087   10      PROTO         ipi:/interfaces/interface[name=eth0]/state
                                         ipi:/interfaces/interface[name=eth0]/state/counters

```

Use Case 4: Monitoring Interface State Using JSON Encoding for IPI Xpath

In this use case, gnmic subscribes to the specified path using the Subscribe RPC, monitoring the state of an interface 'ipi:/interfaces/interface[name=eth0]/state' using the JSON encoding.

```
./gnmic -a 10.12.160.119:9339 -u admin -p admin --insecure --mode STREAM --
stream-mode sample --sample-interval 10s subscribe --path 'ipi:/interfaces/
interface[name=eth0]/state' --encoding json
```

```

{
  "source": "10.12.160.119:9339",
  "subscription-name": "default-1713864619",
  "timestamp": 1550833421055660410,
  "time": "2019-02-22T11:03:41.05566041Z",
  "updates": [
    {
      "Path": "ipi:interfaces/interface[name=eth0]/state",
      "values": {
        "interfaces/interface/state": {
          "admin-status": "up",
          "counters": {
            "in-broadcast-pkts": 0,
            "in-discards": 0,
            "in-errors": 0,
            "in-multicast-pkts": 159470,
            "in-octets": 902867237,
            "in-pkts": 733016,
            "in-unicast-pkts": 0,
            "last-clear": "Never",
            "out-broadcast-pkts": 0,
            "out-discards": 0,

```

```

        "out-errors": 0,
        "out-multicast-pkts": 0,
        "out-octets": 8938196,
        "out-pkts": 105490,
        "out-unicast-pkts": 0
    },
    "ifindex": 3,
    "last-change": 8000,
    "logical": false,
    "oper-status": "up"
}
}
]
}

```

Validation

The below show command provides details about the subscriptions that have been established, including the client ID, sampling interval, encoding type, and the sensor paths that are being monitored.

```
OcNOS#show streaming-telemetry dynamic-subscriptions
```

```
Feature streaming telemetry : Enabled
```

```
VRF : default
Platform type : Standard range
Maximum sensor-paths : 50
Minimum sample-interval : 90
Number of active sensor-paths : 2 (Dial-In : 2, Dial-out : 0)
```

```
SI : Sampling Interval in seconds
Enc-Type : Encoding type
Origin:Path : Sensor Path
```

```
Dial-In STREAM Mode Subscription Details:
```

```

ClientIP:Port          ID      SI      Enc-Type      Origin:Path
-----
10.12.66.120:53340    50687   10      JSON          ipi:/interfaces/interface[name=eth0]/state
                                                           ipi:/interfaces/interface[name=eth0]/state/counters

```

Use Case 5: Monitoring Interface State Using JSON Encoding for OpenConfig Xpath

In this use case, gnmic subscribes to the specified path using the Subscribe RPC, monitoring the state of an interface `'/interfaces/interface[name=eth0]/state'` using the JSON encoding for OpenConfig Xpath.

```
./gnmic -a 10.12.160.119:9339 -u admin -p admin --insecure --mode STREAM --
stream-mode sample --sample-interval 10s subscribe --path '/interfaces/
interface[name=eth0]/state' --encoding json
```

```

{
  "source": "10.12.160.119:9339",
  "subscription-name": "default-1713864712",
  "timestamp": 1550833514789102094,
  "time": "2019-02-22T11:05:14.789102094Z",
  "updates": [
    {
      "Path": "interfaces/interface[name=eth0]/state",
      "values": {
        "interfaces/interface/state": {
          "AdminStatus": "UP",

```

```

    "Counters": {
      "InBroadcastPkts": 0,
      "InDiscards": 0,
      "InErrors": 0,
      "InFcsErrors": null,
      "InMulticastPkts": 159643,
      "InOctets": 902872585,
      "InPkts": 733096,
      "InUnicastPkts": 0,
      "LastClear": 0,
      "OutBroadcastPkts": 0,
      "OutDiscards": 0,
      "OutErrors": 0,
      "OutMulticastPkts": 0,
      "OutOctets": 8944684,
      "OutPkts": 105520,
      "OutUnicastPkts": 0
    },
    "Ifindex": 3,
    "LastChange": 8000,
    "Logical": false,
    "Name": "eth0",
    "OperStatus": "UP"
  }
}
]
}

```

Validation

The below show command provides details about the subscriptions that have been established, including the client ID, sampling interval, encoding type, and the sensor paths that are being monitored.

```
OcNOS#show streaming-telemetry dynamic-subscriptions
```

```
Feature streaming telemetry : Enabled
```

```
VRF : default
Platform type : Standard range
Maximum sensor-paths : 50
Minimum sample-interval : 90
Number of active sensor-paths : 2 (Dial-In : 2, Dial-out : 0)
```

```
SI : Sampling Interval in seconds
Enc-Type : Encoding type
Origin:Path : Sensor Path
```

```
Dial-In STREAM Mode Subscription Details:
```

```

-----
ClientIP:Port      ID      SI      Enc-Type      Origin:Path
-----
10.12.66.120:53390  16340   10      JSON          openconfig:/interfaces/interface[name='eth0']/state;
                                     openconfig:/interfaces/interface[name='eth0']/state/counters;

```

YAML File Input for Multiple Path Subscription

Use Case 1: Configuring One Subscription Requests with Multiple Path Option

This use case illustrates the configuration of a subscription request with multiple paths using a YAML file input. It streamlines the subscription setup process by specifying the desired paths and subscription parameters directly in the YAML file.

YAML File Content (`single_request.yaml`)

```
#cat single_request.yaml

subscriptions:                                #Container for subscriptions
  interface_stats_hw:                          #A named subscription, where the key is the subscription name
    paths:                                     #List of subscription paths for the named subscription
    - "ipi:/interfaces/
      interface[name=xel]/state"
    - "ipi:/interfaces/
      interface[name=vlan1.10]/
      state"

  stream-mode: sample                          #One of [on-change, target-defined, sample]
  sample-interval: 12s                         #Sampling interval (e.g., 12 seconds)
  encoding: json_ietf                          #Encoding format for telemetry data (e.g., JSON_IETF)
```

gnmic Command

```
# gnmic -a 10.12.91.111:9339 -u ocnos -p ocnos --insecure --config
single_request.yaml subscribe
```

```
{
  "source": "10.12.91.111:9339",
  "subscription-name": "interface_stats_hw",
  "timestamp": 1551965792,
  "time": "1970-01-01T05:30:01.551965792+05:30",
  "updates": [
    {
      "Path": "ipi:interfaces/interface[name=xel]/state",
      "values": {
        "interfaces/interface/state": {
          "admin-status": "up",
          "counters": {
            "in-broadcast-pkts": "0",
            "in-discards": "0",
            "in-errors": "0",
            "in-fcs-errors": "0",
            "in-multicast-pkts": "0",
            "in-octets": "0",
            "in-pkts": "0",
            "in-unicast-pkts": "0",
            "last-clear": "Never",
            "out-broadcast-pkts": "0",
            "out-discards": "0",
```

```
        "out-errors": "0",
        "out-multicast-pkts": "2",
        "out-octets": "164",
        "out-pkts": "2",
        "out-unicast-pkts": "0"
    },
    "ifindex": 10001,
    "last-change": 0,
    "logical": false,
    "oper-status": "down"
}
}
]
}
{
  "source": "10.12.91.111:9339",
  "subscription-name": "interface_stats_hw",
  "timestamp": 1551965792,
  "time": "1970-01-01T05:30:01.551965792+05:30",
  "updates": [
    {
      "Path": "ipi:interfaces/interface[name=vlan1.10]/state",
      "values": {
        "interfaces/interface/state": {
          "admin-status": "up",
          "counters": {
            "in-broadcast-pkts": "0",
            "in-discards": "0",
            "in-errors": "0",
            "in-fcs-errors": "0",
            "in-multicast-pkts": "0",
            "in-octets": "0",
            "in-pkts": "0",
            "in-unicast-pkts": "0",
            "last-clear": "Never",
            "out-broadcast-pkts": "0",
            "out-discards": "0",
            "out-errors": "0",
            "out-multicast-pkts": "0",
            "out-octets": "0",
            "out-pkts": "0",
            "out-unicast-pkts": "0"
          },
          "ifindex": 25010,
          "last-change": 22500,
          "logical": false,
          "oper-status": "up"
        }
      }
    }
  ]
}
```

Validation

The below show command provides details about the subscriptions that have been established, including the client ID, sampling interval, encoding type, and the sensor paths that are being monitored.

```
OcNOS#show streaming-telemetry dynamic-subscriptions
```

```
Feature streaming telemetry : Enabled

VRF : default
Platform type : Standard range
Maximum sensor-paths : 50
Minimum sample-interval : 90
Number of active sensor-paths : 4 (Dial-In : 4, Dial-out : 0)

SI : Sampling Interval in seconds
Enc-Type : Encoding type
Origin:Path : Sensor Path

Dial-In STREAM Mode Subscription Details:
~~~~~
ClientIP:Port          ID      SI      Enc-Type      Origin:Path
-----
10.12.43.135:58208    45333  12      JSON_IETF     ipi:interfaces/interface[name=xel]/state/counters
                    ipi:interfaces/interface[name=xel]/state
                    ipi:interfaces/interface[name=vlan1.10]/state/counters
                    ipi:interfaces/interface[name=vlan1.10]/state
```

Use Case 2: Configuring Multiple Subscription Requests with Multiple Path Option

This use case illustrates the configuration of multiple subscription request with multiple paths using a YAML file input. It streamlines the subscription setup process by specifying the desired paths and subscription parameters directly in the YAML file.

YAML File Content (**multiple_subs.yaml**)

```
#cat multiple_subs.yaml

subscriptions: # Container for subscriptions
  RAM_stats_hw: # A named subscription for RAM statistics
    paths: # List of subscription paths for the RAM_stats_hw subscription
    - "ipi:/components/
      component[name=RAM]/ram/state"
    stream-mode: sample # Stream mode for RAM statistics
    sample-interval: 11s # Sampling interval for RAM statistics (e.g., 11 seconds)
    encoding: json_ietf # Encoding format for RAM statistics (e.g., JSON_IETF)

  storage_stats_hw: # A named subscription for storage statistics
    paths: # List of subscription paths for the storage_stats_hw subscription
    - "ipi:/components/
      component[name=HARD-DISK]/
      storage/state"
    stream-mode: sample # Stream mode for storage statistics
    sample-interval: 12s # Sampling interval for storage statistics (e.g., 12 seconds)
    encoding: json_ietf # Encoding format for storage statistics (e.g., JSON_IETF)
```

```

power-supply_stats_hw:           # A named subscription for power supply statistics
  paths:                          # List of subscription paths for the power-supply_stats_hw subscription
- "ipi:/components/
component[name=PSU-1]/power-
supply/state"

- "ipi:/components/
component[name=PSU-2]/power-
supply/state"

  stream-mode: sample             # Stream mode for power supply statistics
  sample-interval: 13s           # Sampling interval for power supply statistics (e.g., 13 seconds)
  encoding: json_ietf            # Encoding format for power supply statistics (e.g., JSON_IETF)

intf-tray_stats_hw:             # A named subscription for interface tray statistics
  paths:                          # List of subscription paths for the intf-tray_stats_hw subscription
- "ipi:/interfaces/
interface[name=xel1]/state"

- "ipi:/interfaces/
interface[name=vlan1.8]/
state"

  stream-mode: sample             # Stream mode for interface tray statistics
  sample-interval: 14s           # Sampling interval for interface tray statistics (e.g., 14 seconds)
  encoding: json_ietf            # Encoding format for interface tray statistics (e.g., JSON_IETF)

```

gnmic Command

```

# gnmic -a 10.12.91.111:9339 -u ocnos -p ocnos --insecure --config
multiple_subs.yaml subscribe

{
  "source": "10.12.91.111:9339",
  "subscription-name": "ram_stats_hw",
  "timestamp": 1551967101,
  "time": "1970-01-01T05:30:01.551967101+05:30",
  "updates": [
    {
      "Path": "ipi:components/component[name=RAM]/ram/state",
      "values": {
        "components/component/ram/state": {
          "available-high-memory": "0",
          "available-memory": "14743",
          "buffers": "15",
          "current-process-count": 232,
          "free-swap": "0",
          "shared-memory": "8",
          "total-high-memory": "0",
          "total-memory": "16012",
          "total-swap": "0",
          "used-memory": "1269"
        }
      }
    }
  ]
}

```

```
    }
  }
}
]
}
{
  "source": "10.12.91.111:9339",
  "subscription-name": "storage_stats_hw",
  "timestamp": 1551967102,
  "time": "1970-01-01T05:30:01.551967102+05:30",
  "updates": [
    {
      "Path": "ipi:components/component[name=HARD-DISK]/storage/state",
      "values": {
        "components/component/storage/state": {
          "free-memory": "16908",
          "total-memory": "30208",
          "used-memory": "5020"
        }
      }
    }
  ]
}
{
  "source": "10.12.91.111:9339",
  "subscription-name": "power-supply_stats_hw",
  "timestamp": 1551967103,
  "time": "1970-01-01T05:30:01.551967103+05:30",
  "updates": [
    {
      "Path": "ipi:components/component[name=PSU-1]/power-supply/state",
      "values": {
        "components/component/power-supply/state": {
          "capacity": "650",
          "fan1-rpm": 24288,
          "operational-status": "not-present",
          "output-current": "8.28",
          "output-voltage": "12.07",
          "power-consumption": "99",
          "temperature-sensor1": "22",
          "temperature-sensor2": "28",
          "temperature-sensor3": "24"
        }
      }
    }
  ]
}
{
  "source": "10.12.91.111:9339",
  "subscription-name": "power-supply_stats_hw",
  "timestamp": 1551967103,
  "time": "1970-01-01T05:30:01.551967103+05:30",
  "updates": [
    {
```



```
"Path": "ipi:components/component[name=PSU-2]/power-supply/state",
"values": {
  "components/component/power-supply/state": {
    "operational-status": "running",
    "temperature-sensor1": "0",
    "temperature-sensor2": "0",
    "temperature-sensor3": "0"
  }
}
]
}
{
  "source": "10.12.91.111:9339",
  "subscription-name": "intf-tray_stats_hw",
  "timestamp": 1551967104,
  "time": "1970-01-01T05:30:01.551967104+05:30",
  "updates": [
    {
      "Path": "ipi:interfaces/interface[name=xel]/state",
      "values": {
        "interfaces/interface/state": {
          "admin-status": "up",
          "counters": {
            "in-broadcast-pkts": "0",
            "in-discards": "0",
            "in-errors": "0",
            "in-fcs-errors": "0",
            "in-multicast-pkts": "0",
            "in-octets": "0",
            "in-pkts": "0",
            "in-unicast-pkts": "0",
            "last-clear": "Never",
            "out-broadcast-pkts": "0",
            "out-discards": "0",
            "out-errors": "0",
            "out-multicast-pkts": "5",
            "out-octets": "410",
            "out-pkts": "5",
            "out-unicast-pkts": "0"
          },
          "ifindex": 10001,
          "last-change": 0,
          "logical": false,
          "oper-status": "down"
        }
      }
    }
  ]
}
{
  "source": "10.12.91.111:9339",
  "subscription-name": "intf-tray_stats_hw",
  "timestamp": 1551967104,
  "time": "1970-01-01T05:30:01.551967104+05:30",
```

```

"updates": [
  {
    "Path": "ipi:interfaces/interface[name=vlan1.8]/state",
    "values": {
      "interfaces/interface/state": {
        "admin-status": "up",
        "counters": {
          "in-broadcast-pkts": "0",
          "in-discards": "0",
          "in-errors": "0",
          "in-fcs-errors": "0",
          "in-multicast-pkts": "0",
          "in-octets": "0",
          "in-pkts": "0",
          "in-unicast-pkts": "0",
          "last-clear": "Never",
          "out-broadcast-pkts": "0",
          "out-discards": "0",
          "out-errors": "0",
          "out-multicast-pkts": "0",
          "out-octets": "0",
          "out-pkts": "0",
          "out-unicast-pkts": "0"
        },
        "ifindex": 25008,
        "last-change": 22500,
        "logical": false,
        "oper-status": "up"
      }
    }
  }
]
}

