# **Telemetry Document**

## **Dial Modes**

1. Dial Out

Consider this from the device perspective. In Dial Out, we push the commands from the atom to the device ( via remote agent ) and then the device creates the connection to the atom ( via remote agent )

Its a 1 step process :

1. Provision :- While provisioning, we push the commands to the device which has destination\_group ( Remote Agent IP address and the port ) and sensor path information.
2. Dial In

Consider this from the device perspective. In Dial In, ATOM creates the connection to the device ( via remote agent ) and then the device sends the data to the ATOMction to the atom ( via remote agent )

Its a 2 step process :

1. Provision :- While provisioning, we push the commands to the device which has and sensor path information. It won’t have the destination group info.
2. Deploy : This will establish the connection from ATOM to the device via remote agent and the subscribed sensors data comes to the ATOM Telemetry Engine.

## **Telemetry Flows:**

1. **Subscription Flow:**

//172.16.20.12, 12456, sensor info

1. **Data Incoming Flow:**

ATOM supports 2 device platforms for Telemetry Flows.

1. Juniper - JUNOS
2. Cisco - IOSXR

## **Juniper - JUNOS**

**JUNOS Combinations possible:**  
**DIAL IN - GRPC - self-describing-gpb**

**DIAL OUT - UDP - compact-gpb**

Notes :

A ) Packages reqd.

1) Juniper base pkg

2) juniper cli

3) juniper telemetry driver -- changed

4) juniper telemetry seed data -- changed  -- In dial out - make sure the proto file is there in the pkg wrt device platform version

-- Extended Inventory should pass and OS version and platform should show in device grid

B) Turn off DRY Run

C) Device should be connected to a remote agent

D) Sometimes the device doesn't respond. ping and verify

E) Open udp port in remote agent in firewall

commands to add port: login in to remote agent VM and fire below commands

sudo firewall-cmd --zone=public --add-port=12456/udp --permanent

sudo firewall-cmd --reload

F) Few UI Validations

Negative flows:

1. a device can be associated with only one collection in a particular tenant ( Device 172.16.4.99 already present in collection  )

2. JUNOS: grpc with dial-out - should fail ( POSTMAN ) (grpc only works with dial-in)

3. collection created with device not onboarded ( POSTMAN )

4. collection created with no sensor selected ( POSTMAN )

5. collection created with frequency value 0 ( POSTMAN )

6. Collection name can not have special characters ( "\_\*.<> )

7. Collections with different owners can not be modified/deleted.

G) After testing, It is better to unprovision the collection (because of streaming data flow will fill the logs very fast)

H) Dial out

- Not supported in 5.95, 5.96, 5.89 (vMX )

admin@sr01.rp-5.95> show interfaces terse | match 172.16.5.95

fxp0.0                  up    up   inet     172.16.5.95/24

So right now, don’t use fxp0 as the output interface to the telemetry server. Try to use other physical interfaces on any FPC to try.

Refer : https://community.juniper.net/communities/community-home/digestviewer/viewthread?MID=71485  --> raised by @rajesh

- Supported in 4.99 (Mx204)

I) Dial In

- Supported in 4.99, 5.95, 5.89 ( If proper interface is there )

## **CISCO - IOSXR**

**CISCO Combinations possible-**

**DIAL IN - GRPC - compact-gpb**

**DIAL IN - GRPC - self-describing-gpb**

**DIAL OUT - GRPC - compact-gpb**

**DIAL OUT - GRPC - self-describing-gpb**

**DIAL OUT - TCP - compact-gpb**

**DIAL OUT - TCP - self-describing-gpb**

\*Note\* - Credential Sets should be proper for cisco ( especially grpc port ) - 57300 ( depends on the device config )

A ) Packages reqd.

1) Cisco base pkg

2) Cisco cli

3) cisco telemetry driver -- changed

4) cisco telemetry seed data -- changed  -- In dial out - make sure the proto file is there in the pkg wrt device platform version

-- Extended Inventory should pass and OS version and platform should show in device grid ( in Dial out )

B) Turn off DRY Run

C) Device should be connected to a remote agent

D) Sometimes the device doesn't respond. ping and verify

E) Open udp, tcp, grpc ports in remote agent in firewall ( for dial out )

To check : ( sudo firewall-cmd --list-all )

commands to add port: login in to remote agent VM and fire below commands

sudo firewall-cmd --zone=public --add-port=12456/udp --permanent

sudo firewall-cmd --zone=public --add-port=12455/tcp --permanent

sudo firewall-cmd --zone=public --add-port=12454/tcp --permanent ( grpc )

sudo firewall-cmd --reload

F) Few UI Validations

Negative flows:

1. If the OS version is not present in the device grid ( in dial out ) - Error is thrown ( Device {deviceId} OS Version is not present in device grid. Run Extended Inventory for the device and try again).

