

# Cisco IOS XE gRPC Model-Driven Telemetry

---

# Contents

Introduction	3
YANG data models	4
gRPC telemetry configuration	5
Telemetry receiver	6
Conclusion and resources	7

## Introduction

Cisco IOS® XE is the network operating system for the enterprise. It runs on switches such as the Cisco® Catalyst® 9000 family; routers such as the Cisco ASR 1000 Series, Cloud Services Router (CSR) 1000v, and 1000 and 4000 Series Integrated Services Routers (ISRs); and the Cisco Catalyst 9800 Series Wireless Controllers, as well as a few other hardware and virtual platforms both on-premises and in the public cloud. Since Cisco IOS XE Release 16.6, there has been support for Model-Driven Telemetry (MDT), which provides network operators with additional options for getting information from their network using the NETCONF interface. The introduction of Google Remote Procedure Call (gRPC) based MDT to Release 16.10 now provides an additional interface for telemetry publication, backed by YANG data models. Unlike the original NETCONF telemetry interface, which is “dial in” and session based, the gRPC interface is “dial out” and is based on configuration within the Cisco IOS XE device.

**Table 1.** MDT interfaces

	NETCONF	gRPC	gNMI
<b>IOS XE Release</b>	16.6+	16.10+	16.12+
<b>Method</b>	Dial-in, pull	Dial-out, push	Dial-in, pull
<b>Configuration</b>	Per session	Configuration based	Per session
<b>Telemetry receiver</b>	Client	Server	Client
<b>Encoding</b>	XML	JSON+Protobuf	JSON_IETF
<b>Security</b>	SSH+PKI	Plain text	TLS Certificate
<b>Data models</b>	YANG	YANG	YANG

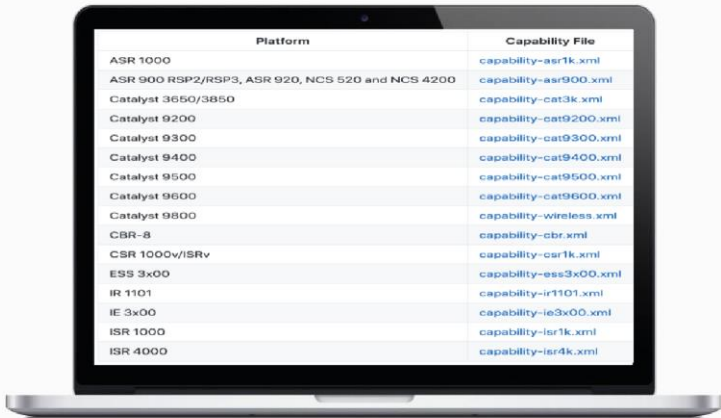
Traditionally, Simple Network Management Protocol (SNMP) has been highly successful for monitoring enterprise networks, but it has limitations: unreliable transport, inconsistent encoding between versions, and limited filtering and data retrieval options, as well as the impact to the CPU and memory of the running device when multiple network monitoring solutions poll the device simultaneously. Model-driven telemetry with gRPC addresses many of the shortfalls of the legacy monitoring capabilities and provides an additional interface that telemetry is now available to be published from. Yes, this is a push-based feature. No longer do you need to poll the device and ask for operational data or the current configuration state. Now you just decide what data is needed, how often, and where to send it to. Once the configuration is in place, the Cisco IOS XE device happily publishes the telemetry data to the third-party collectors, your monitoring tools, big data search and visualization engines such as Splunk and Elastic, or even to a simple text file – what you do with the data is totally configurable. In the example described here we use Telegraf plus InfluxDB plus Grafana to receive, store, and visualize the data.

## YANG data models

YANG data models (Yet Another Next Generation!) are at the heart of model-driven telemetry. These human-readable text-based models define the data that is available, not just for telemetry publication but also for programmatic configuration. These data models reside within the Cisco IOS XE device and can easily be downloaded when using tools such as YANG Explorer. All of the models are also published on the YangModels GitHub page at <https://github.com/YangModels/yang/tree/master/vendor/cisco/xr>, which makes them easy to access and analyze.