```

Validation

The below show command provides details about the subscriptions that have been established, including the client ID, sampling interval, encoding type, and the sensor paths that are being monitored.

```
OcNOS#show streaming-telemetry dynamic-subscriptions
```

```
Feature streaming telemetry : Enabled
```

```
VRF : default
```

```
Platform type : Standard range
```

```
Maximum sensor-paths : 50
```

```
Minimum sample-interval : 90
```

```
Number of active sensor-paths : 8 (Dial-In : 8, Dial-out : 0)
```

```
SI : Sampling Interval in seconds
```

```
Enc-Type : Encoding type
```

```
Origin:Path : Sensor Path
```

```
Dial-In STREAM Mode Subscription Details:
```

```

-----
ClientIP:Port      ID      SI      Enc-Type      Origin:Path
-----
10.12.43.155:58267  9453    14      JSON_IETF     ipi:interfaces/interface[name=xel]/state/counters
                  ipi:interfaces/interface[name=xel]/state
                  ipi:interfaces/interface[name=vlan1.8]/state/counters
                  ipi:interfaces/interface[name=vlan1.8]/state
10.12.43.155:58114  31533   11      JSON_IETF     ipi:components/component[name=RAM]/ram/state

```

```

10.12.43.155:58345      3374      12      JSON_IETF      ipi:components/component[name=HARD-DISK]/storage/state
10.12.43.155:58222      35994     13      JSON_IETF      ipi:components/component[name=PSU-1]/power-supply/state
                                     ipi:components/component[name=PSU-2]/power-supply/state

```

Use Case 3: Configuring Multiple Subscription Requests with Prefix Option

This use case illustrates the configuration of multiple subscription request with prefix option using a YAML file input. It streamlines the subscription setup process by specifying the desired paths and subscription parameters directly in the YAML file.

YAML File Content (`prefix_path.yaml`)

```

#cat prefix_path.yaml

subscriptions:                                #Container for subscriptions
  RAM_stats_hw:                               #A named subscription for RAM statistics
    prefix: "ipi:"                            #Common prefix for paths in this subscription
    paths:                                    #List of subscription paths for the RAM_stats_hw subscription
    - "/components/
component[name=RAM]/ram/
state"
    stream-mode: sample                       #Stream mode for RAM statistics
    sample-interval: 11s                     #Sampling interval for RAM statistics (e.g., 11 seconds)
    encoding: json_ietf                      #Encoding format for RAM statistics (e.g., JSON_IETF)

  intf-tray_stats_hw:                        #A named subscription for interface tray statistics
    prefix: "ipi:"                            #Common prefix for paths in this subscription
    paths:                                    #List of subscription paths for the intf-tray_stats_hw subscription
    - "/interfaces/
interface[name=xel]/state"
    - "/interfaces/
interface[name=vlan1.8]/
state"
    stream-mode: sample                       #Stream mode for interface tray statistics
    sample-interval: 14s                     #Sampling interval for interface tray statistics (e.g., 14 seconds)
    encoding: json_ietf                      #Encoding format for interface tray statistics (e.g., JSON_IETF)

```

gnmic Command

```

# gnmic -a 10.12.91.111:9339 -u ocnos -p ocnos --insecure --config
prefix_path.yaml subscribe
{
  "source": "10.12.91.111:9339",
  "subscription-name": "ram_stats_hw",
  "timestamp": 1551968637,
  "time": "1970-01-01T05:30:01.551968637+05:30",
  "updates": [
    {
      "Path": "components/component[name=RAM]/ram/state",
      "values": {

```

```
"components/component/ram/state": {
  "available-high-memory": "0",
  "available-memory": "14793",
  "buffers": "16",
  "current-process-count": 231,
  "free-swap": "0",
  "shared-memory": "8",
  "total-high-memory": "0",
  "total-memory": "16012",
  "total-swap": "0",
  "used-memory": "1219"
}
}
]
}
{
  "source": "10.12.91.111:9339",
  "subscription-name": "intf-tray_stats_hw",
  "timestamp": 1551968640,
  "time": "1970-01-01T05:30:01.55196864+05:30",
  "updates": [
    {
      "Path": "interfaces/interface[name=xel]/state",
      "values": {
        "interfaces/interface/state": {
          "admin-status": "up",
          "counters": {
            "in-broadcast-pkts": "0",
            "in-discards": "0",
            "in-errors": "0",
            "in-fcs-errors": "0",
            "in-multicast-pkts": "0",
            "in-octets": "0",
            "in-pkts": "0",
            "in-unicast-pkts": "0",
            "last-clear": "Never",
            "out-broadcast-pkts": "0",
            "out-discards": "0",
            "out-errors": "0",
            "out-multicast-pkts": "9",
            "out-octets": "738",
            "out-pkts": "9",
            "out-unicast-pkts": "0"
          },
          "ifindex": 10001,
          "last-change": 0,
          "logical": false,
          "oper-status": "down"
        }
      }
    }
  ]
}
{
```

```

"source": "10.12.91.111:9339",
"subscription-name": "intf-tray_stats_hw",
"timestamp": 1551968640,
"time": "1970-01-01T05:30:01.55196864+05:30",
"updates": [
  {
    "Path": "interfaces/interface[name=vlan1.8]/state",
    "values": {
      "interfaces/interface/state": {
        "admin-status": "up",
        "counters": {
          "in-broadcast-pkts": "0",
          "in-discards": "0",
          "in-errors": "0",
          "in-fcs-errors": "0",
          "in-multicast-pkts": "0",
          "in-octets": "0",
          "in-pkts": "0",
          "in-unicast-pkts": "0",
          "last-clear": "Never",
          "out-broadcast-pkts": "0",
          "out-discards": "0",
          "out-errors": "0",
          "out-multicast-pkts": "0",
          "out-octets": "0",
          "out-pkts": "0",
          "out-unicast-pkts": "0"
        },
        "ifindex": 25008,
        "last-change": 22500,
        "logical": false,
        "oper-status": "up"
      }
    }
  }
]
}

```

Validation

The below show command provides details about the subscriptions that have been established, including the client ID, sampling interval, encoding type, and the sensor paths that are being monitored.

```
OcNOS#show streaming-telemetry dynamic-subscriptions
```

```
Feature streaming telemetry : Enabled
```

```
VRF : default
Platform type : Standard range
Maximum sensor-paths : 50
Minimum sample-interval : 90
Number of active sensor-paths : 5 (Dial-In : 5, Dial-out : 0)
```

```
SI : Sampling Interval in seconds
Enc-Type : Encoding type
Origin:Path : Sensor Path
```

```
Dial-In STREAM Mode Subscription Details:
```

```

ClientIP:Port          ID      SI      Enc-Type      Origin:Path
-----

```

10.12.43.154:50167	32137	11	JSON_IETF	ipi:components/component[name=RAM]/ram/state
10.12.43.154:50614	36412	14	JSON_IETF	ipi:interfaces/interface[name=vlan1.8]/state/counters ipi:interfaces/interface[name=vlan1.8]/state ipi:interfaces/interface[name=xel1]/state/counters ipi:interfaces/interface[name=xel1]/state

Implementation Examples

Typical Use Cases

- Enable Streaming Telemetry to monitor interface counters and the health of the OcNOS target device, including memory, CPU usage, fan speed, and temperature.
- Use telemetry data to trigger automated network tasks based on specific conditions.

Integration with Existing Features

Streaming Telemetry can be used in conjunction with other network monitoring and management features.

Dial-In Mode Commands

The Streaming Telemetry introduces the following configuration commands.

show streaming-telemetry dynamic-subscriptions

Use this command to display the streaming telemetry dial-in configurations.

Command syntax

```
show streaming-telemetry dynamic-subscriptions
```

Parameters

None

Command Mode

Exec mode

Applicability

This command was introduced in OcNOS version 6.4.1.

Examples

The following example displays the streaming telemetry dial-in configuration output.

```
OcNOS#show streaming-telemetry dynamic-subscriptions

Feature streaming telemetry : Enabled

VRF : default
Platform type : Standard range
Maximum sensor-paths : 50
Minimum sample-interval : 90
```

Number of active sensor-paths : 3 (Dial-In : 3, Dial-out : 0)

SI : Sampling Interval in seconds
 Enc-Type : Encoding type
 Origin:Path : Sensor Path

Dial-In STREAM Mode Subscription Details:

```

-----
ClientIP:Port          ID      SI      Enc-Type      Origin:Path
-----
10.12.43.175:59108    12396  10      JSON_IETF     ipi:interfaces/interface[name=eth0]/state/counters
                                     ipi:interfaces/interface[name=eth0]/state
10.12.43.175:59114    6001   15      JSON_IETF     ipi:components/component[name=CPU]/cpu/state
  
```

The below table explains the output fields.

show streaming-telemetry dynamic-subscriptions parameters output details

Field	Description
Feature streaming telemetry	Marked as "Enabled" confirms that streaming telemetry is active on the device.
VRF	Specifies the VRF type.
Platform type	Displays the platform type is standard or high range.
Maximum sensor-paths	Shows the maximum number of sensor paths allowed. For more details, refer to Scale Scenarios section.
Minimum sample-interval	Indicates the minimum sampling interval in seconds. For more details, refer to Scale Scenarios section.
Number of active sensor-paths	Shows the total number of active sensor paths for Dial-In and Dial-Out subscriptions (Stream mode subscriptions).
Dial-In Subscription Details	Check the Dial-in subscription details.
ClientIP: Port	Verify that the client IP and port listed matches the client that should be receiving telemetry data.
SI: Sampling-interval	Confirm that the sampling interval matches the desired frequency at which data is collected and sent.
Enc-type: Encoding-type	Ensure that the encoding type (e.g., JSON_IETF) matches the expected format for telemetry data.
Origin:Path	Review the sensor paths to ensure that they correspond to the specific data sources or paths of interest.

Abbreviations

The following are some key abbreviations and their meanings relevant to this document:

Acronym	Description
JSON	JavaScript Object Notation
RPC	Remote Procedure Call
gNMI	gRPC Network Management Interface
JSON-IETF	JSON-Internet Engineering Task Force

Glossary

The following provides definitions for key terms used throughout this document.

Streaming Telemetry	A monitoring approach that efficiently transmits operational data from OcNOS routers to remote management systems in real-time for analysis, troubleshooting, and network monitoring.
Telemetry Data	Structured operational data generated by routers that is transmitted in real-time to external systems for analysis.
JSON-IETF	JSON-IETF is a data interchange format that follows the specifications defined by the IETF. It is a lightweight, text-based format used for representing structured data. JSON-IETF is commonly used for configuration and data exchange in various network and Internet-related protocols.
Remote Management System	An external system responsible for monitoring, managing, and analyzing data received from network devices.
Network Health	The overall condition and performance of a network, including factors like stability, resource utilization, and data flow.
Resilient Network	A network designed to withstand failures or disruptions, maintaining functionality even in challenging conditions.

CHAPTER 3 Streaming Telemetry Dial-Out Mode

Overview

In OcNOS, dial-out telemetry subscriptions, also known as persistent subscriptions, ensure continuous data streaming, even if the Remote Procedure Call (gRPC) session terminates unexpectedly. With persistent subscriptions, the OcNOS device continuously retries to establish a gRPC connection to the collector server, thus maintaining persistent data streaming.

Feature Characteristics

The dial-out telemetry feature in OcNOS comprises several key aspects ensuring seamless data streaming and connectivity with collector servers:

The described topology outlines a system architecture that utilizes gRPC-based tunneling for persistent streaming telemetry.

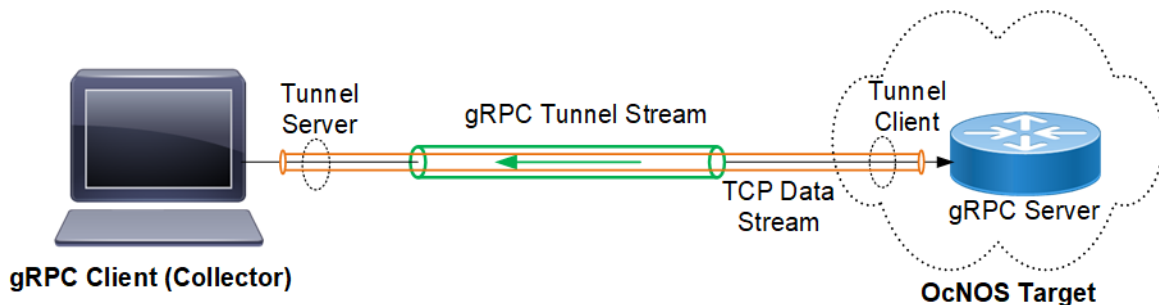


Figure 3-4: Dial-Out Subscription Mode

Here is a detailed explanation of the components and data flow:

- **gNMI Client (gRPC Client):** The gNMI client, which acts as the gRPC client in this scenario, is responsible for handling telemetry data and connecting to the OcNOS target device.
- **Tunnel Server:** The tunnel server, part of the gNMI collector process, listens for incoming gRPC tunnel streams from the gRPC server.
- **gRPC Tunnel Stream:** Represents the persistent communication channel established between the tunnel client (OcNOS) and the tunnel server (collector).
- **Tunnel Client:** The gRPC tunnel client operates on the OcNOS device and connects to the tunnel server. It manages the tunneling of telemetry data.
- **gRPC Server:** Interacts with the tunnel client to establish and manage the tunnel.

Note: Ensure that the tunnel server is reachable over the network from the tunnel client, and configure both the tunnel client and tunnel server with compatible authentication mechanisms.

