The bridge to possible

Streaming Telemetry on Cisco NX-OS

Shangxin Du
Technical Marketing Engineer, Datacenter Switching

Cisco Live!

BRKDCN-2689

Cisco Webex App

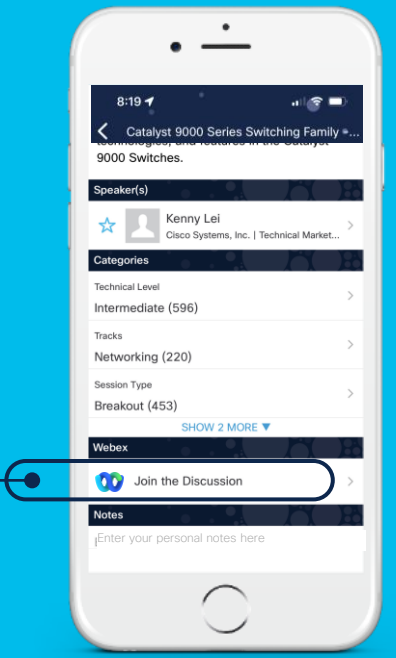
Questions?

Use Cisco Webex App to chat
with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated until February 24, 2023.

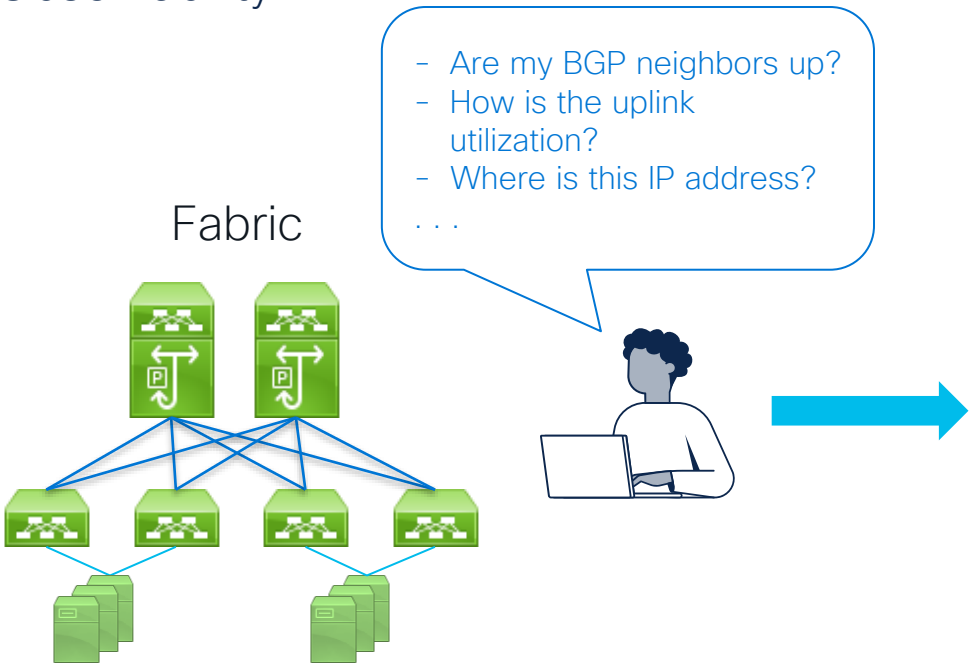


Agenda

- Why do we need streaming telemetry?
- Telemetry data sources, subscription modes, and encodings
- Transport options and design consideration
- How to build telemetry system with opensource tools

Why do we need streaming telemetry

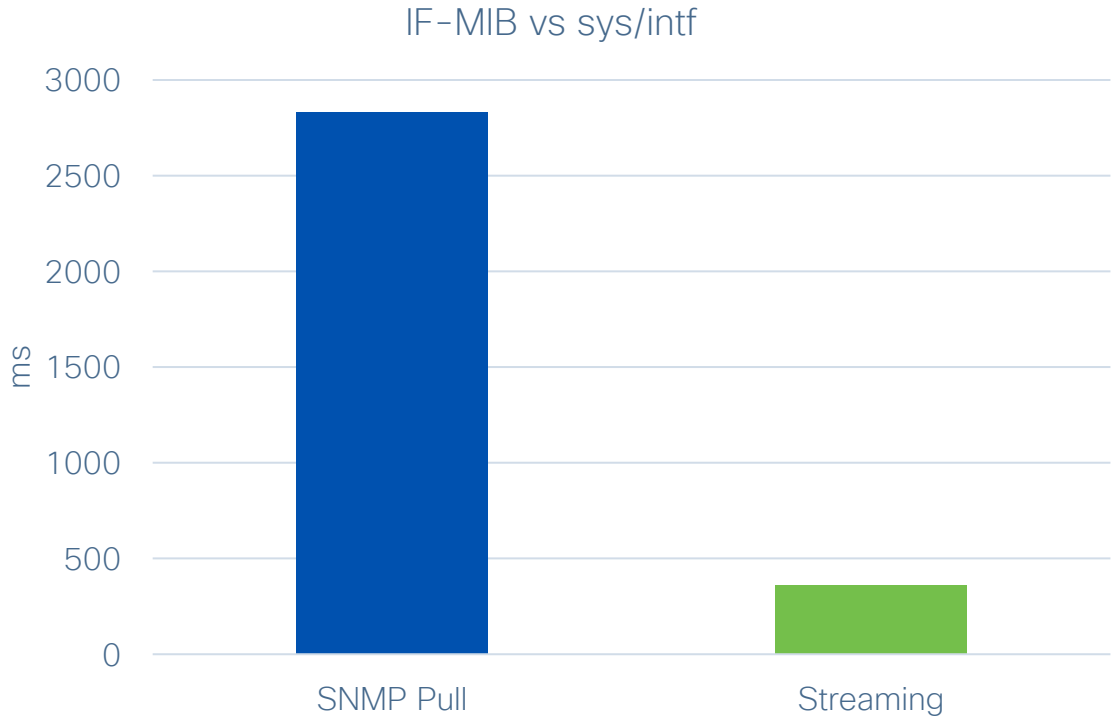
Observability



- What information shall I collect?
- How do I collect those metrics?
- What can I do with those data?