2. collection created with device not onboarded ( POSTMAN )

3. collection created with no sensor selected ( POSTMAN )

4. collection created with frequency value 0 ( POSTMAN )

5. Collection name can not have special characters ( "\_\*.<> )

6. Collections with different owners can not be modified/deleted.

G) After testing, It is better to unprovision the collection (because of streaming data flow will fill the logs very fast)

1. In package( seed data) - deploy on telemetry should be ticked

and In package ( telemetry driver ) - deploy on agent should be ticked

2. Problems with the device time ( data older than 5 mins will get dropped from the kafka ) - NTP ( Nw time protocol ) settings - to sync Time zone with clock( every device )

- command to check date/time on device --->  show clock

( In this case, the metrics will be seen, but the data will not be there at correct time)

3. 172.16.18.176 -- supports TCP flow (Credentials : admin/Elastic+123)  -- Not supported for GRPC due to Device lower version

172.16.17.133/172.16.17.134 -- Support TCP and GRPC (Credentials for SSH & GRPC: root/root)

Things to take care before testing telemetry -

4. This should be there with proper interface - show running-config tpa ( for GRPC - NO-TLS )

tpa

 vrf default

  address-family ipv4

   default-route mgmt

   update-source GigabitEthernet0/0/0/0

  !

 !

!

If the above is not there. This will show  ( Error )

sh telemetry model-driven trace all | i 172.16.11.163

Refer : https://xrdocs.io/telemetry//tutorials/ios-xr-telemetry-power-consumption-docker-compose/

e) Command : show telemetry model-driven subscription <collection name> - it should show State as ACTIVE

show telemetry model-driven subscription InterfaceError

**Proposed Addition**

**SNMP :** Simple Network Management Protocol :

1. A networking protocol used for the management and monitoring of network-connected devices in Internet Protocol Networks.
2. Embedded in the local devices such as Routers, Switches, Servers, Firewalls etc using their IP Address.
3. Provides a Common Mechanism for the Network devices to relay management information.
4. An Application Layer Protocol.
5. Implemented using the UDP(User Datagram Protocol) which works like the TCP but assumes the Error-Checking and Recovery Services are not required.
6. SNMP Management Information Bases are Data Structures that define what can be collected from the Local Device, what can be Changed and Configured. MIB may be standardized or be Proprietary by the Vendors.
7. **SNMP Runtime Components :**
   1. **SNMP-Managed Devices and Resources** : Devices and other Network Elements on which an **Agent** runs.
   2. **SNMP Agent** : This is a Software which runs on the Hardware or Service being monitored by the SNMP, collecting Data on various Metrics like the CPU, Bandwidth or Disk Space.
   3. **SNMP Manager** : Works as the Centralized management station running an SNMP management application on many different operating systems.
   4. **Management Information Base(MIB)** : This data structure is a Text File that describes all data objects used by a particular device that can be queried using SNMP.
8. **SNMP Commands :**
   1. **Get Request :** To retrieve the value of a Variable or a list of Variables
   2. **Set Request :** Sent by the SNMP Manager to issue configurations or commands.
   3. **GetNext Request :** Sent by the SNMP manager to agents to find the values of the next record in the MIB’s hierarchy.
   4. **GetBulk Request :**
   5. **SNMP Response :**
   6. **SNMP Trap :**
   7. **SNMP Inform :**
9. **SNMP Port :**
10. **Limitations of SNMP Management :**
    1. Lack of Useful Data : The information these methods collect about the state of the network and performance metric is incomplete. Under a pull based mechanism the device sends
    2. No insight into user experience
    3. SNMP consists of **MIBs** where **the management information is stored in hierarchical structure** with different branches. The MIBs **are standardized but vendor specific**, **when a Vendor adds** their own **fields** the third party network management tool will **need to update**, which **can be tedious**.
    4. **Scaling issues** occur due to a load on Network Management System used in large networks as the System responsible for Network Management gets overwhelmed by the sheer number of Devices it has to poll.
    5. **Security Issues:** Polling based techniques also adds security risk as we need a list of all devices needed along with their IP and security credentials.

**Telemetry :** The collection of measurement or other data at remote points and their automatic transmission to receiving equipment.

**Streaming Telemetry:** A push based mechanism that removes the inefficiencies associated with polling. The required data is streamed automatically and continuously from network device to management systems, without the use of polling. The network device is configured with the type of data to be collected, the frequency of collection and where it should be sent.

Enables access to real time. Model driven and analytic-ready data that can help with Network Automation, Traffic Optimization and preventive troubleshooting.

**Streaming Telemetry Vs SNMP**

**SNMP**

1. SNMP is used best when retrieving relatively static data, such as inventory or neighbouring devices.
2. SNMP is better for Older Deviceswhich do not support Telemetry.
3. SNMP is better for non-performance data e.g. Serial Numbers, Modules and Slot Locations.
4. SNMP uses UDP.

**Streaming Telemetry**

1. Streaming Telemetry is better for collecting High-Resolution performance data such as statistics.
2. RPC(Remote Procedure Calls) mechanism is more efficient than SNMP or CLI in obtaining data from network devices, making telemetry the obvious choice.