Data Flow

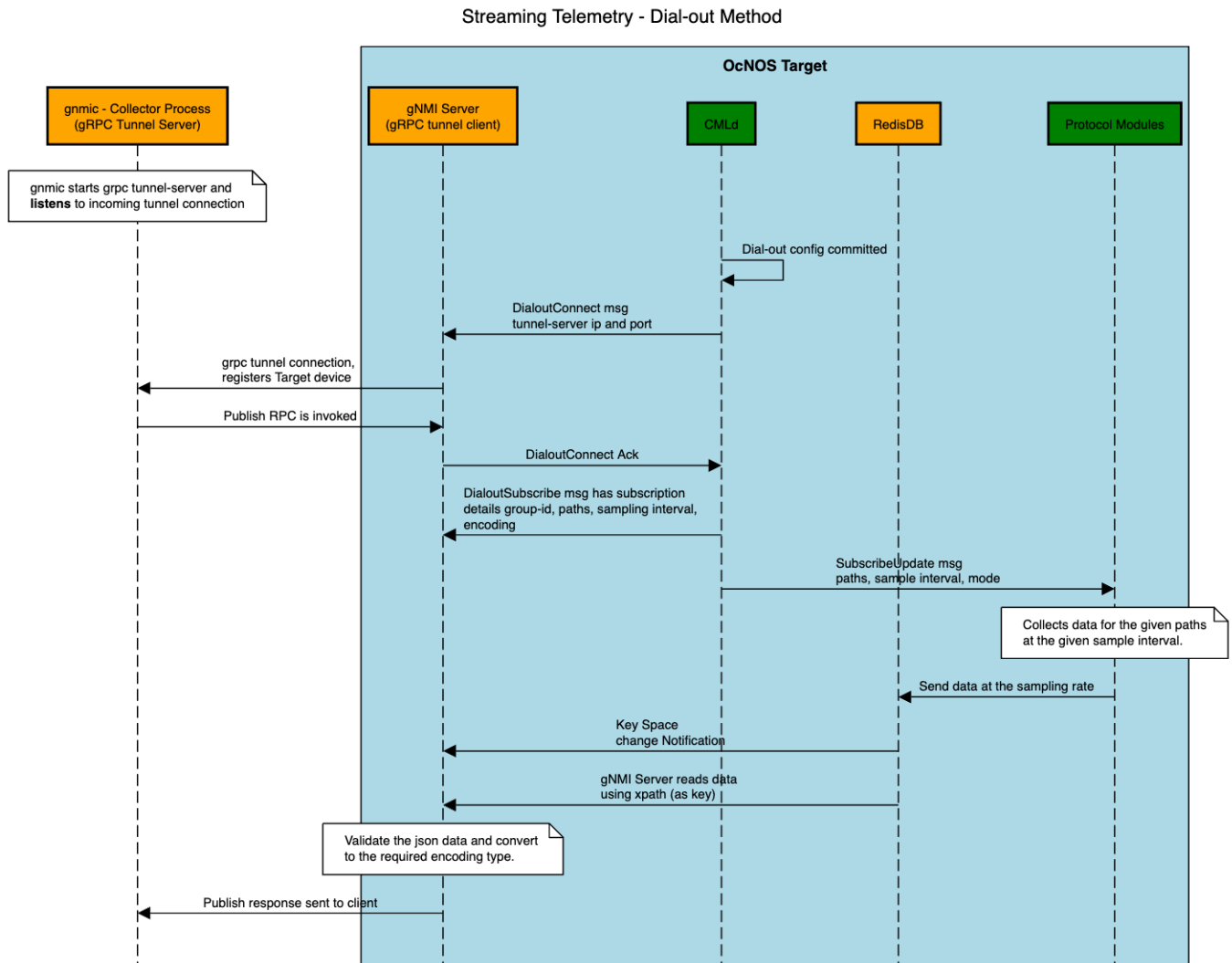
The [Data Flow: Dial-Out Mode](#) flow chart illustrates streaming telemetry in Dial-out Mode.

- **Initialization:** When the dial-out command `subscription-name` is applied successfully, the tunnel client on the OcNOS device initiates a connection to the tunnel server hosted on the collector.
- **Tunnel Establishment:** Upon successful connection, the gRPC client and server establish a persistent tunnel stream. This tunnel facilitates the continuous transmission of telemetry data.

Note: OcNOS supports insecure tunnel connections.

- **Telemetry Data Transmission:** When telemetry data needs to be transmitted from the OcNOS device, the gNMI client sends a Publish RPC request over the established tunnel.
- **Subscription Configuration:** Telemetry commands follow the OpenConfig telemetry model, standardizing the configuration of telemetry subscriptions and related entities.

Figure 3-5: Data Flow: Dial-Out Mode



Benefits

- Ensures continuous data streaming even in the event of gRPC session termination, enhancing network monitoring and troubleshooting capabilities.
- Simplifies configuration and management of telemetry subscriptions using standard OpenConfig models.
- Facilitates secure and reliable communication between the OcNOS device and the collector server.
- Enhances interoperability by enabling integration with third-party gRPC client applications like gNMI client, expanding telemetry options for network operators.

Prerequisites

Before configuring Dial-Out mode, ensure that:

- A supported OcnOS router running a compatible release is required.
- Access to the management interface of the router is necessary.
- Refer to the [gnmic Installation](#) to download the gNMI collector package.

Configuration

Set up the OcnOS router to transmit streaming telemetry data to a gNMI client using the dial-out method.

The sample configuration on the OcnOS router sets up streaming telemetry subscriptions using gNMI to monitor specific paths related to the state of Hard Disk and RAM. The router sends telemetry data to the specified collector (`Collector1`) over a configured tunnel connection. The gNMI client subscribed to these paths will receive updates regarding the state of RAM and Hard Disk at the specified intervals. This setup enables proactive monitoring and management of key hardware components on the network device.

Topology

In this setup, an OcnOS router functions as the data source for streaming telemetry, while a gNMI client acts as the receiver of telemetry data. The OcnOS router sends telemetry data to the gNMI client over a dial-out connection.

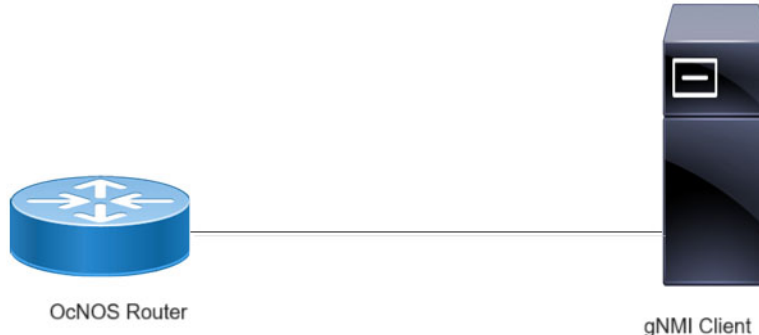


Figure 3-6: Dial-out Streaming Telemetry Topology

Configure OcnOS Router

Note: Before configuring Dial-out, meet all [Prerequisites](#).

1. Enable Streaming Telemetry on an OcnOS router.

```
OcnOS(config)#feature streaming-telemetry
```
2. Create Sensor Group

Create a sensor group (`Platform`) where sensor paths will be specified for dial-out subscriptions. Specify sensor paths within the sensor group (`Platform`) to monitor the state of RAM and Hard Disk.

```
OcnOS(config)#sensor-group Platform
OcnOS(telemetry-sensor-group)#sensor-path ipi:/components/component[name=RAM]/ram/state
```

```
OcNOS (telemetry-sensor-group) #sensor-path ipi:/components/component[name=HARD-DISK]/storage/state
OcNOS (telemetry-sensor-group) #exit
```

3. Create Destination Group

Create a destination group (`Collector1`) where tunnel server settings will be configured for dial-out subscriptions. Specify the tunnel server (`gnmi Client`) IP address (`10.12.101.72`) and port (`11161`) within the destination group (`Collector1`).

```
OcNOS (config) #destination-group Collector1
OcNOS (telemetry-grpc-tunnel-group) #tunnel-server ip 10.12.101.72 port 11161
OcNOS (telemetry-grpc-tunnel-group) #exit
```

4. Create Persistent Subscription

Create a persistent subscription (`storage`), encoding type (`JSON-IETF`), and associate it with the destination group (`Collector1`), and sensor group (`Platform`) to monitor the state of RAM and Hard Disk with a sample interval (`10 seconds`).

```
OcNOS (config) #subscription-name storage
OcNOS (telemetry-subscription) #encoding json-ietf
OcNOS (telemetry-subscription) #destination-group Collector1
OcNOS (telemetry-subscription) #sensor-group Platform sample-interval 10
OcNOS (telemetry-subscription) #exit
```

Streaming Telemetry Snippet Configurations

To verify the telemetry configuration and view the overall commands used for dial-out subscriptions, use the `show running-config streaming-telemetry` command on the router.

```
OcNOS#show running-config streaming-telemetry
!
feature streaming-telemetry
debug telemetry gnmi enable severity debug
!
sensor-group storage
  sensor-path ipi:/components/component[name=RAM]/ram/state
  sensor-path ipi:/components/component[name=HARD-DISK]/storage/state
!
destination-group Collector1
  tunnel-server ip 10.12.101.72 port 11161
!
subscription-name State
  destination-group Collector1
  sensor-group storage sample-interval 10
!
!
```

Telemetry Subscription Invoked via gnmic Command and YAML Input

Start the gNMI collector with the `--use-tunnel-server` and `publish` options to receive the streamed gRPC responses. Execute the following command to start the gRPC tunnel server in listening mode, enabling it to accept incoming connections from gRPC tunnel clients (OcNOS target).

```
./gnmic --insecure --config <path to Tunnel-server yaml file> --use-tunnel-server publish
```

Invoke Publish RPC on OcnOS Target

The following output represents telemetry data published by the `gnmic` command, monitoring the state of Hard Disk and RAM on the specified OcnOS router.

```
# ./gnmic --insecure --config abc.yaml --use-tunnel-server publish
2024/04/12 11:22:50.516313 [gnmic] version=dev, commit=none, date=unknown,
gitURL=, docs=https://gnmic.openconfig.net
2024/04/12 11:22:50.516377 [gnmic] using config file "abc.yaml"
2024/04/12 11:22:50.517770 [gnmic] starting output type file
2024/04/12 11:22:50.517971 [file_output:default-stdout] initialized file
output:
{"Cfg":{"FileName":"","FileType":"stdout","Format":"json","Multiline":true,"In
dent":""
","Separator":"\n","OverrideTimestamps":false,"AddTarget":"","TargetTemplate":
":"","EventProcessors":null,"MsgTemplate":"","ConcurrencyLimit":1000,"EnableMetr
ics":false,"Debug":false}}
2024/04/12 11:22:50.518018 [gnmic] StartPublishCollector is invoked
2024/04/12 11:22:50.518446 [gnmic] Initializing error chan
2024/04/12 11:22:54.508410 [gnmic] tunnel server discovered target
{ID:e8:c5:7a:fe:fd:32 Type:GNMI_GNOI}
2024/04/12 11:22:54.508720 [gnmic] adding target
{"name":"e8:c5:7a:fe:fd:32","address":"e8:c5:7a:fe:fd:32","username":"root","p
assword":"****","timeout":1000000000,"insecure":true,"skip-
verify":false,"buffer-size":100,"retry-timer":1000000000,"log-tls-
secret":false,"gzip":false,"token":"","tunnel-target-type":"GNMI_GNOI"}
2024/04/12 11:22:54.508756 [gnmic] calling publishStream
2024/04/12 11:22:54.508772 [gnmic] publishStream is invoked
2024/04/12 11:22:54.508779 [gnmic] targetPublishStream is invoked
2024/04/12 11:22:54.508830 [gnmic] a.targetsChan: 0xc0004eb1a0
2024/04/12 11:22:54.508840 [gnmic] t.Config.Outputs: []
2024/04/12 11:22:54.508850 [gnmic] starting target "e8:c5:7a:fe:fd:32"
listener
2024/04/12 11:22:54.508879 [gnmic] queuing target "e8:c5:7a:fe:fd:32"
2024/04/12 11:22:54.508902 [gnmic] subscribing to target: "e8:c5:7a:fe:fd:32"
2024/04/12 11:22:54.508918 [gnmic] calling clientPublish
2024/04/12 11:22:54.508930 [gnmic] targetDialOpts: []grpc.DialOption
2024/04/12 11:22:54.508968 [gnmic] a.targetsChan: 0xc0004eb1a0
2024/04/12 11:22:54.508976 [gnmic] t.Config.Outputs: []
2024/04/12 11:22:54.509402 [gnmic] dialing tunnel connection for tunnel target
"e8:c5:7a:fe:fd:32"
Publish Request sent to e8:c5:7a:fe:fd:32{
  "source": "e8:c5:7a:fe:fd:32",
  "subscription-name": "storage",
  "timestamp": 1712920892603436151,
  "time": "2024-04-12T16:51:32.603436151+05:30",
  "updates": [
    {
      "Path": "ipi:components/component[name=HARD-DISK]/storage/state",
      "values": {
        "components/component/storage/state": {
          "free-memory": 0,
          "total-memory": 61057,
          "used-memory": 0
        }
      }
    }
  ]
}
```

```

"source": "e8:c5:7a:fe:fd:32",
"subscription-name": "storage",
"timestamp": 1712920892603253590,
"time": "2024-04-12T16:51:32.60325359+05:30",
"updates": [
  {
    "Path": "ipi:components/component[name=RAM]/ram/state",
    "values": {
      "components/component/ram/state": {
        "available-high-memory": 0,
        "available-memory": 15084,
        "buffers": 101,
        "current-process-count": 227,
        "free-swap": 0,
        "shared-memory": 28,
        "total-high-memory": 0,
        "total-memory": 16010,
        "total-swap": 0,
        "used-memory": 926
      }
    }
  }
]
}

```

The output of the Publish RPC includes the following information:

Publish RPC Output details

Option	Description
source	Displays the MAC address associated with the management port of the target. Each gNMI device have a unique target ID, allowing the collector to distinguish responses between various targets.
subscription-name	The name of the subscription.
timestamp	The timestamp of the response.
time	The timestamp in a human-readable format.
updates	An array of updates, each containing Path and Values.
Path	The path to the published data.
values	The values of the published data.

The telemetry data output includes detailed fields for monitoring the state of the Hard Disk and RAM, offering insights into the memory and storage utilization of the OcNOS router.

1. Hard Disk State

- **Free Memory:** The amount of free memory available on the hard disk.
- **Total Memory:** The total capacity of memory on the hard disk.
- **Used Memory:** The amount of memory currently in use on the hard disk.

2. RAM State

- **Available High Memory:** The available high memory in the RAM.

- **Available Memory:** The total available memory in the RAM.
- **Buffers:** The number of buffer processes running in the RAM.
- **Current Process Count:** The count of active processes in the RAM.
- **Free Swap:** The amount of free swap space in the RAM.
- **Shared Memory:** The shared memory usage in the RAM.
- **Total High Memory:** The total high memory capacity in the RAM.
- **Total Memory:** The total memory capacity in the RAM.
- **Total Swap:** The total swap space available in the RAM.
- **Used Memory:** The amount of memory currently in use in the RAM.

Validation

To verify persistent telemetry configurations and monitor the telemetry data transmission settings on the router, check the output of the `show streaming-telemetry persistent-subscriptions details` command.

```
#show streaming-telemetry persistent-subscriptions details
```

```
Feature streaming telemetry : Enabled
```

```
VRF                               : default
Platform type                     : High range
Maximum sensor-paths              : 100
Minimum sample-interval           : 10
Number of active sensor-paths     : 2 (Dial-In : 0, Dial-out : 2)
Tunnel-server Retry-interval      : Default-60 (seconds)
```

```
Enc-Type      : Encoding type
SI            : Sampling Interval in seconds
Origin:Path   : Sensor Path
```

```
Dial-Out Subscription Details:
```

```
~~~~~
```

```
1. Subscription-name : State
   Status            : ACTIVE
   Enc-Type          : JSON
   Tunnel-server details:
```

```
~~~~~
```

Destination-group	Status	Tunnel-IP:Port
-----	-----	-----
Collector1	IN-ACTIVE	10.12.101.72:11161

```
Sensor-group details:
```

```
~~~~~
```

Sensor-group	SI	Origin:Path
-----	-----	-----
storage	10	ipi:/components/component [name=RAM] /ram/state ipi:/components/component [name=HARD-DISK] /storage/state

Implementation Examples

Real-time Visibility: Operators have real-time visibility into network device health and performance metrics.

Proactive Maintenance: Early detection of issues allows for proactive maintenance and troubleshooting.

Optimized Resource Allocation: Insights from telemetry data help optimize resource allocation and capacity planning.

Enhanced Network Reliability: Continuous monitoring enhances network reliability and reduces downtime.

Dial-Out Commands

The streaming telemetry dial-out mode introduces the following configuration commands.

destination-group

Use this command to create a destination-group for persistent subscriptions on the OcNOS device. The VRF parameter must match the VRF specified in the [feature streaming-telemetry](#) command. Can create and attach multiple destination-groups to activate streaming telemetry subscriptions.

Use the no form of this command to delete a destination-group.

Command Syntax

```
destination-group TUNNEL-NAME (vrf (management|NAME) |)
no destination-group TUNNEL-NAME (vrf (management|NAME) |)
```

Parameters

TUNNEL-NAME	Specify the name assigned to the tunnel server or collector endpoint used for telemetry data transmission.
vrf NAME	(Optional) Creates a destination-group for persistent subscriptions in a user-defined VRF.
vrf management	(Optional) Creates a destination-group for persistent subscriptions in the management VRF.

Default

None

Command Mode

Configure Mode

Applicability

Introduced in OcNOS version 6.5.2.

Example

The following example creates a destination group named `tunnel-1` for transmitting telemetry data.