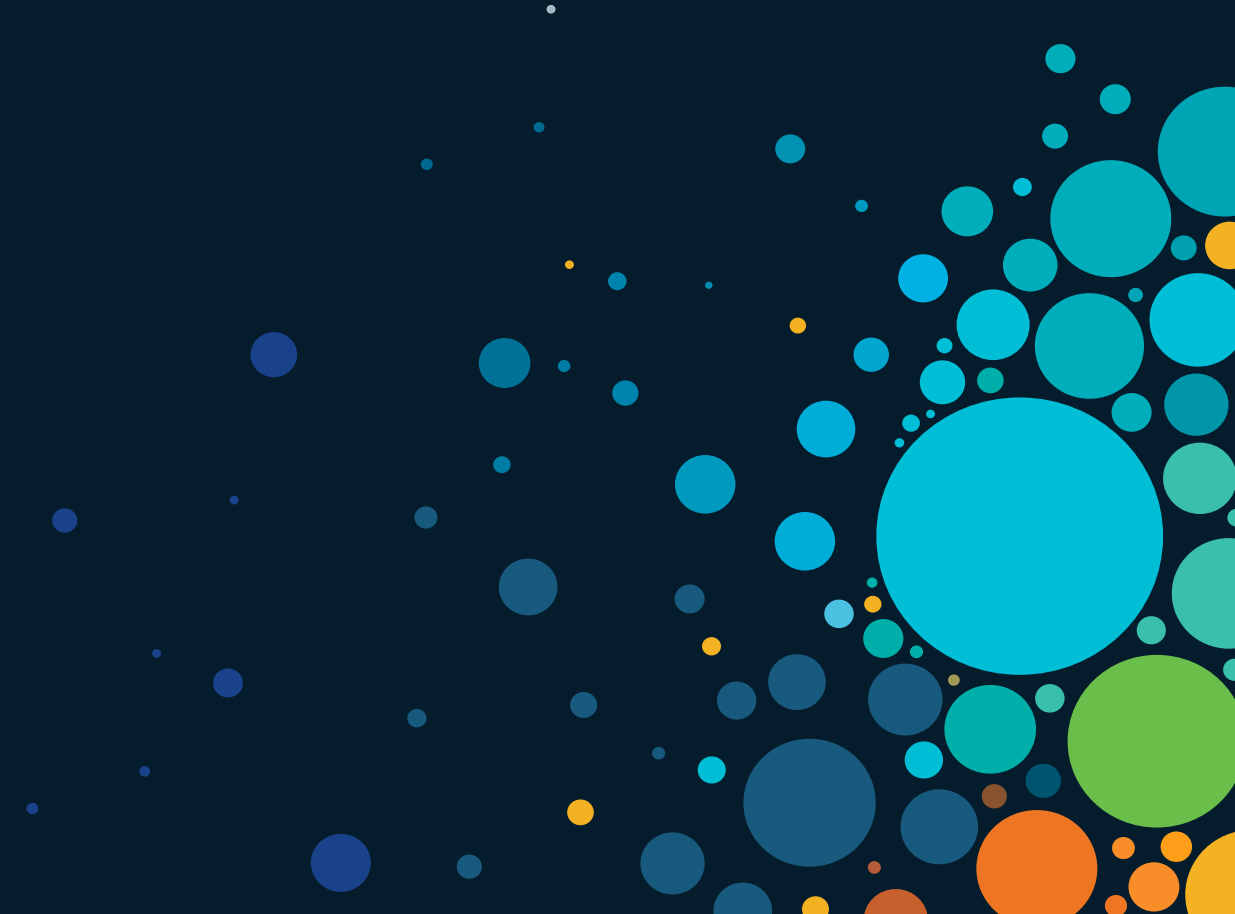
Why do we need streaming telemetry

Performance



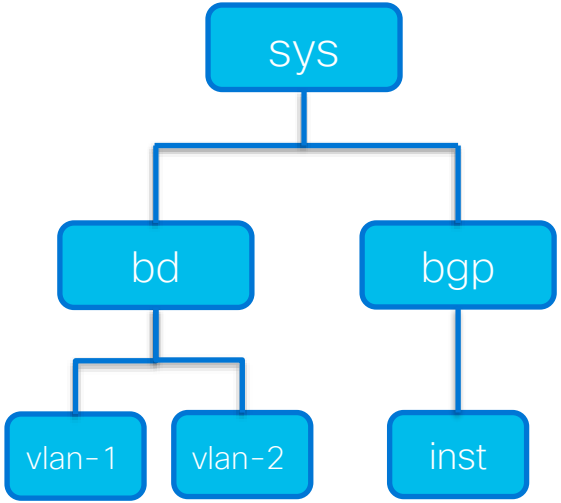
- SNMP bulk-get **`IF-MIB`**
- Streaming DME sensor path **`sys/intf`**

Data Sources



Data Source

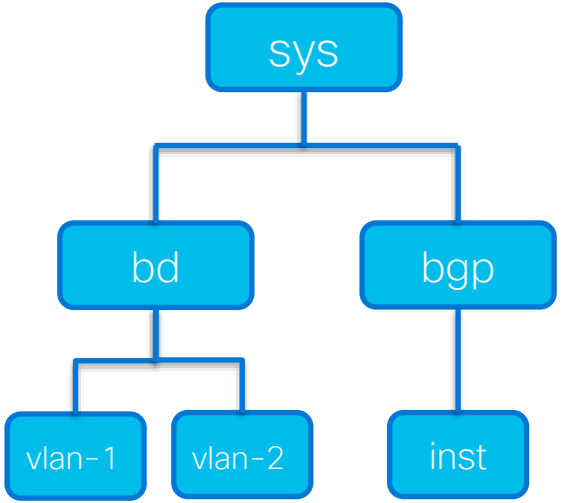
DME



- Tree data structure
  - The root of the tree is **`sys`**
- DN(Distinguished Name) is in `../../..` Format
  - Ex, **`sys/bgp/inst`** is representing the bgp instance on the switch, it contains all config and state of BGP process
- When streaming telemetry, use DN as a sensor path

Data Source

What is available in DME?



- Almost everything
  - As 10.3(2)F, over 90% of the command are DMElized
  - Configuration data and Operational data
- Support event-based and sample-based telemetry
- The extra filter is supported to minimize the data size

Data Source

How to get sensor path of DME

Visore is built-in DME browser of NX-OS, navigate to https://[ip_of_swtich]/visore.html

rmonIfIn	?
broadcastPkts	199779
clearTs	never
discards	0
dn	sys/intf/phys-[eth1/27]/dbgIfIn
errors	0
modTs	2022-03-28T16:45:11.658+00:00
multicastPkts	345290
nUcastPkts	545069
noBuffer	0
octetRate	3657496
octets	11346525403646
packetRate	3438
rateInterval	300
ucastPkts	3777158007
unknownEtype	0
unknownProtos	0

API reference is also available:
https://developer.cisco.com/site/nxapi-dme-model-reference-api/?version=10.2(2)

rmon:IfHCIn

The interface high capacity input statistics.