YANG data models fall into two main categories: operational and configuration. The operational data models contain run-time and operational-level details for features. An example of operational data for an interface might be the statistics and counters about a specific interface. Configuration data models contain the configuration information for the feature, similar to the traditional “show run” feature of Cisco IOS. An example of configuration data from the interface model might be the interface description or which VLAN a specific switch port is assigned to.

**IOS XE YANG Model - Capabilities per platform**



Platform	Capability File
ASR 1000	capability-asr1k.xml
ASR 900 RSP2/RSP3, ASR 920, NCS 520 and NCS 4200	capability-asr900.xml
Catalyst 3650/3850	capability-cat3k.xml
Catalyst 9200	capability-cat9200.xml
Catalyst 9300	capability-cat9300.xml
Catalyst 9400	capability-cat9400.xml
Catalyst 9500	capability-cat9500.xml
Catalyst 9600	capability-cat9600.xml
Catalyst 9800	capability-wireless.xml
CSR-8	capability-csr.xml
CSR 1000v/ISRV	capability-csr1k.xml
ESS 3x00	capability-ess3x00.xml
IR 1101	capability-ir1101.xml
IE 3x00	capability-ie3x00.xml
ISR 1000	capability-isr1k.xml
ISR 4000	capability-isr4k.xml

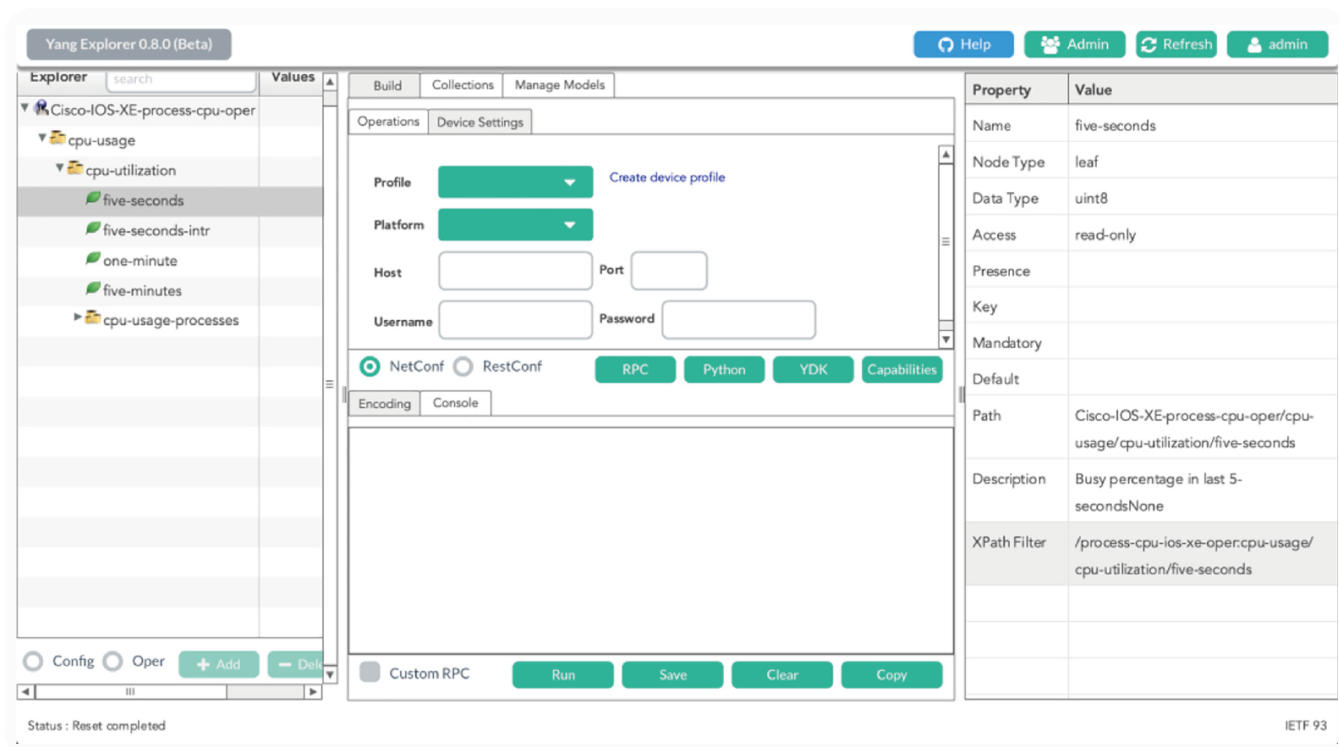
**Figure 1.**  
YANG model capabilities for Cisco platforms

