Addressing Network Operator Challenges in YANG push Data Mesh Integration
## Evolving YANG Push

### Missing puzzle pieces

<table>
<thead>
<tr>
<th>YANG Push</th>
<th>Today at Network Operators</th>
<th>Today at IETF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport Protocol</td>
<td>Many and non-standard</td>
<td>netconf-https-notif and netconf-udp-notif</td>
</tr>
<tr>
<td>Encoding</td>
<td>JSON widely adopted. Propriety protobuf in various variants. CBOR not implemented yet.</td>
<td>XML in RFC7950, JSON in RFC7951, CBOR in RFC9254</td>
</tr>
<tr>
<td>Subscription</td>
<td>Non-standard, periodical widely adopted. On-change sparse.</td>
<td>RFC8639 and RFC8641</td>
</tr>
<tr>
<td>Versioning</td>
<td>Neither covered in subscription nor in publishing.</td>
<td>netmod-yang-module-versioning</td>
</tr>
<tr>
<td>YANG module</td>
<td>Non-standard widely adopted. IETF coverage non-existent.</td>
<td>Many RFC's defined</td>
</tr>
</tbody>
</table>
YANG datastores enabling Closed Loop Operation

Automated data onboarding with bounded context

Digital Twin

YANG is a data modelling language which will not only transform how we managed our networks; it will transform also how we manage our services.

17 industry leading colleagues from 4 network operators, 2 network and 3 analytics providers, and 3 universities commit on a project to integrate YANG and CBOR into data mesh. IETF 116 public side meeting on Wednesday March 29th 12:00 – 12:45.

Automated networks can only run with a common data model. A digital twin YANG data store enables a comparison between intent and reality. Schema preservation enables closed loop operation. Closed Loop is like an autopilot on an airplane. We need to understand what the flight envelope is to keep the airplane within. Without, we crash.
When Big Data and Network becomes one
Marrying two messaging protocols

Data Mesh

• **Data Mesh** is a big data architecture where different domains can exchange data with a **bounded context and SLO’s** are defined in Data Products. **Same principle as in networks.**

• **Semantics** are needed to describe the data. A **gauge32 is not the same as counter32**. Values can increase or decrease. Needs monotonic increasing counter normalization or not.

• **Versioning** is needed to not only understand that the semantic has changed, but also wherever the new semantic is backward compatible or not. **Preventing to break the data processing pipeline.**

• Hostname, publisher ID, sequence numbers and observation timestamping are needed to **measure loss and delay for SLO’s.**

• **YANG push as defined in RFC8641 is missing** hostname, sequence numbers, observation timestamping and versioning. draft-ahuang-netconf-notif-yang, draft-tgraf-netconf-notif-sequencing, draft-tgraf-yang-push-observation-time and draft-tgraf-netconf-yang-notifications-versioning addresses this.
Define **YANG module** for Netconf Notifications

Closing the semantic gap

- With **RFC 5277** the XML schema for NETCONF event notification was defined.
- With **draft-ahuang-netconf-notif-yang** updates RFC 5277 by defining the schema as a YANG module.
- This enables YANG-push to define semantics for the entire YANG push message and use other encodings than XML such as YANG-JSON RFC 7951 or YANG-CBOR RFC 9264.

```xml
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2023-02-04T16:30:11.22Z</eventTime>
    <id>1011</id>
    <datastore-contents>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
        <interface>
          <name>eth0</name>
          <oper-status>up</oper-status>
        </interface>
      </interfaces>
    </datastore-contents>
  </push-update>
</notification>
```
Define **YANG module** for Netconf Notifications

**Status**

- The yang module prefix has changed to “inotif” to be more explicit.
- The namespace is changed to the one used in RFC5277: urn:ietf:params:xml:ns:netconf:notification:1.0
- In IANA section, instead of asking for a new URI, we ask IANA to add this document as a reference to the URI from RFC5277
- Requesting NETCONF working group adoption.
Extend Streaming Update Notifications with Hostname and Sequencing

For push-update and push-change-update

module: ietf-notification-sequencing

augment-structure /notif:notification:
    +-- sysName           inet:host
    +-- publisherId       yang:gauge32
    +-- sequenceNumber    yang:counter32

Extend Streaming Update Notifications with Hostname and Sequencing

For push-update and push-change-update

- When the NETCONF event notification message is forwarded from the YANG push receiver to another system, such as a messaging system or a time series database where the message is stored, the transport context is lost since it is not part of the NETCONF event notification message metadata. Therefore, the downstream system is unable to associate the message to the publishing process (the exporting router), nor able to detect message loss or reordering.

- draft-tgraf-netconf-notif-sequencing extends the NETCONF notification defined in RFC5277 with:
  - **sysName**: Describes the hostname following the 'sysName' object definition in RFC1213 from where the message was published from.
  - **publisherId**: netconf-distributed-notif describes the ability to publish from network processors directly. With this identifier the publishing process from where the message was published from can be uniquely identified.
  - **sequenceNumber**: Generates a unique sequence number as described in RFC9187 for each published message.
Extend Streaming Update Notifications with Observation Timestamping
For push-update and push-change-update

- **To correlate network data** among different Network Telemetry planes as described in Section 3.1 of RFC9232 or among different YANG push subscription types defined in Section 3.1 of RFC8641, network observation timestamping is needed to understand the timely relationship among these different planes and YANG push subscription types.