Telemetry Sensor Path(s)

- sys/mgmt-[id]/dbgIfHCIn
- sys/intf/phys-[id]/dbgIfHCIn
- sys/intf/aggr-[id]/dbgIfHCIn

Operational Properties

PROPERTY NAME	DATA TYPE	DESCRIPTION	POSSIBLE VALUES
broadcastPkts	scalar:Uint64	Broadcast Packets	RANGE: [0, 18446744073709551615]
multicastPkts	scalar:Uint64	Multicast Packets	RANGE: [0, 18446744073709551615]
octets	scalar:Uint64	Octets	RANGE: [0, 18446744073709551615]
ucastPkts	scalar:Uint64	Unicast Packets	RANGE: [0, 18446744073709551615]

Data Source

YANG Model

- YANG(Yet Another Next Generation) is a data modeling language used to describe the data sent over the network
- NX-OS supports two different types of YANG model
  - The Openconfig(OC) YANG model is vendor agnostic
  - The Native/Device YANG model is vendor-specific
- Using XPATH(XML Path Language) for addressing

Example of xpath:

`openconfig-interfaces:interfaces/interface/state/oper-status`

model name
container
list
leaf

Data Source

Supported OC YANG model

model	Revision in 10.3(1)F
openconfig-aaa.yang	2019-10-28
openconfig-acl.yang	2019-11-27
openconfig-bfd.yang	2020-05-08
openconfig-bgp.yang	2019-07-10
openconfig-igmp.yang	2019-07-09
openconfig-interfaces.yang	2019-11-19
openconfig-isis.yang	2020-03-24
openconfig-lacp.yang	2018-11-21
openconfig-lldp.yang	2018-11-21
openconfig-mpls.yang	2019-03-26
openconfig-network-instance.yang	2022-04-20
openconfig-ospfv2.yang	2019-11-28
openconfig-pim.yang	2019-07-09
openconfig-platform.yang	2019-04-16
openconfig-qos.yang	2019-11-28
openconfig-routing-policy.yang	2018-11-21
openconfig-system.yang	2020-03-25

- To support OC YANG
  - Before 10.2(2)F, `mtx-openconfig-all` rpm needs to be installed on the streaming switch, refer to the programmability guide to install the package
  - After 10.2(2)F, use `feature openconfig` to enable
- Beware of deviation, the model is supported doesn't mean all the paths are supported
  - Like all other vendors, the deviation is created when a certain path is not following the definition in OC models, or the path is not supported

- A full list of supported models and deviations is published on GitHub: https://github.com/YangModels/yang/tree/master/vendor/cisco/nx

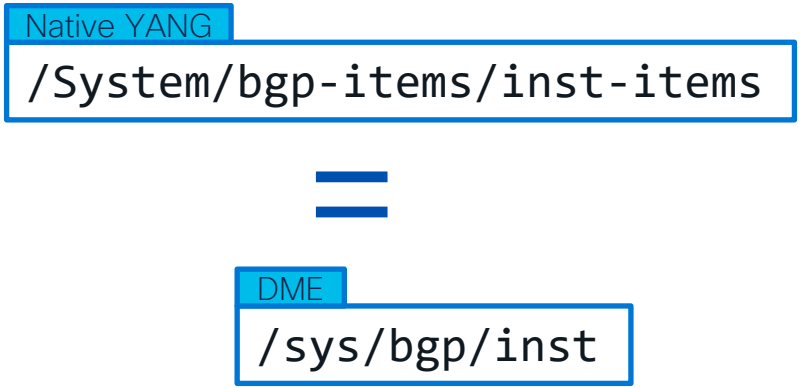
Openconfig VXLAN EVPN model

- Cisco co-authors with Google and Telefónica
- Phase one will focus on operational data of EVPN address family
- Shipping in 10.3(1)F

Phase one components
L2rib
L3fib
Adjacency(ARP/ND)
BGP Type2 Routes
BGP Type5 Routes
VXLAN NVE state