### Use case: Interface statistics, counters, and CPU memory utilization

One common use case is to monitor the interface state, errors and counters, and CPU and memory utilization of the device. Let’s understand where and how we can get this data from our Cisco Catalyst 9300 Series switch running Cisco IOS XE 16.10.

The YANG Explorer tool is available on the CiscoDevNet GitHub page at <https://github.com/CiscoDevNet/yang-explorer>. It can download the YANG models directly from the Cisco IOS XE device over the programmatic interfaces (NETCONF, RETCONF, and gRPC Network Management Interface [gNMI]) and quickly show which data is available and from which model.

The screen shot below from the YANG Explorer shows that we have already downloaded and loaded the Cisco-IOS-XE-process-cpu-oper.yang model and have begun to explore it. It shows that the 1-minute, 5-minute, and 5-second “Busy CPU-Utilization” is available as a percentage, as well as the Interrupt 5-second (five-second-intr) metric. It also shows some metadata about the model that includes the XPath, Prefix, Namespace, and a description with some other details.



## gRPC telemetry configuration

Now that we know which YANG data model contains the needed information, let's enable sending this telemetry from the device. It is possible to configure and verify telemetry subscriptions from the traditional Cisco IOS Command-Line Interface (CLI) as well as through the NETCONF, RESTCONF, and gNMI programmatic interfaces using YANG. In the CLI, the show commands are available with the "show telemetry ietf" set of commands, and their syntax is similar to the "telemetry ietf subscriptions" commands when in configure mode. When YANG is used, the Cisco-IOS-XE-mdt-cfg.yang and Cisco-IOS-XE-mdt-oper.yang YANG models are available for both configuration and operational datasets.

Let's look at a CLI configuration example from a Cisco Catalyst 9300 Series switch running Cisco IOS XE 16.10. This configuration:

- Enables a telemetry configuration with an identification number of 501
- Sets encoding to "kvgbp," the self-describing key-value pair Google Protocol Buffer
- Sets the YANG model filter (filter xpath) to "process-cpu-ios-xe-oper," which represents the Cisco-IOS-XE-process-cpu-oper.yang YANG data model
- Sets the YANG model xpath (filter xpath) to "/cpu-usage/cpu-utilization/five-seconds," which will publish 5-second CPU utilization telemetry
- Sets the source address and source VRF so that the device knows which IP or interface to use
- Sets the periodic update policy to 500 centiseconds (5-second updates)
- Defines the IP address and port that the remote grpc-tcp server is listening to

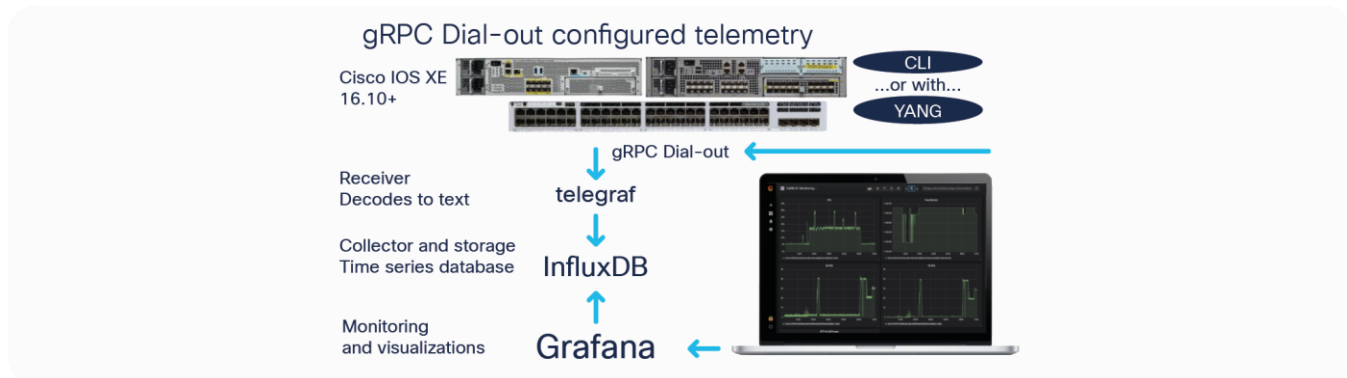
```

C9300#show running-config | section telemetry
telemetry ietf subscription 501
  encoding encode-kvgpb
  filter xpath /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds
  source-address 10.60.0.19
  source-vrf Mgmt-vrf
  stream yang-push
  update-policy periodic 500
  receiver ip address 10.12.252.224 57000 protocol grpc-tcp
C9300#