```
OcNOS(config)#destination-group tunnel-1
OcNOS(telemetry-grpc-tunnel-group)#commit
```

destination-group GRPC

Use this command to add a destination-group under subscriptions. Can create multiple destination-groups within a subscription mode.

Use `no` parameter of this command to remove the destination-groups.

Note: Ensure that the GRPC-GROUP-NAME is configured in the device's configuration mode before adding it to a subscription mode.

Command Syntax

```
destination-group GRPC-GROUP-NAME
no destination-group GRPC-GROUP-NAME
```

Parameters

GRPC-GROUP-NAME	Specify the name assigned to the tunnel server or collector endpoint used for telemetry data transmission.
-----------------	--

Default

None

Command Mode

Telemetry-subscription Mode

Applicability

Introduced in OcNOS version 6.5.2.

Example

Ensure that the GRPC-GROUP-NAME (`tunnel-1`) is already configured in the current configuration mode.

```
OcNOS#configure terminal
OcNOS(config)#show running-config streaming-telemetry
!
feature streaming-telemetry
grpc-tunnel-server retry-interval 60
!
sensor-group stream-1
  sensor-path ipi:/interfaces/interface[name=eth0]/state/counters
!
destination-group tunnel-1
  tunnel-server ip 10.12.66.160 port 11163
!
subscription-name sub-1
  sensor-group stream-1 sample-interval 1000
!
!
```

The following commands illustrates how to add a destination group (`tunnel-1`) under subscription mode (`sub-1`) and verify the configuration using the show command output.

```

OcNOS(config)#subscription-name sub-1
OcNOS (telemetry-subscription)#destination-group tunnel-1
OcNOS (telemetry-subscription)#commit
OcNOS (telemetry-subscription)#exit
OcNOS(config)#show running-config streaming-telemetry
!
feature streaming-telemetry
grpc-tunnel-server retry-interval 60
!
sensor-group stream-1
  sensor-path ipi:/interfaces/interface[name=eth0]/state/counters
!
destination-group tunnel-1
  tunnel-server ip 10.12.66.160 port 11163
!
subscription-name sub-1
  destination-group tunnel-1
  sensor-group stream-1 sample-interval 1000
!
!

```

encoding

Use this command to specify or modify encoding types for subscriptions in streaming telemetry.

Use `no` parameter of this command to remove the encoding option.

Note: Modifying the encoding type is not allowed for active subscriptions.

Command Syntax

```

encoding (json-ietf|json|proto)
no encoding

```

Parameters

<code>json-ietf</code>	Specifies the JSON encoding based on the IETF draft standard.
<code>json</code>	Specifies the default JSON encoding type.
<code>proto</code>	Specifies the Protocol Buffers v3 encoding type.

Default

None

Command Mode

Telemetry-subscription Mode

Applicability

Introduced in OcNOS version 6.5.2.

Example

The following commands demonstrate how to create a telemetry subscription named `sub-3` using the JSON encoding type.

```
OcNOS#configure terminal
OcNOS (config)#subscription-name sub-3
OcNOS (telemetry-subscription)#encoding json
OcNOS (telemetry-subscription)#commit
```

grpc-tunnel-server retry-interval

Use this command to set the interval for retry attempts when establishing a connection for the GNMI server to the tunnel-server. The VRF parameter must match the VRF specified in the [feature streaming-telemetry](#) command.

Use `no` parameter of this command to unset the retry-interval timer.

Command Syntax

```
grpc-tunnel-server retry-interval <30-3000> (vrf (management|NAME) |)
no grpc-tunnel-server retry-interval (vrf (management|NAME) |)
```

Parameters

<code>retry-interval <30-3000></code>	Specifies the duration between retry attempts. The default retry-interval is 60 seconds.
<code>vrf management</code>	(Optional) Sets the retry-interval in the management VRF.
<code>vrf NAME</code>	(Optional) Sets the retry-interval in a user-defined VRF.

Default

None

Command Mode

Configure mode

Applicability

Introduced in the OcNOS version 6.5.2.

Example

The following configuration illustrates how to set the retry-interval timer for the gNMI server to the tunnel-server with a value of 80 seconds.

```
OcNOS#configure terminal
OcNOS (config)#feature streaming-telemetry
OcNOS (config)#grpc-tunnel-server retry-interval 80
OcNOS (config)#commit
```

sensor-group

Use this command to create a sensor group for persistent subscriptions in an OcNOS device. Multiple sensor groups can be created to specify the paths of interest for streaming telemetry. The VRF parameter must match the VRF specified in the [feature streaming-telemetry](#) command. These sensor groups are attached to subscriptions to activate streaming telemetry.

Use `no` parameter of this command to remove a created sensor group.

Command Syntax

```
sensor-group SENSOR-NAME (vrf (management|NAME) |)
no sensor-group SENSOR-NAME (vrf (management|NAME) |)
```

Parameters

<code>SENSOR-NAME</code>	Specifies the name of the sensor group.
<code>vrf</code>	(Optional) Creates a sensor group in the management VRF.
<code>management</code>	
<code>vrf NAME</code>	(Optional) Creates a sensor group in a user-defined VRF.

Default

None

Command Mode

Configure mode

Applicability

Introduced in OcNOS version 6.5.2.

Example

The following commands demonstrate how to create a sensor group named "stream-1" for persistent telemetry subscriptions on an OcNOS device:

```
OcNOS#configure terminal
OcNOS (config)#sensor-group stream-1
OcNOS (telemetry-sensor-group) #commit
OcNOS (telemetry-sensor-group) #exit
```

sensor-group sample-interval

Use this command to to associate a sensor group with a specific sampling interval under subscriptions for activating streaming telemetry. Multiple sensor groups can be created.

Use `no` parameter of this command to remove the sensor-groups from a subscription.

Note: Before adding a `SENSOR-GROUP-NAME` to a subscription, ensure the sensor group is already configured in the configuration mode.

Command Syntax

```
sensor-group SENSOR-GROUP-NAME sample-interval <10-3600>
no sensor-group SENSOR-GROUP-NAME
```

Parameters

SENSOR-GROUP-NAME	Specifies the name of the sensor group to be associated with the subscription.
sample-interval <10-3600>	Defines the sampling interval in seconds for the sensor group. The interval can range from 10 to 3600 seconds.

Default

None

Command Mode

Telemetry-subscription Mode

Applicability

Introduced in OcNOS version 6.5.2.

Example

Ensure that the SENSOR-GROUP-NAME (`stream-1`) is already configured in the current configuration mode.

```
OcNOS#configure terminal
OcNOS(config)#show running-config streaming-telemetry
!
feature streaming-telemetry
grpc-tunnel-server retry-interval 60
!
sensor-group stream-1
  sensor-path ipi:/interfaces/interface[name=eth0]/state/counters
!
subscription-name sub-1
!
!
```

The following commands illustrates how to add a sensor group (`stream-1`) under subscription mode (`sub-1`) and verify the configuration using the show command output.

```
OcNOS(config)#subscription-name sub-1
OcNOS(telemetry-subscription)#sensor-group stream-1 sample-interval 1000
OcNOS(telemetry-subscription)#commit
OcNOS(telemetry-subscription)#exit
OcNOS(config)#show running-config streaming-telemetry
!
feature streaming-telemetry
grpc-tunnel-server retry-interval 60
!
sensor-group stream-1
  sensor-path ipi:/interfaces/interface[name=eth0]/state/counters
!
subscription-name sub-1
  sensor-group stream-1 sample-interval 1000
!
!
```

sensor-path

Use this command to add sensor paths under sensor-groups. Can add multiple sensor paths to a single sensor group. Use `no` parameter of this command to remove sensor paths.

Command Syntax

```
sensor-path SENSOR-PATH
no sensor-path SENSOR-PATH
```

Parameters

`SENSOR-PATH` Specifies the path of the telemetry data to include in the sensor group.

Default

None

Command Mode

Telemetry-sensor-group Mode

Applicability

Introduced in OcNOS version 6.5.2.

Example

The following example demonstrates how to configure a sensor group (`stream-1`) and add multiple sensor paths to it for streaming telemetry.

```
OcNOS#configure terminal
OcNOS (config)#sensor-group stream-1
OcNOS (telemetry-sensor-group)#sensor-path ipi:/interfaces/
interface[name=eth0]/state/counters
OcNOS (telemetry-sensor-group)#sensor-path /interfaces/interface[name=xe2]/
state/counters
OcNOS (telemetry-sensor-group)#sensor-path openconfig:/interfaces/
interface[name=xe3]/state/counters
OcNOS (telemetry-sensor-group)#commit
OcNOS (telemetry-sensor-group)#exit
```

show streaming-telemetry persistent-subscriptions

Use this command to display a brief summary of the streaming-telemetry dial-out configurations. This command provides a concise view of the persistent subscription settings configured on the device.

Command Syntax

```
show streaming-telemetry persistent-subscriptions brief
show streaming-telemetry persistent-subscriptions details (SUBSCRIPTION-NAME|)
```

Parameters

SUBSCRIPTION-NAME Displays detailed configuration information specific to the named persistent subscription.

Default

None

Command Mode

Exec mode

Applicability

Introduced in OcNOS version 6.5.2.

Example

The command output lists each persistent subscription with its associated details.

```
OcNOS#show streaming-telemetry persistent-subscriptions details
```

```
Feature streaming telemetry : Enabled
```

```
VRF : default
Platform type : High range
Maximum sensor-paths : 100
Minimum sample-interval : 10
Number of active sensor-paths : 2 (Dial-In : 0, Dial-out : 2)
Tunnel-server Retry-interval : Default-60 (seconds)
```

```
Enc-Type : Encoding type
SI : Sampling Interval in seconds
Origin:Path : Sensor Path
```

```
Dial-Out Subscription Details:
```

```
~~~~~
```

```
1. Subscription-name : State
   Status : ACTIVE
   Enc-Type : JSON
   Tunnel-server details:
```

```
~~~~~
```

Destination-group	Status	Tunnel-IP:Port
Collector1	IN-ACTIVE	10.12.101.72:11161

```
Sensor-group details:
```

```
~~~~~
```

Sensor-group	SI	Origin:Path
storage	10	ipi:/components/component [name=RAM] /ram/state ipi:/components/component [name=HARD-DISK] /storage/state

The following table explains the output fields.

Field	Description
Feature streaming telemetry	Marked as "Enabled" confirms that streaming telemetry is active on the device.
VRF	Specifies the VRF type.
Platform type	Displays the platform type is standard or high range.
Maximum sensor-paths	Shows the maximum number of sensor paths allowed. For more details, refer to Scale Scenarios section.
Minimum sample-interval	Indicates the minimum sampling interval in seconds. For more details, refer to Scale Scenarios section.
Number of active sensor-paths	Shows the total number of active sensor paths for Dial-In and Dial-Out subscriptions (Stream mode subscriptions).
Tunnel-server Default-Retry-interval	The duration between retry attempts when establishing a connection for the GNMI server to the tunnel server.
Subscription Name	Name of the persistent subscription.
Storage Status or Status	Current status of the subscription (ACTIVE or IN-ACTIVE).
Enc-Type	Encoding type used for telemetry data (JSON, JSON-IETF, Proto).
Destination Group	Define the tunnel server settings to which telemetry data is sent for dial-out subscriptions.
Sensor Group	Sensor group associated with the subscription.
Sample Interval (SI)	Sampling interval for the sensor group.
Tunnel-IP:Port	IP address and port of the tunnel server for dial-out subscriptions.
Origin:Path	The specific sensor paths that are being monitored or streamed by the telemetry system.

subscription-name

Use this command to create named subscriptions for persistent telemetry configurations in an OcNOS device. The VRF parameter must match the VRF specified in the [feature streaming-telemetry](#) command. Multiple subscriptions can be created. These subscriptions are essential for activating streaming telemetry, as they define specific settings such as associated destination groups and sensor groups.

Use `no` parameter of this command to delete a subscription.

Command Syntax

```
subscription-name NAME (vrf (management|NAME) |)
no subscription-name NAME (vrf (management|NAME) |)
```


Parameters

<code>subscription-name NAME</code>	Specifies the unique name to the persistent subscription.
<code>vrf NAME</code>	(Optional) Creates named subscriptions in a user-defined VRF.
<code>vrf management</code>	(Optional) Creates named subscriptions in the management VRF.

Default

None

Command Mode

Configure Mode

Applicability

Introduced in OcNOS version 6.5.2.

Example

The following command demonstrates configuring a subscription (`sub-1`) on an OcNOS device. The subscription remains `in-active` because the sensor groups and destination groups have not been added to it.

```
OcNOS#configure terminal
OcNOS(config)#subscription-name sub-1
OcNOS(telemetry-subscription)#commit
Subscription sub-1 is "in-active": sensor-group(s) and destination-group(s)
are not configured.
OcNOS(telemetry-subscription)#exit
OcNOS(config)#show running-config streaming-telemetry
!
feature streaming-telemetry
!
subscription-name sub-1
!
!
```

tunnel-server

Use this command to add tunnel-servers under destination groups. Can create multiple tunnel servers within a destination group.

Use `no` parameter of this command to remove a tunnel server from the destination group.

Command Syntax

```
tunnel-server ip A.B.C.D port <1-65535>
no tunnel-server ip A.B.C.D port <1-65535>
```

Parameters

<code>ip A.B.C.D</code>	Specifies the tunnel server IP address.
<code>port <1-65535></code>	Specifies the tunnel server port-number.

Default

None

Command Mode

Telemetry-GRPC-tunnel-group Mode

Applicability

Introduced in OcNOS version 6.5.2.

Example

The following command demonstrates how to add a tunnel server within the destination group.

```
OcNOS#configure terminal
OcNOS(config)#destination-group tunnel-1
OcNOS(telemetry-grpc-tunnel-group)#tunnel-server ip 10.12.66.160 port 11163
OcNOS(telemetry-grpc-tunnel-group)#commit
OcNOS(telemetry-grpc-tunnel-group)#exit
```

Revised CLI Commands

The following is the revised command for telemetry.

show techsupport

- The existing syntax now includes the newly added parameter for telemetry, namely `gnmi`.
- The command `show techsupport gnmi` collects gNMI-related information for technical support. For more details, refer to the `show techsupport` command in the Software Monitoring and Reporting chapter in the *System Management Guide*.