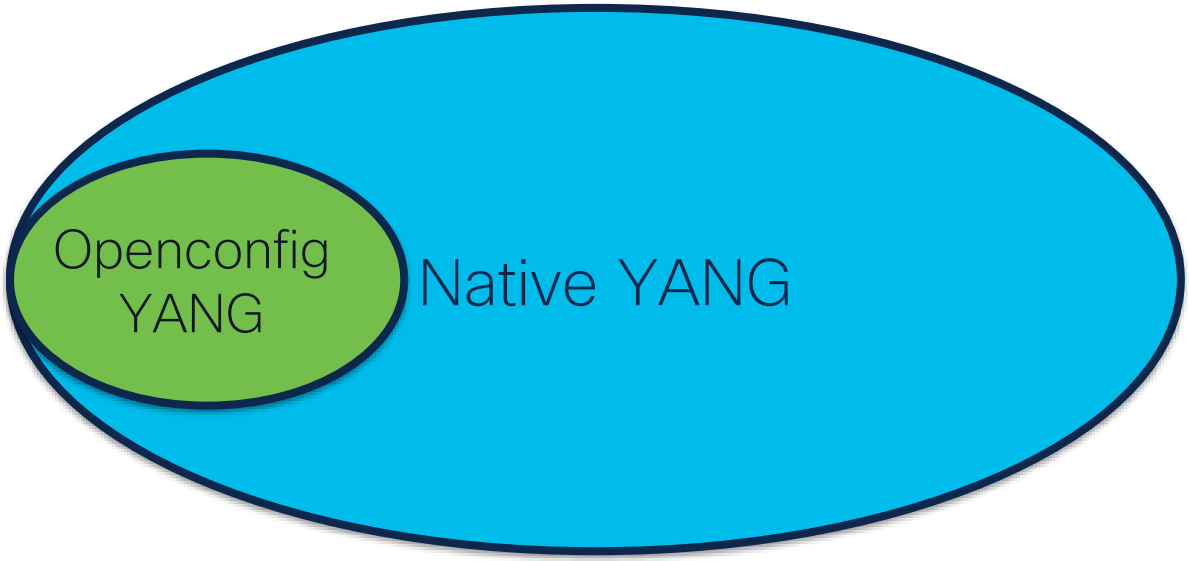
Data Source

Native YANG



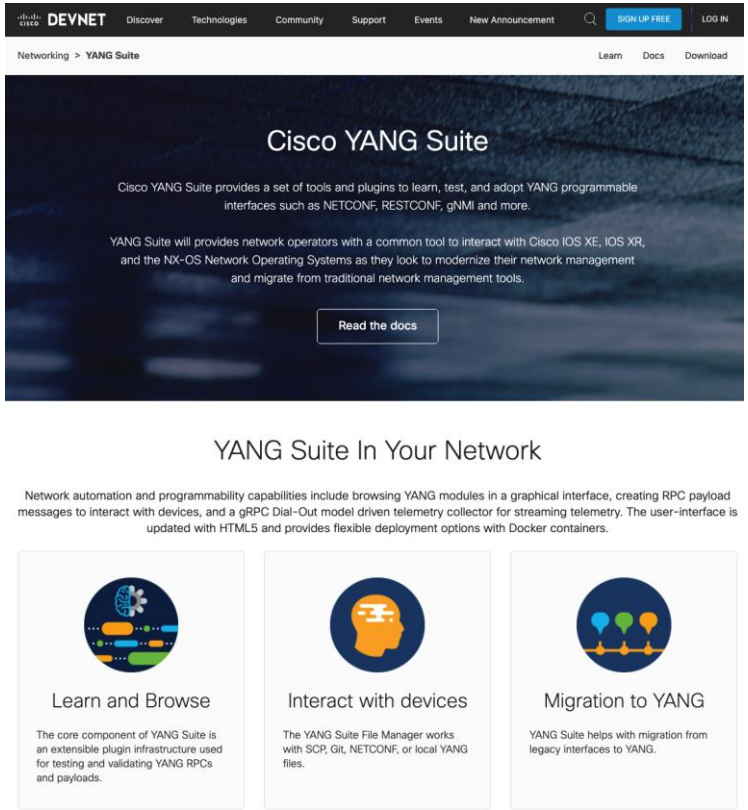
- The Native YANG model is vendor-specific but still described in YANG, aka Device Yang.
- NX-OS Native YANG is defined in *Cisco-NX-OS-device.yang*
- It is1:1 mapping from DME objects to Native YANG

Openconfig YANG and Native YANG



YANG Suite

Swiss knife of Yang



- One-stop tool for automating network devices using the YANG model
- Construct and test YANG base API interface over NETCONF, RESTCONF and gNMI
- YANG model browser built-in

https://developer.cisco.com/yangsuite

Data Source

CLI/NX-API

```
93240YC-FX2-L02-S4# show nve vni  | json-pretty
{
    "TABLE_nve_vni": {
        "ROW_nve_vni": [
            {
                "if-name": "nve1",
                "vni": "30000",
                "mcast": "239.1.1.1",
                "vni-state": "Up",
                "mode": "CP",
                "type": "L2 [2300]",
                "flags": null,
                "dci-mcast": "Unconfigured"
            },
...
```

- Well-known CLI with structure output
- 100% of customer-facing show command of NX-OS has structured output

- Only supports sample-based telemetry
- CLI doesn‘t have a native data type, all the value is a string type, and the collector need to parse the result to “guess” the type of the data

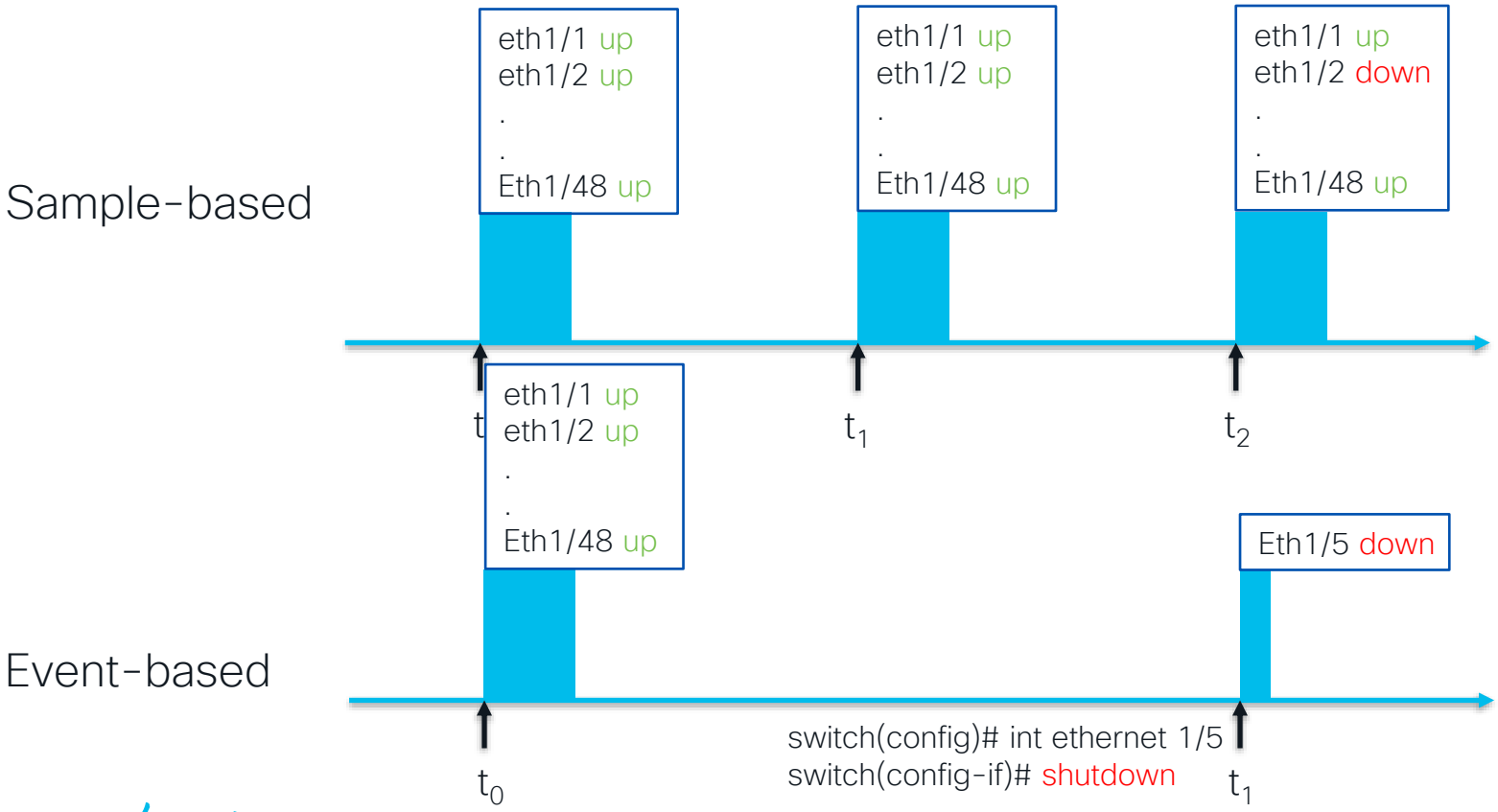
Data Model Platform Support

Nexus Platform	DME	CLI/NX-API	YANG	Release
3000 with 8G+ RAM	✓	✓	✓ *	7.0(3)I7(1)
9300	✓	✓	✓ *	7.0(3)I5(1)
9500/9400/9800	✓	✓	✓ *	7.0(3)I7(1)
7000/7700	✗	✓	✗	8.3(1)