```

## Telemetry receiver

Software is required to receive and process the telemetry data, and there are many options available to accomplish this. Splunk, LogRhythm, and Elasticsearch are off-the-shelf solutions, while custom and open-source options written with Python or Go are also available. One option is to use the open source software stack that allows easy reception, decoding, and processing of the “kvgpb” telemetry, which is referred to as the TIG stack. TIG represents three separate software components: Telegraf, which receives the telemetry data, InfluxDB, which stores it; and Grafana, which is responsible for visualizations and alerting.



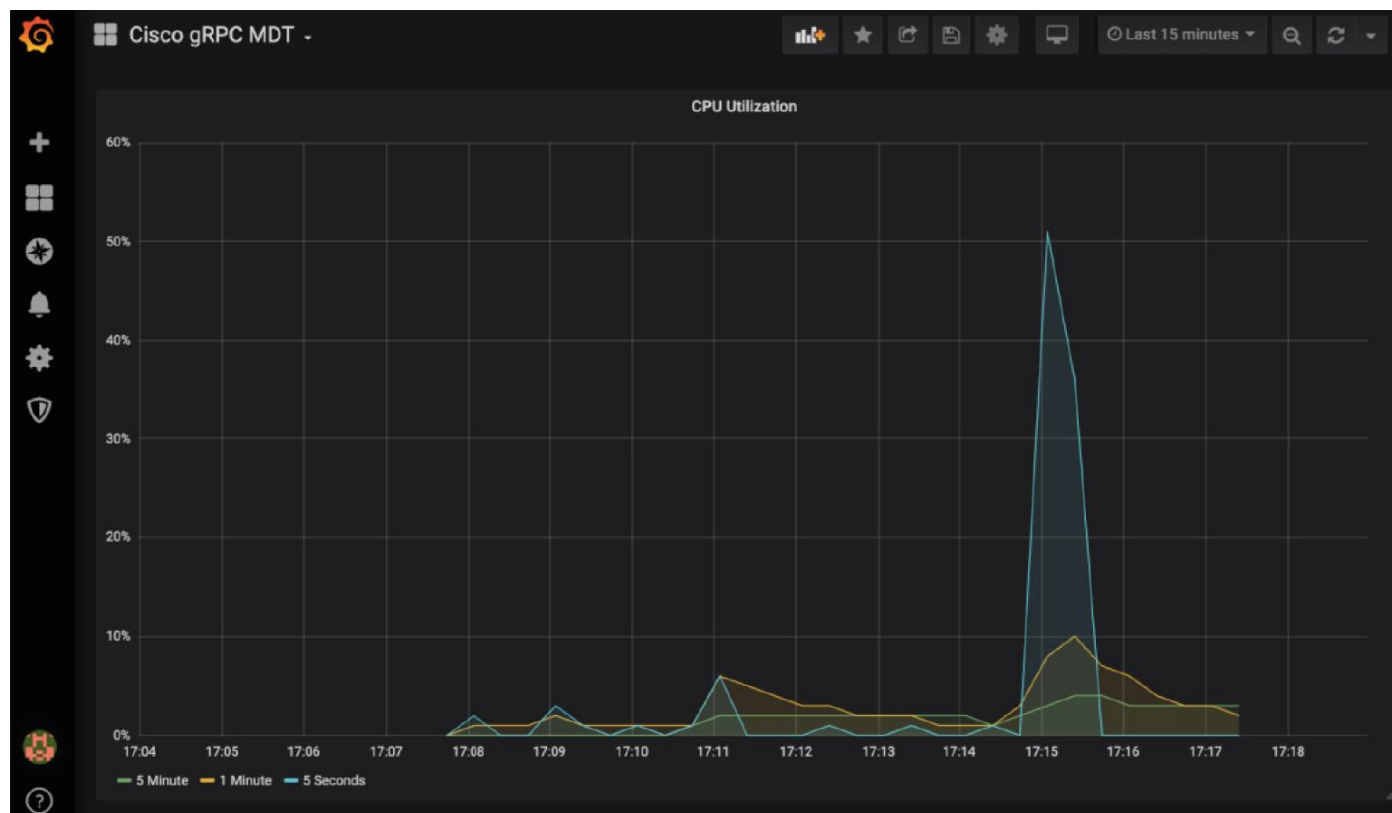
**Figure 2.**  
gRPC dial-out configured telemetry