Glossary

The following provides definitions for key terms or abbreviations and their meanings used throughout this document:

Key Terms/Acronym	Description
Remote Procedure Call (gRPC)	gRPC protocol that uses HTTP/2 for transport and protocol buffers for serialization.
Persistent Subscription	Telemetry subscription that maintains continuous data streaming even after interruptions in connectivity.

gRPC Network Management Interface (gNMI)	A standardized protocol for network management using gRPC and protocol buffers.
Destination Group	Specifies the collector server's details and connection parameters for telemetry subscriptions.
Sensor Group	Contains sensor paths that define the specific data to be monitored and transmitted.
OpenConfig	Standardized model for network configuration and telemetry using a vendor-neutral approach.

CHAPTER 4 Streaming Telemetry Data Models

IPI Data Models

Streaming telemetry incrementally supports all IPI data models, listed below. Telemetry supports only operational containers and a subset of leaf attributes.

Telemetry IPI Pyang Tree

The Pyang tree output below illustrates the supported containers or leaf, along with a list of supported container-level paths.

ipi-platform

```

+--rw components
  +--ro component* [name]
    +--ro name                               -> ../state/name
    +--ro state
      | +--ro name?                          string
      | +--ro type?                          ipi-platform-types:cmm_component_type_t
      | +--ro location?                       string
      | +--ro mfg-name?                       string
      | +--ro mfg-date?                       yang:date-and-time
      | +--ro description?                   string
      | +--ro hardware-version?              string
      | +--ro firmware-version?              string
      | +--ro software-version?              string
      | +--ro serial-no?                     string
      | +--ro part-no?                       string
      | +--ro removable?                     boolean
      | +--ro oper-status?                   ipi-platform-types:cmm_component_oper_status_t
      | +--ro product-name?                  string
      | +--ro asset-tag?                     string
      | +--ro component-additional-details*  string
      | +--ro parent?                        -> /components/component/state/name
      | +--ro empty?                         boolean
      | +--ro memory
      | | +--ro available?   uint64
      | | +--ro utilized?   uint64
      | | +--ro board-fru
      | | | +--ro board-name?   string
      | | | +--ro board-serial-no? string
      | | | +--ro board-mfg-name? string
      | | | +--ro board-mfg-date? yang:date-and-time
      | +--ro temperature
      | | +--ro instant?       decimal64
      | | +--ro min?           decimal64
      | | +--ro max?           decimal64
      | | +--ro avg?           decimal64
      | | +--ro interval?     uint32
      | | +--ro sensor-name?   string
      | | +--ro sensor-index?  uint8
      | | +--ro alarm-status?  boolean
      | | +--ro alarm-threshold? decimal64
      | | +--ro alarm-severity? cml_alarm_severity_t
      | | +--ro minimum-emergency-temperature? decimal64
      | | +--ro maximum-emergency-temperature? decimal64
      | | +--ro minimum-alert-temperature?   decimal64
      | | +--ro maximum-alert-temperature?   decimal64
      | | +--ro minimum-critical-temperature? decimal64
      | | +--ro maximum-critical-temperature? decimal64
    +--ro bmc-sensor-data-record
      | +--ro state
      | | +--ro sensor-name?   string
  
```

```

|     +--ro value?                ipi-platform-types:cmm_bmc_value_t
|     +--ro units?                string
|     +--ro lower-non-recoverable? decimal64
|     +--ro lower-non-critical?   decimal64
|     +--ro lower-critical?       decimal64
|     +--ro upper-non-critical?   decimal64
|     +--ro upper-critical?       decimal64
|     +--ro upper-non-recoverable? decimal64
|     +--ro operational-status?   string
|     +--ro threshold?            decimal64
|     +--ro event-type?           ipi-platform-types:cmm_bmc_event_type_t
+--ro cpu
| +--ro state
| | +--ro cpu-1min-load-percentage? decimal64
| | +--ro cpu-5min-load-percentage? decimal64
| | +--ro cpu-15min-load-percentage? decimal64
| | +--ro cpu-utilization?         decimal64
+--ro storage
| +--ro state
| | +--ro total-memory?   uint64
| | +--ro used-memory?   uint64
| | +--ro free-memory?   uint64
+--ro ram
| +--ro state
| | +--ro total-memory?   uint64
| | +--ro used-memory?   uint64
| | +--ro available-memory? uint64
| | +--ro shared-memory? uint64
| | +--ro buffers?       uint64
| | +--ro total-swap?    uint64
| | +--ro free-swap?     uint64
| | +--ro current-process-count? uint16
| | +--ro total-high-memory? uint64
| | +--ro available-high-memory? uint64
+--ro transceiver
| +--ro state
| | +--ro grid-spacing?          decimal64
| | +--ro first-frequency?       decimal64
| | +--ro last-frequency?        decimal64
| | +--ro transceiver-temperature? decimal64
| | +--ro transceiver-voltage?   decimal64
+--ro cmis-module
| +--ro eeprom
| | +--ro state
| | | +--ro identifier?          ipi-platform-sff8024-types:cmm_sff8024_identifier_t
| | | +--ro vendor-name?         string
| | | +--ro vendor-oui?         string
| | | +--ro part-number?        string
| | | +--ro revision-level?     string
| | | +--ro serial-number?      string
| | | +--ro manufacturing-date? string
| | | +--ro clei-code?          string
| | | +--ro module-power-class? ipi-platform-cmis-types:cmm_cmis_module_power_class_t
| | | +--ro module-max-power?   decimal64
| | | +--ro cooling-implemented? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | +--ro temperature-max?    int16
| | | +--ro temperature-min?   int16
| | | +--ro operatin-voltage-min? decimal64
| | | +--ro optical-detector?   ipi-platform-cmis-types:cmm_cmis_optical_detector_t
| | | +--ro rx-power-measurement? ipi-platform-cmis-types:cmm_cmis_rx_power_measur_t
| | | +--ro tx-disable-module-wide? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | +--ro cable-assembly-link-length? int16
| | | +--ro connector-type?     ipi-platform-sff8024-types:cmm_sff8024_connector_type_t
| | | +--ro cca-5ghz?           uint8
| | | +--ro cca-7ghz?           uint8
| | | +--ro cca-12p9ghz?        uint8
| | | +--ro cca-25p8ghz?        uint8
| | | +--ro media-interface-technology? ipi-platform-cmis-types:cmm_cmis_media_intf_tech_t
| | | +--ro cmis-revision?      string
| | | +--ro memory-model?       ipi-platform-cmis-types:cmm_cmis_memory_model_t
| | | +--ro mci-max-speed?      ipi-platform-cmis-types:cmm_cmis_mci_max_speed_t
| | | +--ro active-firmware-revision? string
| | | +--ro inactive-firmware-revision? string

```

```

| | | +--ro hardware-revision? string
| | | +--ro media-type? ipi-platform-cmis-types:cmm_cmis_media_type_t
| | | +--ro max-smf-link-length? decimal64
| | | +--ro max-mmfm-om2-link-length? uint8
| | | +--ro max-mmfm-om3-link-length? uint16
| | | +--ro max-mmfm-om4-link-length? uint16
| | | +--ro max-mmfm-om5-link-length? uint16
| | | +--ro wavelength-nominal? decimal64
| | | +--ro wavelength-tolerance? decimal64
| +--ro advertisement
| | +--ro applications
| | | +--ro application* [id]
| | | | +--ro id -> ../state/id
| | | | +--ro state
| | | | | +--ro id? uint8
| | | | +--ro host
| | | | | +--ro state
| | | | | | +--ro interface-type? ipi-platform-cmis-types:cmm_cmis_interface_type_t
| | | | | | +--ro application-bitrate? ipi-platform-cmis-types:cmm_cmis_bit_rate_t
| | | | | | +--ro lane-count? uint8
| | | | | | +--ro signal-bitrate? ipi-platform-cmis-types:cmm_cmis_bit_rate_t
| | | | | | +--ro modulation-format? ipi-platform-cmis-types:cmm_cmis_modulation_format_t
| | | | | | +--ro bits-per-unit-interval? decimal64
| | | | | | +--ro lane-assignment? ipi-platform-cmis-types:cmm_cmis_lane_assignment_t
| | | | +--ro media
| | | | | +--ro state
| | | | | | +--ro interface-type? ipi-platform-cmis-types:cmm_cmis_interface_type_t
| | | | | | +--ro application-bitrate? ipi-platform-cmis-types:cmm_cmis_bit_rate_t
| | | | | | +--ro lane-count? uint8
| | | | | | +--ro signal-bitrate? ipi-platform-cmis-types:cmm_cmis_bit_rate_t
| | | | | | +--ro modulation-format? ipi-platform-cmis-types:cmm_cmis_modulation_format_t
| | | | | | +--ro bits-per-unit-interval? decimal64
| | | | | | +--ro lane-assignment? ipi-platform-cmis-types:cmm_cmis_lane_assignment_t
| | +--ro controls
| | | +--ro state
| | | | +--ro wavelength-control? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | +--ro tunable-transmitter? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | +--ro tx-output-squelching-method? ipi-platform-cmis-types:cmm_cmis_tx_squelch_method_t
| | | | +--ro forced-tx-output-squelching? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | +--ro tx-output-squelching-disable? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | +--ro tx-output-disable? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | +--ro input-polarity-flip-tx? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | +--ro rx-output-squelching-disable? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | +--ro rx-output-disable? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | +--ro output-polarity-flip-rx? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | +--ro diagnostics
| | | +--ro module
| | | | +--ro state
| | | | | +--ro simultaneous-host-and-media-loopback? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | +--ro report-bit-error-ratio? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | +--ro count-bits-and-errors? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | +--ro host
| | | | | +--ro state
| | | | | | +--ro output-loopback? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | +--ro input-loopback? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | +--ro per-lane-loopback? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | +--ro report-input-snr? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | +--ro report-fec? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | +--ro prbs-checker-post-fec? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | +--ro prbs-checker-pre-fec? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | +--ro prbs-checker-types? ipi-platform-cmis-types:cmm_cmis_prbs_support_type_t
| | | | | | +--ro prbs-generator-post-fec? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | +--ro prbs-generator-pre-fec? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | +--ro prbs-generator-types? ipi-platform-cmis-types:cmm_cmis_prbs_support_type_t
| | | | +--ro media
| | | | | +--ro state
| | | | | | +--ro output-loopback? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | +--ro input-loopback? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | +--ro per-lane-loopback? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | +--ro report-input-snr? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | +--ro report-fec? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | +--ro prbs-checker-post-fec? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | +--ro prbs-checker-pre-fec? ipi-platform-cmis-types:cmm_cmis_yes_no_t

```

```

| | | +--ro prbs-checker-types? ipi-platform-cmis-types:cmm_cmis_prbs_support_type_t
| | | +--ro prbs-generator-post-fec? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | +--ro prbs-generator-pre-fec? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | +--ro prbs-generator-types? ipi-platform-cmis-types:cmm_cmis_prbs_support_type_t
| | +--ro durations
| | | +--ro state
| | | | +--ro modsel-wait-time? uint8
| | | | +--ro dpinit-maximum-duration? ipi-platform-cmis-types:cmm_cmis_durations_t
| | | | +--ro dpdeinit-maximum-duration? ipi-platform-cmis-types:cmm_cmis_durations_t
| | | | +--ro dptxtturnon-maximum-duration? ipi-platform-cmis-types:cmm_cmis_durations_t
| | | | +--ro dptxtturnoff-maximum-duration? ipi-platform-cmis-types:cmm_cmis_durations_t
| | | | +--ro modulepwrup-maximum-duration? ipi-platform-cmis-types:cmm_cmis_durations_t
| | | | +--ro modulepwrdown-maximum-duration? ipi-platform-cmis-types:cmm_cmis_durations_t
| | | | +--ro npinit-maximum-duration? ipi-platform-cmis-types:cmm_cmis_durations_t
| | | | +--ro npdeinit-maximum-duration? ipi-platform-cmis-types:cmm_cmis_durations_t
| | | | +--ro nptxtturnon-maximum-duration? ipi-platform-cmis-types:cmm_cmis_durations_t
| | | | +--ro nptxtturnoff-maximum-duration? ipi-platform-cmis-types:cmm_cmis_durations_t
| | +--ro laser
| | | +--ro state
| | | | +--ro supported-grids? ipi-platform-cmis-types:cmm_cmis_laser_grid_support_t
| | | | +--ro fine-tune-supported? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | +--ro fine-tune-resolution? decimal64
| | | | +--ro fine-tune-low-offset? decimal64
| | | | +--ro fine-tune-high-offset? decimal64
| | | | +--ro per-lane-programmable-output-power? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | +--ro minimum-programmable-output-power? decimal64
| | | | +--ro maximum-programmable-output-power? decimal64
| | | +--ro grids
| | | | +--ro grid* [id]
| | | | | +--ro id -> ../state/id
| | | | | +--ro state
| | | | | | +--ro id? ipi-platform-cmis-types:cmm_cmis_laser_grid_spacing_t
| | | | | | +--ro lowest-channel-frequency? decimal64
| | | | | | +--ro highest-channel-frequency? decimal64
| | | | | | +--ro channel-count? uint16
| | +--ro monitoring
| | | +--ro module
| | | | +--ro monitors
| | | | | +--ro monitor* [id]
| | | | | | +--ro id -> ../state/id
| | | | | | +--ro state
| | | | | | | +--ro id? ipi-platform-cmis-types:cmm_cmis_module_monitor_id_t
| | | | | | | +--ro supported? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | +--ro host
| | | | +--ro monitors
| | | | | +--ro monitor* [id]
| | | | | | +--ro id -> ../state/id
| | | | | | | +--ro state
| | | | | | | | +--ro id? ipi-platform-cmis-types:cmm_cmis_host_monitor_id_t
| | | | | | | | +--ro supported? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | | +--ro lanes-assigned? ipi-platform-cmis-types:cmm_cmis_lane_assignment_t
| | | | +--ro flags
| | | | | +--ro flag* [id]
| | | | | | +--ro id -> ../state/id
| | | | | | | +--ro state
| | | | | | | | +--ro id? ipi-platform-cmis-types:cmm_cmis_host_flag_id_t
| | | | | | | | +--ro supported? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | | +--ro lanes-assigned? ipi-platform-cmis-types:cmm_cmis_lane_assignment_t
| | +--ro media
| | | +--ro monitors
| | | | +--ro monitor* [id]
| | | | | +--ro id -> ../state/id
| | | | | | +--ro state
| | | | | | | +--ro id? ipi-platform-cmis-types:cmm_cmis_media_monitor_id_t
| | | | | | | +--ro supported? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | | +--ro lanes-assigned? ipi-platform-cmis-types:cmm_cmis_lane_assignment_t
| | | +--ro flags
| | | | +--ro flag* [id]
| | | | | +--ro id -> ../state/id
| | | | | | +--ro state
| | | | | | | +--ro id? ipi-platform-cmis-types:cmm_cmis_media_flag_id_t
| | | | | | | +--ro supported? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| | | | | | | +--ro lanes-assigned? ipi-platform-cmis-types:cmm_cmis_lane_assignment_t