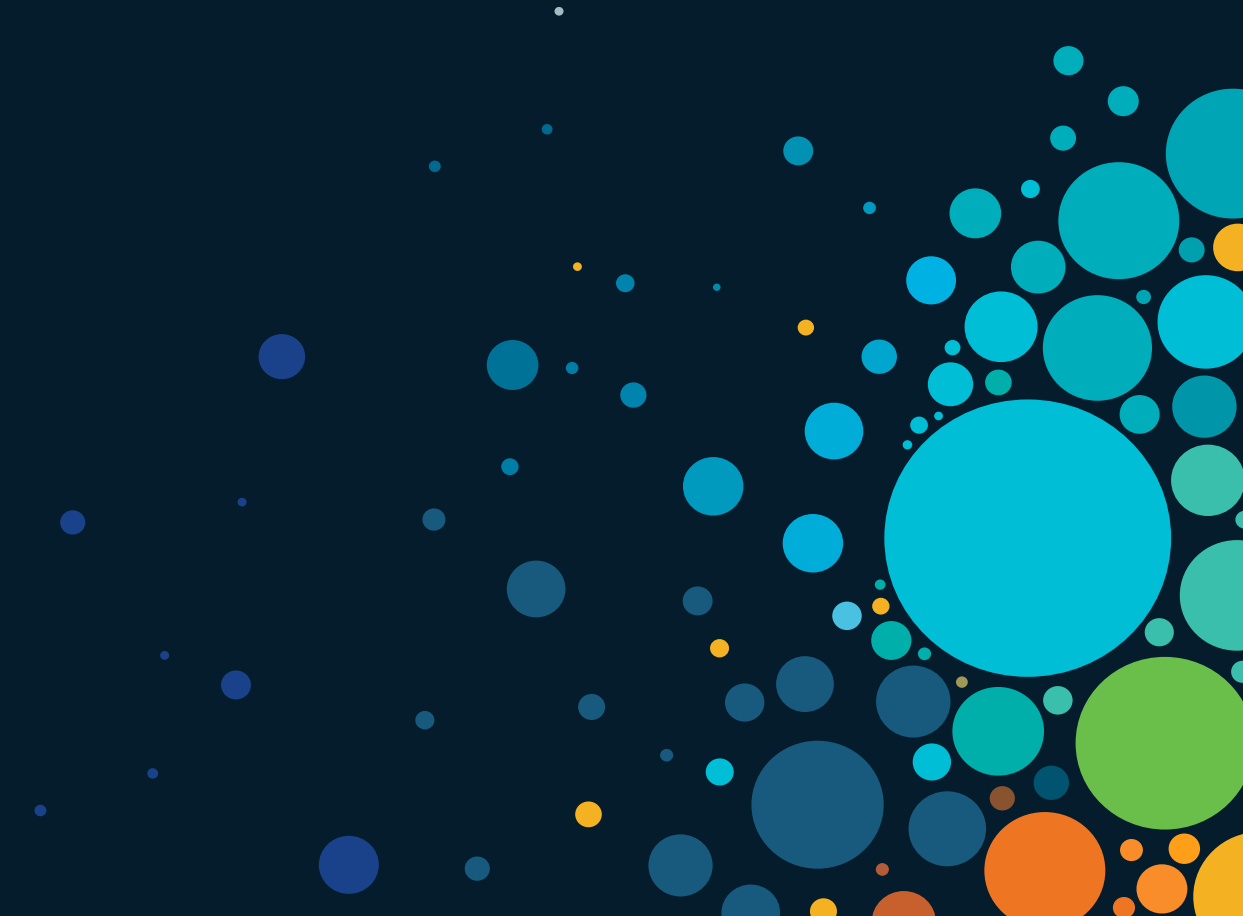
* Streaming Yang models start from 9.2(1)

Sample-base or Event-based telemetry

Sample-based vs Event-based



Encodings



How does GPB(Google Protocol Buffers) work

```
<interface>
  <name>eth1/49</name>
  <state>
    <counters>
      <in-broadcast-pkts>2</in-broadcast-pkts>
      <in-discards>0</in-discards>
      <in-errors>0</in-errors>
      <in-fcs-errors>0</in-fcs-errors>
      <in-multicast-pkts>30543</in-multicast-pkts>
      <in-octets>13320913920</in-octets>
      <in-unicast-pkts>5406026</in-unicast-pkts>
      <in-unknown-protos>0</in-unknown-protos>
      <out-broadcast-pkts>3</out-broadcast-pkts>
      <out-discards>0</out-discards>
      <out-errors>0</out-errors>
      <out-multicast-pkts>26070</out-multicast-pkts>
      <out-octets>143144868</out-octets>
      <out-unicast-pkts>1424051</out-unicast-pkts>
    </counters>
  </state>
</interface>
```


```
1:"eth1/49"
2:{
  1:{
    1:2
    2:0
    3:0
    4:0
    5:30543
    6:13320913920
    7:5406026
    8:0
    9:3
    10:0
    11:0
    12:26070
    13:143144868
    14:1424051
  }
}
```