- **draft-tgraf-yang-push-observation-time** extends the YANG push streaming update notification defined in RFC8641 with:
  - **observation-time**: Describes the measurement observation time for the "push-update" notification in a "periodical" subscription.
  - **state-changed-observation-time**: Describes in the "push-change-update" notification in an "on-change" subscription the time when the network state change was observed after the subscription was initially established. In case of an "on-change sync on start" subscription it describes the time when the network state change was observed before the subscription was established.
Extend Datastore Selection and Subscription State Change Notifications with revision and revision-label

- Network operators need to control semantics in its data processing pipeline. That includes YANG push.
- This is today only possible during YANG push subscription but not when nodes are being upgraded or messages are being published for configured subscription.
- draft-tgraf-netconf-yang-notifications-versioning extends the YANG push subscription and publishing mechanism defined in RFC8641:
  - By adding the ability to subscribe to a specific revision or latest-compatible-semversion.
  - By extending the YANG push Subscription State Change Notifications Message so that the YANG push receiver learns beside the xpath and the sub-tree filter also the revision and revision-label.
Extend Datastore Selection and Subscription State Change Notifications with revision and revision-label

**Status**

- Moved revision and revision-label from push-update to subscription-start and subscription-modified Subscription State change notification message to reduce message overhead as per suggestion from Jason Stern and Rob Wilton.

- **YANG full tree view added in section 4.1.2**, added descriptions and resolved some issues in the YANG module raised by the YANG validation.

- Fengchong and Rob questioned that revision and revision-label for the subscribed xpath, sub-tree, might be not enough to include all use cases. The authors agree that when the subscribed xpath revision did not change after software upgrade, but the included yang modules did, this needs to be covered as well. pkg-name and pkg-version from draft-ietf-netmod-yang-packages might be the best answer.
From YANG push to Analytics
Next steps

- Do you realize the gaps and how it could be resolved?
  - By defining a YANG module for NETCONF notification and adding hostname, publisher ID, sequence number, observation time, revision and revision-label into YANG push-update and Subscription State Change notification messages an automated data processing pipeline which starts with YANG push, consolidates at Data Mesh and ends at Network Analytics would become at reach.

- What are your thoughts and comments?

- Interested to learn more? Join the IETF 116 public side meeting on Wednesday March 29th 12:00-12:45 in room G301 or look at the project page:


18. March 2023

alex.huang-feng@insa-lyon.fr, pierre.francois@insa-lyon.fr
thomas.graf@swisscom.com
benoit.claise@huawei.com, jean.quilbeuf@huawei.com
Backup
State of the Union
From data mess to data mesh
From YANG push to Analytics
Aiming for an automated processing pipeline

- A network operator aims for:
  - An automated data processing pipeline which starts with YANG push, consolidates at Data Mesh and ends at Network Analytics.
  - Operational metrics where IETF defines the semantics.
  - Analytical metrics where network operators gain actionable insights.

- We achieve this by integrating YANG push into Data Mesh to:
  - Produce metrics from networks with timestamps when network events were observed.
  - Hostname, Publisher ID and sequence numbers help us to understand from where metrics were exported and measure its delay and loss.
  - Forward metrics unchanged from networks
  - Learn semantics from networks and validate messages.
  - Control semantic changes end to end.
Evolving Big Data Architecture
Domain oriented, like networks

1st Generation
- Proprietary
- Enterprise Data Warehouse

2nd Generation
- Data lake
- Big data ecosystem

3rd Generation
- Kappa
- Adds streaming for real-time data

4th Generation
- Data Mesh
- Distributed and organized in domains.

From Principles to Logical Architecture

Data Plane
- Domain A
  - Collect
  - Publish
  - Serve
- Domain B
  - Collect
  - Publish
  - Serve
- Domain C
  - Collect
  - Publish
  - Serve

Data Infra as a Platform
- Operational Delivery Platform
- Analytical Data Platform

Federated Computational Governance for interoperability
1. A single link down results in multiple device topology, control-plane and forwarding-plane events being exposed at different times.

2. Determine which interfaces and BGP peerings are being used first and then observe state. Observe BGP withdrawals and updates, traffic drop spikes and missing traffic. Generate multiple concerns.

3. Calculate for each observation a concern score between 0 and 1. The higher, the more probable the changes impacted forwarding.

4. Unify several concerns for one VPN connectivity service to one alert identifier.
L3 VPN Network Anomaly Detection
Verify operational changes automatically

**Analytical Perspectives**
Monitors the network service and wherever it is congested or not.

> BGP updates and withdrawals.
> UDP vs. TCP missing traffic.
> Interface state changes.

**Network Events**
1. VPN orange lost connectivity.
   VPN blue lost redundancy.
2. VPN blue lost connectivity.

**Key Point**
> AI/ML requires network intent and network modelled data to deliver dependable results.