```

```

| | +--ro pages
| |   +--ro state
| |     +--ro network-path-pages-supported? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| |     +--ro vdm-pages-supported?         ipi-platform-cmis-types:cmm_cmis_yes_no_t
| |     +--ro vdm-groups?                   ipi-platform-cmis-types:cmm_cmis_vdm_pages_support_t
| |     +--ro diagnostics-pages-supported? ipi-platform-cmis-types:cmm_cmis_yes_no_t
| |     +--ro user-page-supported?         ipi-platform-cmis-types:cmm_cmis_yes_no_t
| |     +--ro banks-per-page?              ipi-platform-cmis-types:cmm_cmis_bank_per_page_support_t
| +--ro module-state
| | +--ro state
| | | +--ro fault-state? ipi-platform-cmis-types:cmm_cmis_module_fault_state_t
| | | +--ro current-state? ipi-platform-cmis-types:cmm_cmis_module_state_t
| | +--ro datapaths
| |   +--ro datapath* [lane]
| |     +--ro lane -> ../state/lane
| |     +--ro state
| |       +--ro lane? uint8
| |       +--ro current-state? ipi-platform-cmis-types:cmm_cmis_datapath_states_t
| |       +--ro host-rate? decimal64
| |       +--ro media-rate? decimal64
| |       +--ro interface-name? string
| +--ro module-monitors
| | +--ro monitors
| | | +--ro monitor* [id]
| | |   +--ro id -> ../state/id
| | |   +--ro state
| | |     +--ro id? ipi-platform-cmis-types:cmm_cmis_module_monitor_id_t
| | |     +--ro description? string
| | |     +--ro value? decimal64
| | |     +--ro high-alarm? decimal64
| | |     +--ro high-warning? decimal64
| | |     +--ro low-warning? decimal64
| | |     +--ro low-alarm? decimal64
| | +--ro monitor-alarm
| |   +--ro state
| |     +--ro alarm-id? ipi-platform-cmis-types:cmm_cmis_module_monitor_id_t
| |     +--ro alarm-type? ipi-platform-cmis-types:cmm_cmis_threshold_alarm_t
| |     +--ro current-value? decimal64
| |     +--ro threshold-minimum? decimal64
| |     +--ro threshold-maximum? decimal64
| +--ro host-monitors
| | +--ro lanes
| |   +--ro lane* [number]
| |     +--ro number -> ../state/number
| |     +--ro state
| |       +--ro number? uint8
| |       +--ro dp-assigned? boolean
| |     +--ro monitors
| |       +--ro monitor* [id]
| |         +--ro id -> ../state/id
| |         +--ro state
| |           +--ro id? ipi-platform-cmis-types:cmm_cmis_host_monitor_id_t
| |           +--ro description? string
| |           +--ro value? decimal64
| |           +--ro high-alarm? decimal64
| |           +--ro high-warning? decimal64
| |           +--ro low-warning? decimal64
| |           +--ro low-alarm? decimal64
| |     +--ro flags
| |       +--ro flag* [id]
| |         +--ro id -> ../state/id
| |         +--ro state
| |           +--ro id? ipi-platform-cmis-types:cmm_cmis_host_flag_id_t
| |           +--ro description? string
| |           +--ro value? boolean
| |     +--ro monitor-alarm
| |       +--ro state
| |         +--ro alarm-id? ipi-platform-cmis-types:cmm_cmis_host_monitor_id_t
| |         +--ro alarm-type? ipi-platform-cmis-types:cmm_cmis_threshold_alarm_t
| |         +--ro current-value? decimal64
| |         +--ro threshold-minimum? decimal64
| |         +--ro threshold-maximum? decimal64
| |     +--ro flag-alarm

```



```

|         +---ro state
|         |         +---ro alarm-id?   ipi-platform-cmis-types:cmm_cmis_host_flag_id_t
+---ro media-monitors
|         +---ro lanes
|         |         +---ro lane* [number]
|         |         |         +---ro number           -> ../state/number
|         |         |         +---ro state
|         |         |         |         +---ro number?   uint8
|         |         |         +---ro monitors
|         |         |         |         +---ro monitor* [id]
|         |         |         |         |         +---ro id           -> ../state/id
|         |         |         |         |         +---ro state
|         |         |         |         |         |         +---ro id?           ipi-platform-cmis-types:cmm_cmis_media_monitor_id_t
|         |         |         |         |         |         +---ro description?   string
|         |         |         |         |         |         +---ro value?         decimal64
|         |         |         |         |         |         +---ro high-alarm?     decimal64
|         |         |         |         |         |         +---ro high-warning?   decimal64
|         |         |         |         |         |         +---ro low-warning?    decimal64
|         |         |         |         |         |         +---ro low-alarm?     decimal64
|         |         |         +---ro flags
|         |         |         |         +---ro flag* [id]
|         |         |         |         |         +---ro id           -> ../state/id
|         |         |         |         |         +---ro state
|         |         |         |         |         |         +---ro id?           ipi-platform-cmis-types:cmm_cmis_media_flag_id_t
|         |         |         |         |         |         +---ro description?   string
|         |         |         |         |         |         +---ro value?         boolean
+---ro monitor-alarm
|         |         |         |         +---ro state
|         |         |         |         |         +---ro alarm-id?       ipi-platform-cmis-types:cmm_cmis_media_monitor_id_t
|         |         |         |         |         +---ro alarm-type?     ipi-platform-cmis-types:cmm_cmis_threshold_alarm_t
|         |         |         |         |         +---ro current-value?    decimal64
|         |         |         |         |         +---ro threshold-minimum?  decimal64
|         |         |         |         |         +---ro threshold-maximum?  decimal64
+---ro flag-alarm
|         |         |         +---ro state
|         |         |         |         +---ro alarm-id?   ipi-platform-cmis-types:cmm_cmis_media_flag_id_t
+---ro power-supply
|         +---ro state
|         |         +---ro operational-status?   cml_cmm_power_supply_operstatus_t
|         |         +---ro capacity?            decimal64
|         |         +---ro power-consumption?   decimal64
|         |         +---ro input-power?        decimal64
|         |         +---ro input-voltage?      decimal64
|         |         +---ro output-voltage?     decimal64
|         |         +---ro input-current?      decimal64
|         |         +---ro output-current?     decimal64
|         |         +---ro temperature-sensor1? decimal64
|         |         +---ro temperature-sensor2? decimal64
|         |         +---ro temperature-sensor3? decimal64
|         |         +---ro fan1-rpm?          uint32
|         |         +---ro fan2-rpm?          uint32
|         |         +---ro fan3-rpm?          uint32
|         |         +---ro fan4-rpm?          uint32
+---ro fan
|         +---ro state
|         |         +---ro rpm?                uint32
|         |         +---ro fan-status?        cml_cmm_fan_status_t
|         |         +---ro fan-location?      cml_cmm_fan_location_t
+---ro fan-tray
|         +---ro state
|         |         +---ro status?           cml_cmm_fan_tray_status_t

```

ipi-interface

```

+---rw interfaces
+---rw interface* [name]
|         +---rw name           -> ../config/name
+---rw config
|         +---rw name?         string
+---ro state
|         +---ro ifindex?      uint32
|         +---ro admin-status? ipi-if-types:if_interface_admin_status_t
|         +---ro oper-status?  ipi-if-types:if_interface_oper_status_t

```

```

| +--ro last-change?      yang:timeticks
| +--ro logical?         boolean
| +--ro counters
|   +--ro in-octets?     yang:counter64
|   +--ro in-pkts?      yang:counter64
|   +--ro in-unicast-pkts? yang:counter64
|   +--ro in-broadcast-pkts? yang:counter64
|   +--ro in-multicast-pkts? yang:counter64
|   +--ro in-discards?   yang:counter64
|   +--ro in-errors?    yang:counter64
|   +--ro in-fcs-errors? yang:counter64
|   +--ro out-octets?    yang:counter64
|   +--ro out-pkts?     yang:counter64
|   +--ro out-unicast-pkts? yang:counter64
|   +--ro out-broadcast-pkts? yang:counter64
|   +--ro out-multicast-pkts? yang:counter64
|   +--ro out-discards?  yang:counter64
|   +--ro out-errors?   yang:counter64
|   +--ro last-clear?   ipi-if-types:if_last_clear_time_t
+--rw ethernet
  +--ro state
    +--ro counters
      +--ro extended-ethernet-counters
        +--ro receive-pkt-rate? yang:counter64
        +--ro receive-bit-rate? yang:counter64
        +--ro send-pkt-rate?    yang:counter64
        +--ro send-bit-rate?    yang:counter64

```

ipi-vxlan

```

+--rw vxlan
  +--ro tunnel* [destination-vtep-ip]
    +--ro destination-vtep-ip -> ../state/destination-vtep-ip
    +--ro state
      +--ro destination-vtep-ip? inet:ipv4-address
      +--ro counters
        +--ro out-bytes?      yang:counter64
        +--ro out-packets?   yang:counter64
        +--ro in-bytes?      yang:counter64
        +--ro in-packets?    yang:counter64

```

Container Level Sensor Paths and Leaf Attributes

The below section lists the container level sensor paths and leaf attributes supported for IPI data models.

ipi-interface

Interface State

Sensor Path
 ipi:/interfaces/interface[name]/state

Leaf Attributes
 /interfaces/interface[name]/state/name
 /interfaces/interface[name]/state/ifindex
 /interfaces/interface[name]/state/admin-status
 /interfaces/interface[name]/state/oper-status
 /interfaces/interface[name]/state/last-change
 /interfaces/interface[name]/state/logical

Interface Counters

Sensor Path
 ipi:/interfaces/interface[name]/state/counters

Leaf Attributes

```

/interfaces/interface[name]/state/counters/in-octets
/interfaces/interface[name]/state/counters/in-pkts
/interfaces/interface[name]/state/counters/in-unicast-pkts
/interfaces/interface[name]/state/counters/in-broadcast-pkts
/interfaces/interface[name]/state/counters/in-multicast-pkts
/interfaces/interface[name]/state/counters/in-discards
/interfaces/interface[name]/state/counters/in-errors
/interfaces/interface[name]/state/counters/in-fcs-errors
/interfaces/interface[name]/state/counters/out-octets
/interfaces/interface[name]/state/counters/out-pkts
/interfaces/interface[name]/state/counters/out-unicast-pkts
/interfaces/interface[name]/state/counters/out-broadcast-pkts
/interfaces/interface[name]/state/counters/out-multicast-pkts
/interfaces/interface[name]/state/counters/out-discards
/interfaces/interface[name]/state/counters/out-errors
/interfaces/interface[name]/state/counters/last-clear

```

Extended Ethernet Counters

Sensor Path

```

ipi:/interfaces/interface[name]/ethernet/state/counters/extended-ethernet-counters

```

Leaf Attributes

```

/interfaces/interface[name]/ethernet/state/counters/extended-ethernet-counters/receive-pkt-rate
/interfaces/interface[name]/ethernet/state/counters/extended-ethernet-counters/send-pkt-rate
/interfaces/interface[name]/ethernet/state/counters/extended-ethernet-counters/receive-bit-rate
/interfaces/interface[name]/ethernet/state/counters/extended-ethernet-counters/send-bit-rate

```

ipi-platform

The paths listed below represent telemetry paths for monitoring the state of various components, including CPU, storage, RAM, power supply, fans, fan trays, CMIS, and transceivers.

CPU

Sensor Path

```

ipi:/components/component[name]/cpu/state

```

Leaf Attributes

```

/components/component[name]/cpu/state/cpu-1min-load-percentage
/components/component[name]/cpu/state/cpu-5min-load-percentage
/components/component[name]/cpu/state/cpu-15min-load-percentage
/components/component[name]/cpu/state/cpu-utilization

```

Storage

Sensor Path

```

ipi:/components/component[name]/storage/state/

```

Leaf Attributes

```

/components/component[name]/storage/state/total-memory
/components/component[name]/storage/state/used-memory
/components/component[name]/storage/state/free-memory

```

RAM

Sensor Path

```
ipi:/components/component[name]/ram/state/
```

Leaf Attributes

```
/components/component[name]/ram/state/total-memory  
/components/component[name]/ram/state/used-memory  
/components/component[name]/ram/state/available-memory  
/components/component[name]/ram/state/shared-memory  
/components/component[name]/ram/state/buffers  
/components/component[name]/ram/state/total-swap  
/components/component[name]/ram/state/free-swap  
/components/component[name]/ram/state/current-process-count  
/components/component[name]/ram/state/total-high-memory  
/components/component[name]/ram/state/available-high-memory
```

Power-Supply

Sensor Path

```
ipi:/components/component[name]/power-supply/state/
```

Leaf Attributes

```
/components/component[name]/power-supply/state/capacity  
/components/component[name]/power-supply/state/power-consumption  
/components/component[name]/power-supply/state/input-power  
/components/component[name]/power-supply/state/input-voltage  
/components/component[name]/power-supply/state/input-current  
/components/component[name]/power-supply/state/output-voltage  
/components/component[name]/power-supply/state/output-current  
/components/component[name]/power-supply/state/operational-status  
/components/component[name]/power-supply/state/fan1-rpm  
/components/component[name]/power-supply/state/fan2-rpm  
/components/component[name]/power-supply/state/fan3-rpm  
/components/component[name]/power-supply/state/fan4-rpm  
/components/component[name]/power-supply/state/temperature-sensor1  
/components/component[name]/power-supply/state/temperature-sensor2  
/components/component[name]/power-supply/state/temperature-sensor3
```

Fan

Sensor Path

```
ipi:/components/component[name]/fan/state/
```

Leaf Attributes

```
/components/component[name]/fan/state/rpm  
/components/component[name]/fan/state/fan-status  
/components/component[name]/fan/state/fan-location
```

Fan-Tray

Sensor Path

```
ipi:/components/component[name]/fan-tray/state/
```

Leaf Attributes

```
/components/component[name]/fan-tray/state/status
```

Transceiver

Sensor Path

```
ipi:/components/component[name]/transceiver/state/
```

Leaf Attributes

```

    /components/component[name]/transceiver/state/grid-spacing
    /components/component[name]/transceiver/state/first-frequency
    /components/component[name]/transceiver/state/last-frequency
    /components/component[name]/transceiver/state/transceiver-
temperature
    /components/component[name]/transceiver/state/transceiver-voltage
```

Platform State

Sensor Path

```
ipi:/components/component[name]/state/
```

Leaf Attributes

```

    /components/component[name]/state/name
    /components/component[name]/state/type
    /components/component[name]/state/location
    /components/component[name]/state/mfg-name
    /components/component[name]/state/description
    /components/component[name]/state/hardware-version
    /components/component[name]/state/firmware-version
    /components/component[name]/state/software-version
    /components/component[name]/state/serial-no
    /components/component[name]/state/part-no
    /components/component[name]/state/removable
    /components/component[name]/state/oper-status
    /components/component[name]/state/product-name
    /components/component[name]/state/asset-tag
    /components/component[name]/state/component-additional-details
    /components/component[name]/state/parent
    /components/component[name]/state/empty
```

Sensor Path

```
ipi:/components/component[name]/state/memory
```

Leaf Attributes

```

    /components/component[name]/state/memory/available
    /components/component[name]/state/memory/utilized
```

Sensor Path

```
ipi:/components/component[name]/state/board-fru
```

Leaf Attributes

```

    /components/component[name]/state/board-fru/board-name
    /components/component[name]/state/board-fru/board-serial-no
    /components/component[name]/state/board-fru/board-mfg-name
    /components/component[name]/state/board-fru/board-mfg-date
```

Sensor Path

```
ipi:/components/component[name]/state/temperature
```

Leaf Attributes

```

    /components/component[name]/state/temperature/instant
    /components/component[name]/state/temperature/min
    /components/component[name]/state/temperature/max
    /components/component[name]/state/temperature/avg
    /components/component[name]/state/temperature/interval
    /components/component[name]/state/temperature/sensor-name
```

```

    /components/component[name]/state/temperature/sensor-index
    /components/component[name]/state/temperature/alarm-status
    /components/component[name]/state/temperature/alarm-threshold
    /components/component[name]/state/temperature/alarm-severity
temperature /components/component[name]/state/temperature/minimum-emergency-
temperature /components/component[name]/state/temperature/maximum-emergency-
temperature /components/component[name]/state/temperature/minimum-alert-
temperature /components/component[name]/state/temperature/maximum-alert-
temperature /components/component[name]/state/temperature/minimum-critical-
temperature /components/component[name]/state/temperature/maximum-critical-
temperature