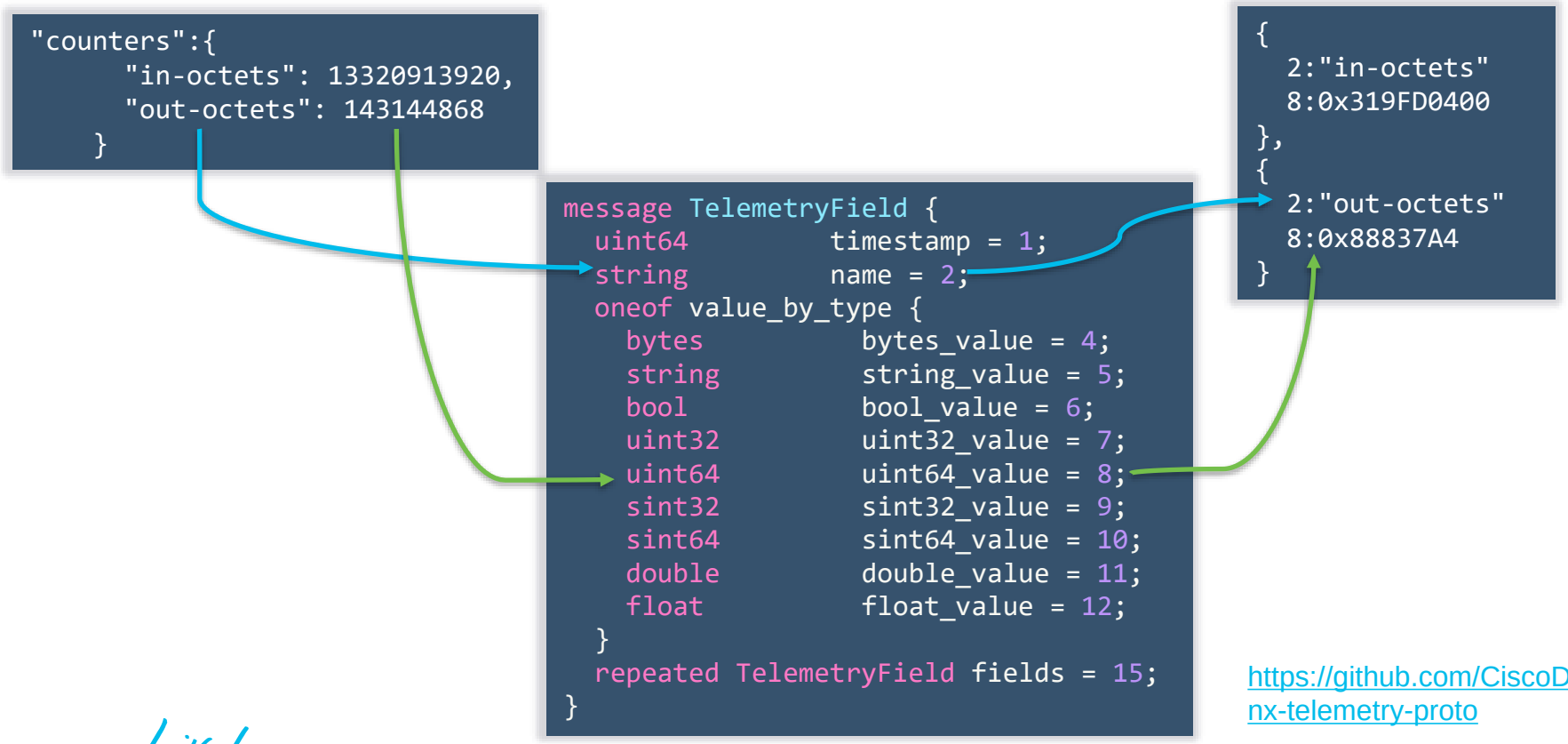
How does GPB(Google Protocol Buffers) work

```
<interface>
  <name>eth1/49</name>
  <state>
    <counters>
      <in-broadcast-pkts>2</in-broadcast-pkts>
      <in-discards>0</in-discards>
      <in-errors>0</in-errors>
      <in-fcs-errors>0</in-fcs-errors>
      <in-multicast-pkts>30543</in-multicast-pkts>
      <in-octets>13320913920</in-octets>
      <in-unicast-pkts>5406026</in-unicast-pkts>
      <in-unknown-protos>0</in-unknown-protos>
      <out-broadcast-pkts>3</out-broadcast-pkts>
      <out-discards>0</out-discards>
      <out-errors>0</out-errors>
      <out-multicast-pkts>26070</out-multicast-pkts>
      <out-octets>143144868</out-oc
      <out-unicast-pkts>1424051</ou
    </counters>
  </state>
</interface>
```


```
1:"eth1/49"
2:{
  1:{
    1:2
    2:0
    3:0
    4:0
    5:30543
    6:13320913920
    7:5406026
    8:0
    9:3
    10:0
    11:0
    12:26070
    13:143144868
    14:1424051
```

High wire efficiency
But hard to develop the encoder and decoder

How does GPB-KV(Key-Value) work



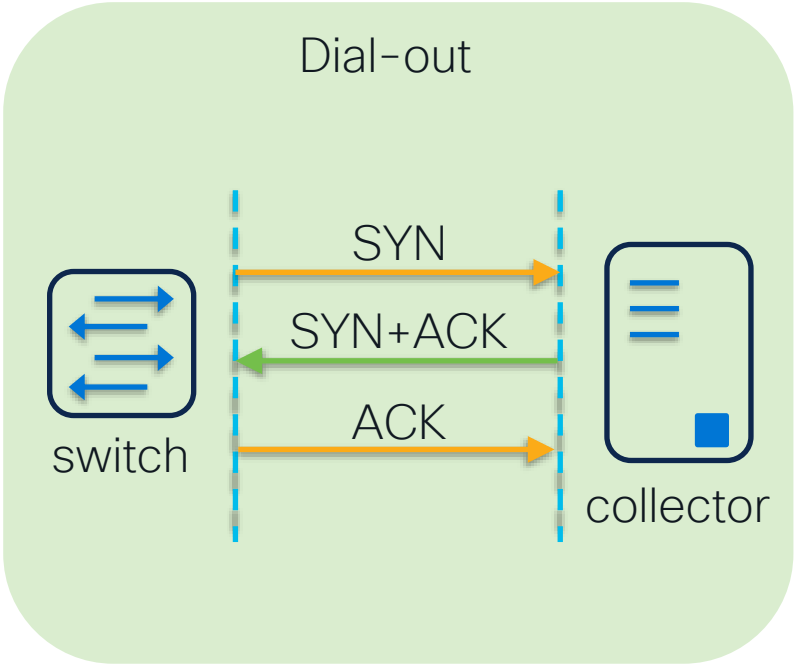
https://github.com/CiscoDevNet/nx-telemetry-proto

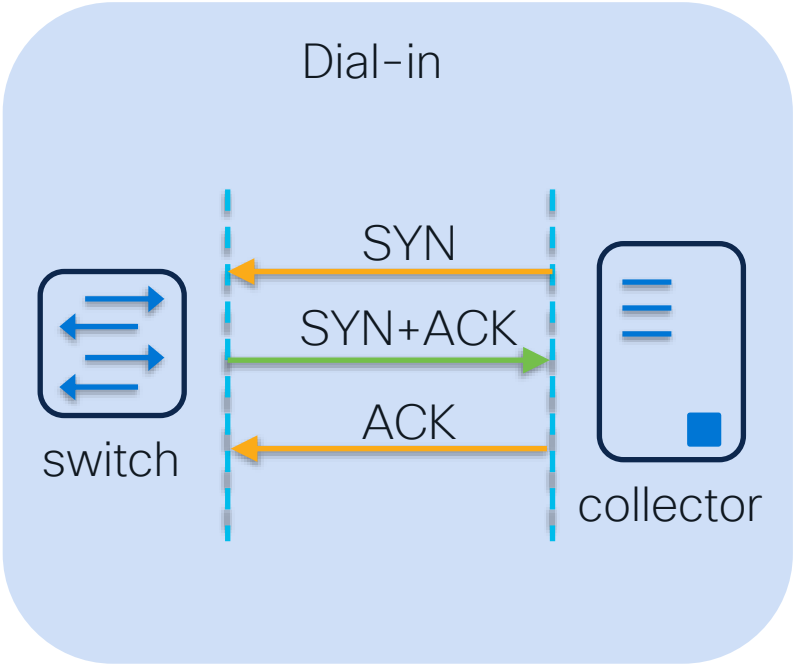
Transport options

- Dial-out vs Dial-in

Dial-out vs Dial-in

- TCP connection is always persistent in telemetry
- The difference is which part initializes the connection