**Telegraf** has the “cisco\_telemetry\_mdt” input plug-in that receives and decodes the gRPC messages that the Cisco IOS XE device sends. It also has an output plug-in that can send this data into the InfluxDB datastore, where it is stored. The configuration for Telegraf is simple and static because once it’s set up it rarely needs to be reconfigured or modified. Simply define a few global parameters, the input, and the output, and then start the Telegraf service to start receiving telemetry from the Cisco IOS XE device. In this example we configure the gRPC input listener on port 57000 – this is the port that Cisco IOS XE will publish telemetry to. We have also configured where to send the data to once it is received by Telegraf: InfluxDB running on the localhost, port 8086, as well as the database, username, and password to use for the datastore.

**InfluxDB** is used to store the telemetry data that is received from Telegraf. It can hold many “databases,” or a single database can be used for all telemetry data. For simplicity we are using one database in our example, but in production deployments multiple smaller databases may be beneficial for speed and search optimization.

**Grafana** is the visualization engine that is used to display the telemetry data. It calls into InfluxDB to access the data that is stored there, which is the same data that Telegraf received from Cisco IOS XE. In this example there are three unique queries for each of the CPU metrics: 5 minute, 1 minute, and 5 seconds, and those are shown in the graph. We can see that the 5-second CPU average in blue peaks at around 50% utilization; meanwhile the slower averages of 1 minute and 5 minutes increase only to around 10%.

Refer to the official guides in order to set up, configure, and operate Telegraf, InfluxDB, and Grafana in your environment as needed.



## Conclusion and resources

We understand why moving away from SNMP is important, and now that additional streaming telemetry protocols and interfaces are available in Cisco IOS XE we can start transitioning toward their use. The open source Telegraf, InfluxDB, and Grafana software stack can easily be set up to receive, store, and visualize this data from the network. A DevNet Learning Lab has been created that has the Cisco Catalyst 9300 Series switch configured to send gRPC telemetry into the TIG stack on an Ubuntu Linux host. Reserve the sandbox to get hands-on experience with this feature, available from [https://developer.cisco.com/learning/modules/iosxe\\_telemetry](https://developer.cisco.com/learning/modules/iosxe_telemetry). There are several Cisco blogs about leveraging the new model-driven telemetry capabilities; these can be found at <https://blogs.cisco.com/developer/its-time-to-move-away-from-snmp-and-cli-and-use-model-driven-telemetry> and <https://blogs.cisco.com/developer/getting-started-with-model-driven-telemetry>. Finally, the Cisco IOS XE Programmability Configuration Guide has detailed information about MDT. It is linked from the Cisco IOS XE site at <https://developer.cisco.com/site/ios-xe/>.

---

**Americas Headquarters**  
Cisco Systems, Inc.  
San Jose, CA

**Asia Pacific Headquarters**  
Cisco Systems (USA) Pte. Ltd.  
Singapore

**Europe Headquarters**  
Cisco Systems International BV Amsterdam,  
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at <https://www.cisco.com/go/offices>.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)