```

ipi-vxlan

VXLAN State

Sensor Path

```
ipi:/vxlan/tunnel[destination-vtep-ip]/state/counters
```

Leaf Attributes

```

    /vxlan/tunnel[destination-vtep-ip]/state/counters/out-bytes
    /vxlan/tunnel[destination-vtep-ip]/state/counters/out-packets
    /vxlan/tunnel[destination-vtep-ip]/state/counters/in-bytes
    /vxlan/tunnel[destination-vtep-ip]/state/counters/in-packets

```

ipi-cmis

Transceiver EEPROM State

Sensor Path

```
ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmis-
module/eeprom/state
```

Leaf Attributes

```

    /components/component[name]/transceiver/cmis-module/eeprom/state/identifier
    /components/component[name]/transceiver/cmis-module/eeprom/state/vendor-name
    /components/component[name]/transceiver/cmis-module/eeprom/state/vendor-oui
    /components/component[name]/transceiver/cmis-module/eeprom/state/part-number
    /components/component[name]/transceiver/cmis-module/eeprom/state/revision-level
    /components/component[name]/transceiver/cmis-module/eeprom/state/serial-number
    /components/component[name]/transceiver/cmis-module/eeprom/state/manufacturing-date
    /components/component[name]/transceiver/cmis-module/eeprom/state/clei-code
    /components/component[name]/transceiver/cmis-module/eeprom/state/module-power-class
    /components/component[name]/transceiver/cmis-module/eeprom/state/module-max-power
    /components/component[name]/transceiver/cmis-module/eeprom/state/cooling-implemented
    /components/component[name]/transceiver/cmis-module/eeprom/state/temperature-max
    /components/component[name]/transceiver/cmis-module/eeprom/state/temperature-min
    /components/component[name]/transceiver/cmis-module/eeprom/state/operatin-voltage-min
    /components/component[name]/transceiver/cmis-module/eeprom/state/optical-detector
    /components/component[name]/transceiver/cmis-module/eeprom/state/rx-power-measurement
    /components/component[name]/transceiver/cmis-module/eeprom/state/tx-disable-module-wide
    /components/component[name]/transceiver/cmis-module/eeprom/state/cable-assembly-link-length
    /components/component[name]/transceiver/cmis-module/eeprom/state/connector-type
    /components/component[name]/transceiver/cmis-module/eeprom/state/cca-5ghz
    /components/component[name]/transceiver/cmis-module/eeprom/state/cca-7ghz
    /components/component[name]/transceiver/cmis-module/eeprom/state/cca-12p9ghz
    /components/component[name]/transceiver/cmis-module/eeprom/state/cca-25p8ghz
    /components/component[name]/transceiver/cmis-module/eeprom/state/media-interface-technology
    /components/component[name]/transceiver/cmis-module/eeprom/state/cmis-revision
    /components/component[name]/transceiver/cmis-module/eeprom/state/memory-model

```

```

/components/component[name]/transceiver/cm1s-module/eepr0m/state/mci-max-speed
/components/component[name]/transceiver/cm1s-module/eepr0m/state/active-firmvare-revision
/components/component[name]/transceiver/cm1s-module/eepr0m/state/inactive-firmvare-revision
/components/component[name]/transceiver/cm1s-module/eepr0m/state/hardware-revision
/components/component[name]/transceiver/cm1s-module/eepr0m/state/media-type
/components/component[name]/transceiver/cm1s-module/eepr0m/state/max-smf-link-length
/components/component[name]/transceiver/cm1s-module/eepr0m/state/max-mmF-om2-link-length
/components/component[name]/transceiver/cm1s-module/eepr0m/state/max-mmF-om3-link-length
/components/component[name]/transceiver/cm1s-module/eepr0m/state/max-mmF-om4-link-length
/components/component[name]/transceiver/cm1s-module/eepr0m/state/max-mmF-om5-link-length
/components/component[name]/transceiver/cm1s-module/eepr0m/state/wavelength-nominal
/components/component[name]/transceiver/cm1s-module/eepr0m/state/wavelength-tolerance

```

CMIS State

Sensor Path

```
ipi:/components/component[name=CMIS-MODULE-NAME]/state
```

Leaf Attributes

```

/components/component[name]/state/name
/components/component[name]/state/type
/components/component[name]/state/location
/components/component[name]/state/description
/components/component[name]/state/mfg-name
/components/component[name]/state/mfg-date
/components/component[name]/state/hardware-version
/components/component[name]/state/firmvare-version
/components/component[name]/state/serial-no
/components/component[name]/state/part-no
/components/component[name]/state/removable
/components/component[name]/state/parent

```

Transceiver Advertisement Control

Sensor Path

```
ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cm1s-
module/advertisement/controls/state
```

Leaf Attributes

```

/components/component[name]/transceiver/cm1s-module/advertisement/controls/state/forced-tx-output-squelching
/components/component[name]/transceiver/cm1s-module/advertisement/controls/state/input-polarity-flip-tx
/components/component[name]/transceiver/cm1s-module/advertisement/controls/state/output-polarity-flip-rx
/components/component[name]/transceiver/cm1s-module/advertisement/controls/state/rx-output-disable
/components/component[name]/transceiver/cm1s-module/advertisement/controls/state/rx-output-squelching-disable
/components/component[name]/transceiver/cm1s-module/advertisement/controls/state/tunable-transmitter
/components/component[name]/transceiver/cm1s-module/advertisement/controls/state/tx-output-disable
/components/component[name]/transceiver/cm1s-module/advertisement/controls/state/tx-output-squelching-disable
/components/component[name]/transceiver/cm1s-module/advertisement/controls/state/tx-output-squelching-method
/components/component[name]/transceiver/cm1s-module/advertisement/controls/state/wavelength-control

```

Transceiver Advertisement Diagnostics Module

Sensor Path

```
ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cm1s-module/advertisement/diagnostics/module/state
```

Leaf Attributes

```

/components/component[name]/transceiver/cm1s-module/advertisement/diagnostics/module/state/count-bits-and-errors
/components/component[name]/transceiver/cm1s-module/advertisement/diagnostics/module/state/report-bit-error-
ratio
/components/component[name]/transceiver/cm1s-module/advertisement/diagnostics/module/state/simultaneous-host-
and-media-loopback

```

Transceiver Advertisement Diagnostics Host

Sensor Path

```
ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cm1s-module/advertisement/diagnostics/host/state
```

Leaf Attributes

```

/components/component[name]/transceiver/cmisis-module/advertisement/diagnostics/host/state/input-loopack
/components/component[name]/transceiver/cmisis-module/advertisement/diagnostics/host/state/output-loopack
/components/component[name]/transceiver/cmisis-module/advertisement/diagnostics/host/state/per-lane-loopack
/components/component[name]/transceiver/cmisis-module/advertisement/diagnostics/host/state/prbs-checker-post-fec
/components/component[name]/transceiver/cmisis-module/advertisement/diagnostics/host/state/prbs-checker-pre-fec
/components/component[name]/transceiver/cmisis-module/advertisement/diagnostics/host/state/prbs-generator-post-fec
/components/component[name]/transceiver/cmisis-module/advertisement/diagnostics/host/state/prbs-generator-pre-fec
/components/component[name]/transceiver/cmisis-module/advertisement/diagnostics/host/state/report-fec
/components/component[name]/transceiver/cmisis-module/advertisement/diagnostics/host/state/report-input-snr

```

Transceiver Advertisement Diagnostics Media

Sensor Path

```

ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/advertisement/diagnostics/media/state

```

Leaf Attributes

```

/components/component[name]/transceiver/cmisis-module/advertisement/diagnostics/media/state/input-loopack
/components/component[name]/transceiver/cmisis-module/advertisement/diagnostics/media/state/output-loopack
/components/component[name]/transceiver/cmisis-module/advertisement/diagnostics/media/state/per-lane-loopack
/components/component[name]/transceiver/cmisis-module/advertisement/diagnostics/media/state/prbs-checker-post-fec
/components/component[name]/transceiver/cmisis-module/advertisement/diagnostics/media/state/prbs-checker-pre-fec
fec
/components/component[name]/transceiver/cmisis-module/advertisement/diagnostics/media/state/prbs-generator-post-fec
/components/component[name]/transceiver/cmisis-module/advertisement/diagnostics/media/state/prbs-generator-pre-fec
/components/component[name]/transceiver/cmisis-module/advertisement/diagnostics/media/state/report-fec
/components/component[name]/transceiver/cmisis-module/advertisement/diagnostics/media/state/report-input-snr

```

Transceiver Advertisement Duration

Sensor Path

```

ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/advertisement/durations/state

```

Leaf Attributes

```

/components/component[name]/transceiver/cmisis-module/advertisement/durations/state/dpdeinit-maximum-duration
/components/component[name]/transceiver/cmisis-module/advertisement/durations/state/dpinit-maximum-duration
/components/component[name]/transceiver/cmisis-module/advertisement/durations/state/dptxturnoff-maximum-duration
/components/component[name]/transceiver/cmisis-module/advertisement/durations/state/dptxturnon-maximum-duration
/components/component[name]/transceiver/cmisis-module/advertisement/durations/state/modsel-wait-time
/components/component[name]/transceiver/cmisis-module/advertisement/durations/state/modulepwrdown-maximum-duration
/components/component[name]/transceiver/cmisis-module/advertisement/durations/state/modulepwrup-maximum-duration

```

Transceiver Advertisement Laser

Sensor Path

```

ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/advertisement/laser/state

```

Leaf Attributes

```

/components/component[name]/transceiver/cmisis-module/advertisement/laser/state/fine-tune-high-offse
/components/component[name]/transceiver/cmisis-module/advertisement/laser/state/fine-tune-low-offset
/components/component[name]/transceiver/cmisis-module/advertisement/laser/state/fine-tune-resolution
/components/component[name]/transceiver/cmisis-module/advertisement/laser/state/fine-tune-supported
/components/component[name]/transceiver/cmisis-module/advertisement/laser/state/maximum-programmable-output-power
/components/component[name]/transceiver/cmisis-module/advertisement/laser/state/minimum-programmable-output-power
/components/component[name]/transceiver/cmisis-module/advertisement/laser/state/per-lane-programmable-output-power

```

Transceiver Advertisement Laser Grid

Sensor Path

```

ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/advertisement/laser/grids/grid[id=SPACING
TYPE]/state

```

Leaf Attributes

```

/components/component[name]/transceiver/cmisis-module/advertisement/laser/grids/grid/state/channel-count
/components/component[name]/transceiver/cmisis-module/advertisement/laser/grids/grid/state/highest-channel-
frequency
/components/component[name]/transceiver/cmisis-module/advertisement/laser/grids/grid/state/id
/components/component[name]/transceiver/cmisis-module/advertisement/laser/grids/grid/state/lowest-channel-
frequency

```


Transceiver Advertisement Monitoring

Sensor Path

```
ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/advertisement/monitoring/module/monitors/monitor[id=APP ID]/state
```

Leaf Attributes

```
/components/component[name]/transceiver/cmisis-module/advertisement/monitoring/module/monitors/monitor/state/id
supported
/components/component[name]/transceiver/cmisis-module/advertisement/monitoring/module/monitors/monitor/state/
```

Transceiver Advertisement Host Monitoring

Sensor Path

```
ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/advertisement/monitoring/host/monitors/monitor[id= APP ID]/state
```

Leaf Attributes

```
/components/component[name]/transceiver/cmisis-module/advertisement/monitoring/host/monitors/monitor/state/id
supported
/components/component[name]/transceiver/cmisis-module/advertisement/monitoring/host/monitors/monitor/state/
```

Transceiver Advertisement Host Flags Monitoring

Sensor Path

```
ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/advertisement/monitoring/host/flags/flag[id=APP ID]/state
```

Leaf Attributes

```
/components/component[name]/transceiver/cmisis-module/advertisement/monitoring/host/flags/flag/state/id
/components/component[name]/transceiver/cmisis-module/advertisement/monitoring/host/flags/flag/state/supported
```

Transceiver Advertisement Media Monitoring

Sensor Path

```
ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/advertisement/monitoring/media/monitors/monitor[id=APP ID]/state
```

Leaf Attributes

```
/components/component[name]/transceiver/cmisis-module/advertisement/monitoring/media/monitors/monitor/state/id
supported
/components/component[name]/transceiver/cmisis-module/advertisement/monitoring/media/monitors/monitor/state/
```

Transceiver Advertisement Media Flags Monitoring

Sensor Path

```
ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/advertisement/monitoring/media/flags/flag[id = APP ID]/state
```

Leaf Attributes

```
/components/component[name]/transceiver/cmisis-module/advertisement/monitoring/media/flags/flag/state/id
/components/component[name]/transceiver/cmisis-module/advertisement/monitoring/media/flags/flag/state/supported
```

Transceiver Advertisement Pages

Sensor Path

```
ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/advertisement/pages/state
```

Leaf Attributes

```
/components/component[name]/transceiver/cmisis-module/advertisement/pages/state/banks-per-page
/components/component[name]/transceiver/cmisis-module/advertisement/pages/state/network-path-pages-supported
/components/component[name]/transceiver/cmisis-module/advertisement/pages/state/user-page-supported
/components/component[name]/transceiver/cmisis-module/advertisement/pages/state/vdm-groups
/components/component[name]/transceiver/cmisis-module/advertisement/pages/state/vdm-pages-supported
```

Transceiver Advertisement Host Application

Sensor Path

```
ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/advertisement/applications/application[id=APP_ID]/host/state
```

Leaf Attributes

```

/components/component[name]/transceiver/cmismodule/advertisement/applications/application[id]/host/state/
interface-type
/components/component[name]/transceiver/cmismodule/advertisement/applications/application[id]/host/state/
application-bitrate
/components/component[name]/transceiver/cmismodule/advertisement/applications/application[id]/host/state/lane-
count
/components/component[name]/transceiver/cmismodule/advertisement/applications/application[id]/host/state/
signal-bitrate
/components/component[name]/transceiver/cmismodule/advertisement/applications/application[id]/host/state/
modulation-format
/components/component[name]/transceiver/cmismodule/advertisement/applications/application[id]/host/state/bits-
per-unit-interval
/components/component[name]/transceiver/cmismodule/advertisement/applications/application[id]/host/state/lane-
assignment

```

Transceiver Advertisement Media Application

Sensor Path

```

ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmismodule/advertisement/applications/
application[id=APP_ID]/media/state

```

Leaf Attributes

```

/components/component[name]/transceiver/cmismodule/advertisement/applications/application[id]/media/state/
interface-type
/components/component[name]/transceiver/cmismodule/advertisement/applications/application[id]/media/state/
application-bitrate
/components/component[name]/transceiver/cmismodule/advertisement/applications/application[id]/media/state/lane-
count
/components/component[name]/transceiver/cmismodule/advertisement/applications/application[id]/media/state/
signal-bitrate
/components/component[name]/transceiver/cmismodule/advertisement/applications/application[id]/media/state/
modulation-format
/components/component[name]/transceiver/cmismodule/advertisement/applications/application[id]/media/state/bits-
per-unit-interval
/components/component[name]/transceiver/cmismodule/advertisement/applications/application[id]/media/state/lane-
assignment

```

Transceiver CMIS Module State

Sensor Path

```

ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmismodule/module-state/datapaths/
datapath[lane=LANE_ID]/state