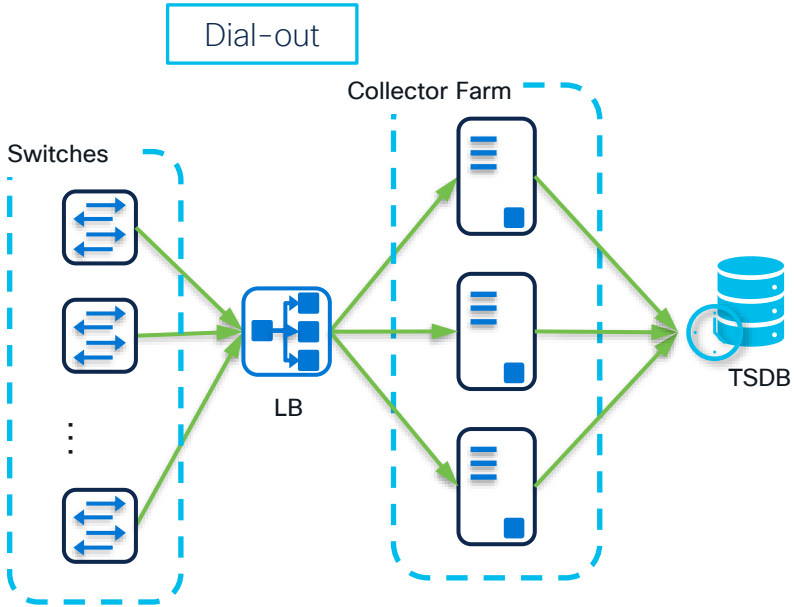


Dial-out vs Dial-in

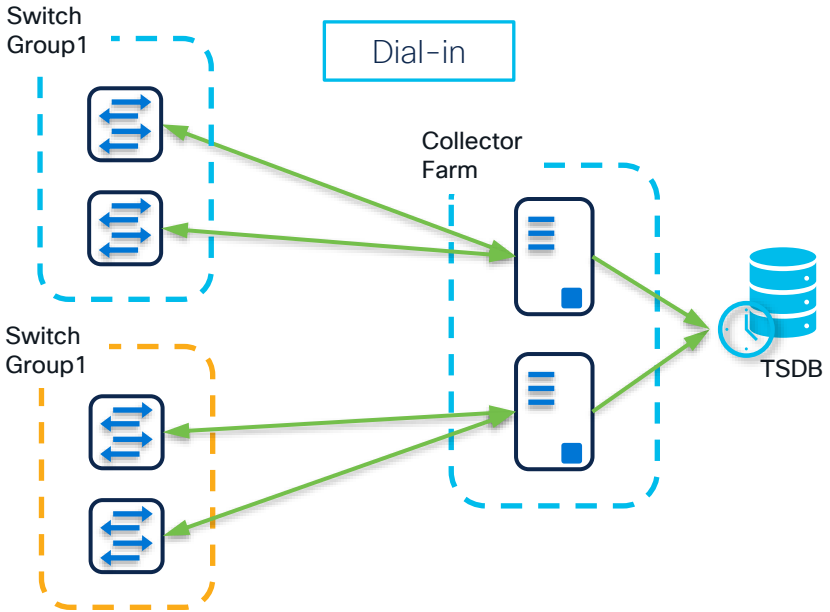
Dial-out	Dial-in
Support gRPC, HTTP, UDP as the transport protocol	Only gNMI is supported as the protocol
Configuration needs to be done from CLI or other management interfaces	Single-channel for subscription and data transport
No need to open a specific port to the management interface of the switch	The firewall rule needs to apply to the ingress direction to switch for gRPC
Load balancing is easy by setting up collector behind VIP	gRPC/gNMI clients need to be distributed between switches

Dial-out vs Dial-in

Design Consideration



Collectors can be set up behind load balancer, all switches stream to the same VIP of collector



To distribute the workload, the collectors need to dial into different switch groups, extra effort to keep the sensor configure synchronized across the cluster

gNMI

gRPC Network Management Interface

gNMI Introduction

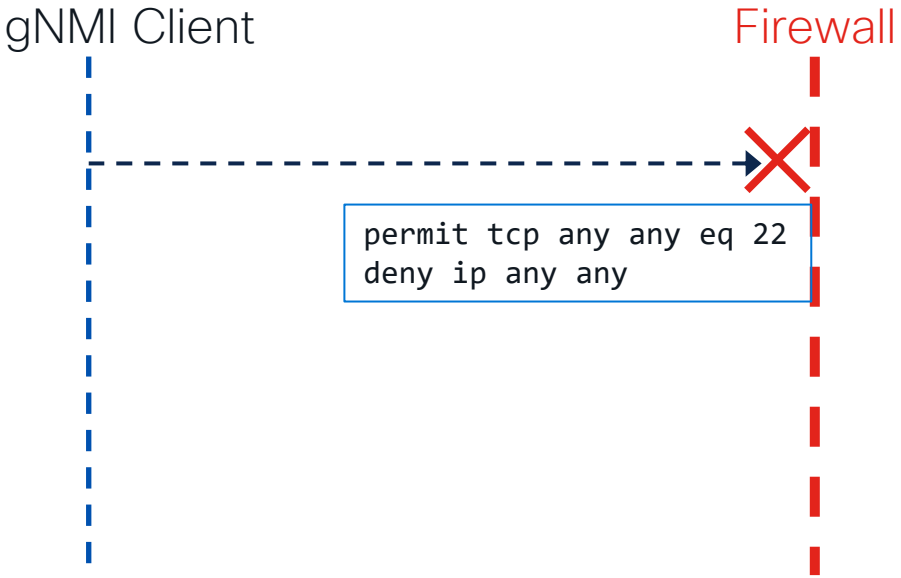
gRPC Network Management Interface

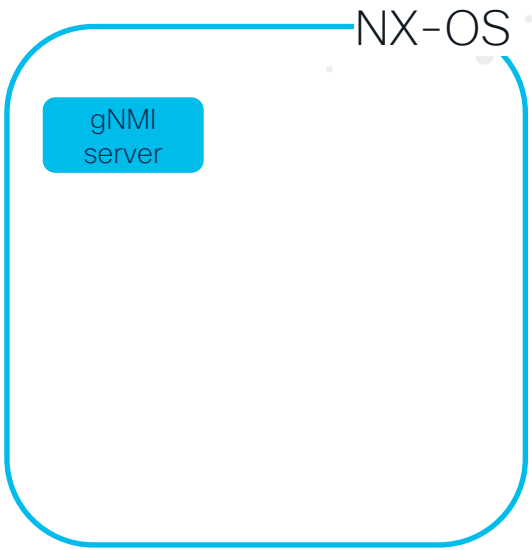
- Built on the gRPC framework
  - Specification of RPCs and behaviors for managing state on the network device
- Supports both configuration management and steaming telemetry
- Design to carry any tree-structured data
- Offers an alternative to NETCONF/RESTCONF

gNMI RPCs

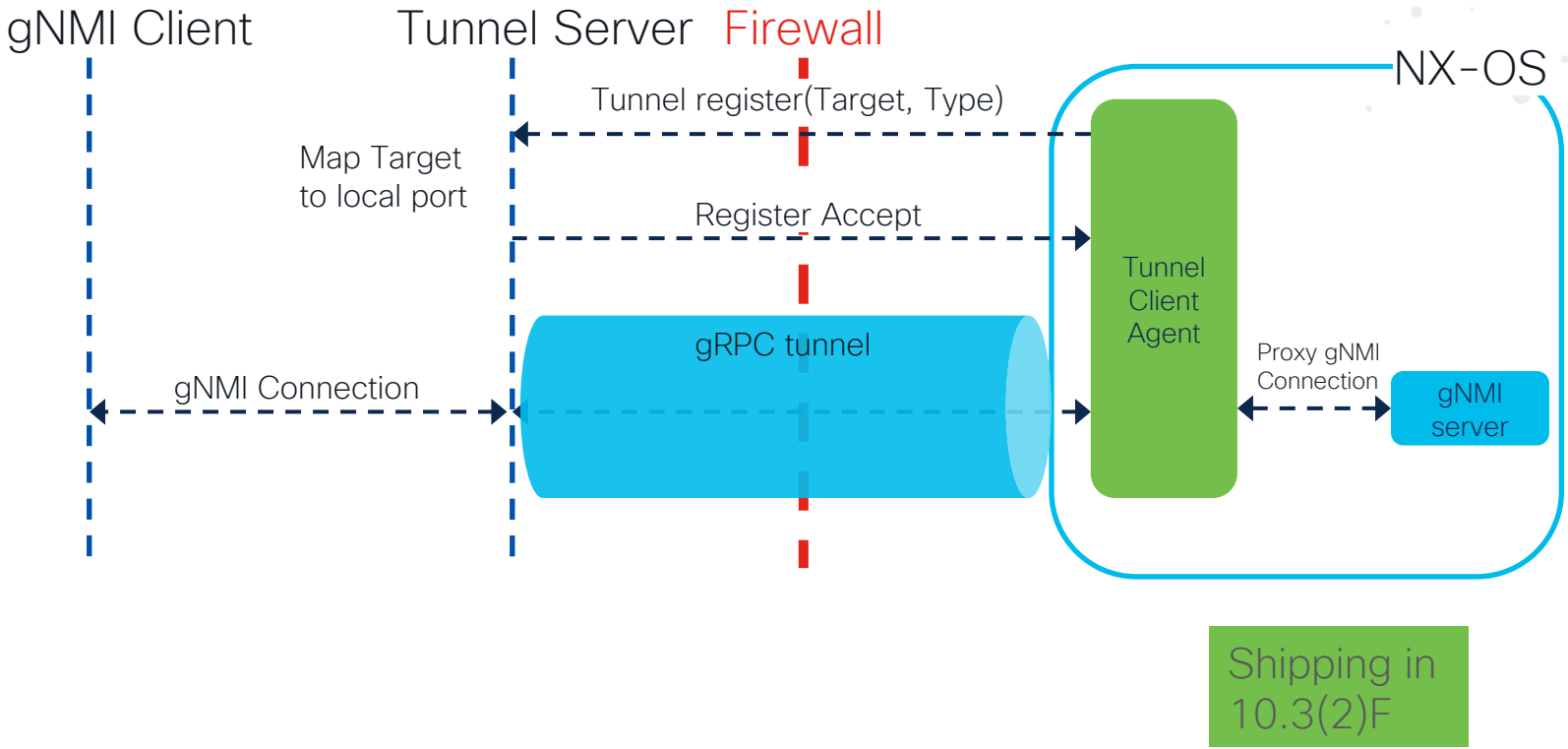
- Capabilities, Retrieve the set of capabilities supported by the target, which usually happened during initial communication
- Get, retrieve a snapshot of data from the target
- Set, Modify the state of data on the target
- Subscribe, Subscribe to a stream of values of paths within the data tree

Firewall doesn't like gNMI





gRPC tunnel

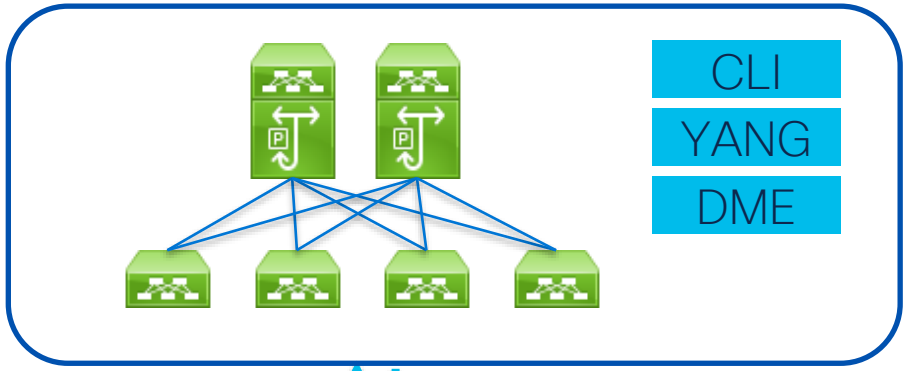


gNMI implementation in NX-OS

Standard	gNMI in NX-OS 9.x is based on version 0.5.0
RPC Capabilities	Complete set of gNMI operation are supported since 9.3(5) Supports both ON_CHANGE and SAMPLE streaming mode `target_defined` is supported in 10.2(1)F `suppress_redundant` and `heatbeat_interval` is supported in 10.2(3)F
Security	TLS is mandatory, supports Mutual TLS
Data Model Encoding	Native and Openconfig Yang Model Supports KV-GPB and JSON as encoding Wild card is supported in 10.2(2)F

How to build a telemetry system with opensource tools

Opensource Software Stack



https://github.com/dsx1123/telemetry_collector

Takeaways

- NX-OS has a wide choice of the data model and streaming transport options, customers can choose based on business requirements
- Most of the customers are interested in gNMI dial-in but there are pros and cons between dial-out and dial-in
- To optimize resource utilization, only stream what you need
- Use GPB-KV when possible
- Use Openconfig YANG models first, fall back to Native YANG mode and DME when data is not available in OC yang

Complete your Session Survey

- Please complete your session survey after each session. Your feedback is important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (open from Thursday) to receive your Cisco Live t-shirt.

- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Session Catalog and clicking the "Attendee Dashboard" at https://www.ciscolive.com/emea/learn/sessions/session-catalog.html

Continue Your Education

Visit the Cisco Showcase for related demos.

Book your one-on-one Meet the Engineer meeting.

Attend any of the related sessions at the DevNet, Capture the Flag, and Walk-in Labs zones.

Visit the On-Demand Library for more sessions at ciscolive.com/on-demand.


cisco
The bridge to possible
Thank you
cisco Live!


CISCO Live!
ALL IN