```

Leaf Attributes

```

/components/component[name]/transceiver/cmismodule/module-state/datapaths/datapath[lane]/state/current-state
/components/component[name]/transceiver/cmismodule/module-state/datapaths/datapath[lane]/state/host-rate
/components/component[name]/transceiver/cmismodule/module-state/datapaths/datapath[lane]/state/interface-name
/components/component[name]/transceiver/cmismodule/module-state/datapaths/datapath[lane]/state/lane
/components/component[name]/transceiver/cmismodule/module-state/datapaths/datapath[lane]/state/media-rate

```

Transceiver CMIS Module Monitor States

Sensor Path

```

ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmismodule/module-monitors/monitors/
monitor[id=MONITOR_ID]/state

```

Leaf Attributes

```

/components/component[name]/transceiver/cmismodule/module-monitors/monitors/monitor[id]/state/description
/components/component[name]/transceiver/cmismodule/module-monitors/monitors/monitor[id]/state/high-alarm
/components/component[name]/transceiver/cmismodule/module-monitors/monitors/monitor[id]/state/high-warning
/components/component[name]/transceiver/cmismodule/module-monitors/monitors/monitor[id]/state/id
/components/component[name]/transceiver/cmismodule/module-monitors/monitors/monitor[id]/state/low-alarm
/components/component[name]/transceiver/cmismodule/module-monitors/monitors/monitor[id]/state/low-warning
/components/component[name]/transceiver/cmismodule/module-monitors/monitors/monitor[id]/state/value

```

Transceiver CMIS Module Monitor Alarms

Sensor Path

```

ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmismodule/module-monitors/monitor-alarm/state

```

Leaf Attributes

```

/components/component[name]/transceiver/cmismodule/module-monitors/monitor-alarm/state/alarm-id

```

```

/components/component[name]/transceiver/cmisis-module/module-monitors/monitor-alarm/state/alarm-type
/components/component[name]/transceiver/cmisis-module/module-monitors/monitor-alarm/state/current-value
/components/component[name]/transceiver/cmisis-module/module-monitors/monitor-alarm/state/threshold-minimum
/components/component[name]/transceiver/cmisis-module/module-monitors/monitor-alarm/state/threshold-maximum

```

Transceiver CMIS Host Monitor States

Sensor Path

```

ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/host-monitors/lanes/
lane[number=LANE_NUMBER]/state

```

Leaf Attributes

```

/components/component[name]/transceiver/cmisis-module/host-monitors/lanes/lane[number]/state/dp-assigned
/components/component[name]/transceiver/cmisis-module/host-monitors/lanes/lane[number]/state/number

```

Transceiver CMIS Host Monitoring for Monitors

Sensor Path

```

ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/host-monitors/lanes/
lane[number=LANE_NUMBER]/monitors/monitor[id=MONITOR_ID]/state

```

Leaf Attributes

```

description /components/component[name]/transceiver/cmisis-module/host-monitors/lanes/lane[number]/monitors/monitor[id]/state/
id /components/component[name]/transceiver/cmisis-module/host-monitors/lanes/lane[number]/monitors/monitor[id]/state/
value /components/component[name]/transceiver/cmisis-module/host-monitors/lanes/lane[number]/monitors/monitor[id]/state/

```

Transceiver CMIS Host Monitoring for Flags

Sensor Path

```

ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/host-monitors/lanes/
lane[number=LANE_NUMBER]/flags/flag[id=FLAG_ID]/state

```

Leaf Attributes

```

description /components/component[name]/transceiver/cmisis-module/host-monitors/lanes/lane[number]/flags/flag[id]/state/
id /components/component[name]/transceiver/cmisis-module/host-monitors/lanes/lane[number]/flags/flag[id]/state/id
value /components/component[name]/transceiver/cmisis-module/host-monitors/lanes/lane[number]/flags/flag[id]/state/value

```

Transceiver CMIS Host Monitoring for Alarms

Sensor Path

```

ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/host-monitors/lanes/
lane[number=LANE_NUMBER]/monitor-alarm/state

```

Leaf Attributes

```

id /components/component[name]/transceiver/cmisis-module/host-monitors/lanes/lane[number]/monitor-alarm/state/alarm-
type /components/component[name]/transceiver/cmisis-module/host-monitors/lanes/lane[number]/monitor-alarm/state/alarm-
current-value /components/component[name]/transceiver/cmisis-module/host-monitors/lanes/lane[number]/monitor-alarm/state/
threshold-minimum /components/component[name]/transceiver/cmisis-module/host-monitors/lanes/lane[number]/monitor-alarm/state/
threshold-maximum /components/component[name]/transceiver/cmisis-module/host-monitors/lanes/lane[number]/monitor-alarm/state/

```

Transceiver CMIS Host Monitoring for Alarm Flags

Sensor Path

```

ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/host-monitors/lanes/
lane[number=LANE_NUMBER]/flag-alarm/state

```

Leaf Attributes

```

/components/component[name]/transceiver/cmisis-module/host-monitors/lanes/lane[number]/flag-alarm/state/alarm-id

```

Transceiver CMIS Media Monitoring for State

Sensor Path

```

ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/media-monitors/lanes/
lane[number=LANE_NUMBER]/monitor/monitor[id=MONITOR_ID]/state

```

Leaf Attributes

```

/components/component[name]/transceiver/cmisis-module/media-monitors/lanes/lane[number]/monitors/monitor[id]/
state/description
/components/component[name]/transceiver/cmisis-module/media-monitors/lanes/lane[number]/monitors/monitor[id]/
state/high-alarm
/components/component[name]/transceiver/cmisis-module/media-monitors/lanes/lane[number]/monitors/monitor[id]/
state/high-warning
/components/component[name]/transceiver/cmisis-module/media-monitors/lanes/lane[number]/monitors/monitor[id]/
state/id
/components/component[name]/transceiver/cmisis-module/media-monitors/lanes/lane[number]/monitors/monitor[id]/
state/low-alarm
/components/component[name]/transceiver/cmisis-module/media-monitors/lanes/lane[number]/monitors/monitor[id]/
state/low-warning
/components/component[name]/transceiver/cmisis-module/media-monitors/lanes/lane[number]/monitors/monitor[id]/
state/value

```

Transceiver CMIS Media Monitoring for Flags

Sensor Path

```

ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/media-monitors/lanes/
lane[number=LANE_NUMBER]/flags/flag[id=FLAG_ID]/state

```

Leaf Attributes

```

/components/component[name]/transceiver/cmisis-module/media-monitors/lanes/lane[number]/flags/flag[id]/state/
description
/components/component[name]/transceiver/cmisis-module/media-monitors/lanes/lane[number]/flags/flag[id]/state/id
/components/component[name]/transceiver/cmisis-module/media-monitors/lanes/lane[number]/flags/flag[id]/state/value

```

Transceiver CMIS Media Monitoring for Alarms

Sensor Path

```

ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/media-monitors/lanes/
lane[number=LANE_NUMBER]/monitor-alarm/state

```

Leaf Attributes

```

/components/component[name]/transceiver/cmisis-module/media-monitors/lanes/lane[number]/monitor-alarm/state/alarm-
id
/components/component[name]/transceiver/cmisis-module/media-monitors/lanes/lane[number]/monitor-alarm/state/alarm-
type
/components/component[name]/transceiver/cmisis-module/media-monitors/lanes/lane[number]/monitor-alarm/state/
current-value
/components/component[name]/transceiver/cmisis-module/media-monitors/lanes/lane[number]/monitor-alarm/state/
threshold-minimum
/components/component[name]/transceiver/cmisis-module/media-monitors/lanes/lane[number]/monitor-alarm/state/
threshold-maximum

```

Transceiver CMIS Media Monitoring for Alarm Flags

Sensor Path

```

ipi:/components/component[name=CMIS-MODULE-NAME]/transceiver/cmisis-module/media-monitors/lanes/
lane[number=LANE_NUMBER]/flag-alarm/state

```

Leaf Attributes

```

/components/component[name]/transceiver/cmisis-module/media-monitors/lanes/lane[number]/flag-alarm/state/alarm-id

```

OpenConfig Data Models

Streaming telemetry incrementally supports all listed OpenConfig data models, providing standardized representations of network configurations and telemetry data.

Telemetry OpenConfig Pyang Tree

The Pyang tree output below illustrates the supported containers or leaf, along with a list of supported container-level paths.

openconfig-platform

```

+--rw components
  +--ro component* [name]
    +--ro name                               -> ../state/name
    +--ro state
      | +--ro name?                          string
      | +--ro type?                          union
      | +--ro id?                            string
      | +--ro location?                      string
      | +--ro description?                   string
      | +--ro mfg-name?                      string
      | +--ro mfg-date?                      oc-yang:date
      | +--ro hardware-version?             string
      | +--ro firmware-version?            string
      | +--ro software-version?            string
      | +--ro serial-no?                    string
      | +--ro part-no?                      string
      | +--ro removable?                    boolean
      | +--ro oper-status?                  identityref
      | +--ro empty?                        boolean
      | +--ro parent?                       -> ../../../../component/state/name
      | +--ro temperature
      | | +--ro instant?                    decimal64
      | | +--ro avg?                        decimal64
      | | +--ro min?                        decimal64
      | | +--ro max?                        decimal64
      | | +--ro interval?                   oc-types:stat-interval
      | | +--ro alarm-status?               boolean
      | | +--ro alarm-threshold?           decimal64
      | | +--ro alarm-severity?            identityref
      | +--ro memory
      | | +--ro available?                  uint64
      | | +--ro utilized?                   uint64
    +--ro power-supply
      | +--ro state
      | | +--ro oc-platform-psu:enabled?    boolean
      | | +--ro oc-platform-psu:capacity?   decimal64
      | | +--ro oc-platform-psu:input-current? decimal64
      | | +--ro oc-platform-psu:input-voltage? decimal64
      | | +--ro oc-platform-psu:output-current? decimal64
      | | +--ro oc-platform-psu:output-voltage? decimal64
      | | +--ro oc-platform-psu:output-power? decimal64
    +--ro fan
      | +--ro state
    +--ro cpu
      | +--ro oc-cpu:utilization
      | | +--ro oc-cpu:state
      | | | +--ro oc-cpu:instant?          oc-types:percentage
      | | | +--ro oc-cpu:avg?              oc-types:percentage
      | | | +--ro oc-cpu:interval?        oc-types:stat-interval

```

openconfig-interfaces

```

+--rw interfaces
  +--ro interface* [name]
    +--ro name                               -> ../state/name
    +--ro state
      +--ro name?                          string
      +--ro ifindex?                       uint32
      +--ro admin-status                    enumeration
      +--ro oper-status                     enumeration
      +--ro last-change?                    oc-types:timeticks64
      +--ro logical?                        boolean
      +--ro counters
        +--ro in-octets?                    oc-yang:counter64
        +--ro in-pkts?                     oc-yang:counter64
        +--ro in-unicast-pkts?              oc-yang:counter64
        +--ro in-broadcast-pkts?           oc-yang:counter64
        +--ro in-multicast-pkts?           oc-yang:counter64
        +--ro in-discards?                  oc-yang:counter64
        +--ro in-errors?                    oc-yang:counter64

```

```

+--ro in-fcs-errors?      oc-yang:counter64
+--ro out-octets?        oc-yang:counter64
+--ro out-pkts?          oc-yang:counter64
+--ro out-unicast-pkts?  oc-yang:counter64
+--ro out-broadcast-pkts? oc-yang:counter64
+--ro out-multicast-pkts? oc-yang:counter64
+--ro out-discards?      oc-yang:counter64
+--ro out-errors?        oc-yang:counter64
+--ro last-clear?        oc-types:timeticks64

```

Container Level Sensor Paths and Leaf Attributes

The below section lists the container level sensor paths and leaf attributes supported for OpenConfig data models.

openconfig-platform

The listed paths below represent telemetry OpenConfig paths for monitoring the platform state of various components, including CPU, RAM, power supply, fans, CMIS, and transceivers.

CPU

```

Sensor Path
  /components/component[name]/cpu/utilization/state

```

```

Leaf Attributes
  /components/component[name]/cpu/utilization/state/avg
  /components/component[name]/cpu/utilization/state/instant
  /components/component[name]/cpu/utilization/state/interval

```

RAM

```

Sensor Path
  /components/component[name]/state/memory/

```

```

Leaf Attributes
  /components/component[name]/state/memory/available
  /components/component[name]/state/memory/utilized

```

Power-Supply

```

Sensor Path
  /components/component[name]/power-supply/state/

```

```

Leaf Attributes
  /components/component[name]/power-supply/state/capacity
  /components/component[name]/power-supply/state/output-power
  /components/component[name]/power-supply/state/input-voltage
  /components/component[name]/power-supply/state/input-current
  /components/component[name]/power-supply/state/output-voltage
  /components/component[name]/power-supply/state/output-current
  /components/component[name]/power-supply/state/enabled

```

Fan

```

Sensor Path
  /components/component[name]/fan/state

```

```

Leaf Attributes
  /components/component[name]/fan/state/speed

```

Platform State

Sensor Path

/components/component[name]/state

Leaf Attributes

/components/component[name]/state/id
 /components/component[name]/state/type
 /components/component[name]/state/location
 /components/component[name]/state/mfg-date
 /components/component[name]/state/description
 /components/component[name]/state/hardware-version
 /components/component[name]/state/firmware-version
 /components/component[name]/state/software-version
 /components/component[name]/state/serial-no
 /components/component[name]/state/part-no
 /components/component[name]/state/removable
 /components/component[name]/state/oper-status
 /components/component[name]/state/parent
 /components/component[name]/state/empty

Sensor Path

/components/component[name]/state/memory

Leaf Attributes

/components/component[name]/state/memory/available
 /components/component[name]/state/memory/utilized

Sensor Path

/components/component[name]/state/temperature

Leaf Attributes

/components/component[name]/state/temperature/instant
 /components/component[name]/state/temperature/min
 /components/component[name]/state/temperature/max
 /components/component[name]/state/temperature/avg
 /components/component[name]/state/temperature/interval
 /components/component[name]/state/temperature/alarm-status
 /components/component[name]/state/temperature/alarm-threshold
 /components/component[name]/state/temperature/alarm-severity

openconfig-interface

The listed paths below represent telemetry OpenConfig paths for monitoring the interface and counters state.

Interface State

Sensor Path

/interfaces/interface[name]/state

Leaf Attributes

/interfaces/interface[name]/state/name
 /interfaces/interface[name]/state/ifindex
 /interfaces/interface[name]/state/admin-status
 /interfaces/interface[name]/state/oper-status
 /interfaces/interface[name]/state/last-change
 /interfaces/interface[name]/state/logical

Counters State

Sensor Path

```
/interfaces/interface[name]/state/counters
```

Leaf Attributes

```
/interfaces/interface[name]/state/counters/in-octets  
/interfaces/interface[name]/state/counters/in-pkts  
/interfaces/interface[name]/state/counters/in-unicast-pkts  
/interfaces/interface[name]/state/counters/in-broadcast-pkts  
/interfaces/interface[name]/state/counters/in-multicast-pkts  
/interfaces/interface[name]/state/counters/in-discards  
/interfaces/interface[name]/state/counters/in-errors  
/interfaces/interface[name]/state/counters/in-fcs-errors  
/interfaces/interface[name]/state/counters/out-octets  
/interfaces/interface[name]/state/counters/out-pkts  
/interfaces/interface[name]/state/counters/out-unicast-pkts  
/interfaces/interface[name]/state/counters/out-broadcast-pkts  
/interfaces/interface[name]/state/counters/out-multicast-pkts  
/interfaces/interface[name]/state/counters/out-discards  
/interfaces/interface[name]/state/counters/out-errors  
/interfaces/interface[name]/state/counters/last-clear
```