Network Operator Challenges in Network Telemetry Data Mesh Integration
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.
From Network Telemetry to Analytics
Aiming for an automated processing pipeline

• A network operator aims for:
  • An **automated data processing pipeline** which starts with Network Telemetry, 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 Network Telemetry into Data Mesh to:
  • Forward **metrics unchanged** from networks
  • **Learn semantics** from networks and validate messages.
  • **Control semantic** changes end to end.
State of the Union

From data mess to data mesh
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
## 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 in draft status</td>
</tr>
<tr>
<td>Encoding</td>
<td>JSON widely adopted. Propriety protobuf in various variants. CBOR not implemented yet.</td>
<td>JSON and XML in RFC8040, 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>
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.

**News:** 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. Starts with IETF 115 public side meeting on Monday November 7th 2022 at 13:30.

**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.
Data Mesh

- Data Mesh is a big data architecture where different domains can exchange data with a bounded context. Same principle as in networks.
- Bounded context relies that data always has semantics and versioning.
- 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 and metadata are 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.
- YANG push as defined in RFC8641 is missing semantics and versioning. draft-tgraf-netconf-yang-notifications-versioning addresses both.
Extending the Datastore Selection and Notification Message Metadata
draft-tgraf-netconf-yang-notifications-versioning Introduction

module: ietf-yang-push-metadata

augment /yp:push-update:
  +--ro module? string
  +--ro namespace? string
  +--ro revision? rev:revision-date-or-label
  +--ro revision-label? yver:version
  +--ro datastore-xpath-filter? yang:xpath1.0 {sn:xpath}?
  +--ro datastore-subtree-filter? <anydata> {sn:subtree}?

augment /yp:push-change-update:
  +--ro module? string
  +--ro namespace? string
  +--ro revision? rev:revision-date-or-label
  +--ro revision-label? yver:version
  +--ro datastore-xpath-filter? yang:xpath1.0 {sn:xpath}?
  +--ro datastore-subtree-filter? <anydata> {sn:subtree}?

augment /sn:establish-subscription/sn:input/sn:target:
  +-- revision? rev:revision-date-or-label
  +-- revision-label? yver:version
augment /sn:modify-subscription/sn:input/sn:target:
  +-- revision? rev:revision-date-or-label
  +-- revision-label? yver:version
augment /sn:subscription-started/sn:target:
  +-- revision? rev:revision-date-or-label
  +-- revision-label? yver:version
augment /sn:subscription-modified/sn:target:
  +-- revision? rev:revision-date-or-label
  +-- revision-label? yver:version
augment /sn:subscriptions/sn:subscription/sn:target:
  +--rw revision? rev:revision-date-or-label
  +--rw revision-label? yver:version

• 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.

draft-tgraf-netconf-yang-notifications-versioning extends the YANG push subscription and publishing mechanism defined in RFC8641:

• By extending the YANG push header so that the YANG push receiver learns the semantic reference in the notification message directly.

• By adding the ability to subscribe to a specific revision or latest-compatible-semversion.
Example Notification Message
In JSON and XML encoding

```json
{
    "ietf-notification:notification": {
        "eventTime": "2022-09-02T10:59:55.322",
        "ietf-yang-push:push-update": {
            "id": 101,
            "module": "ietf-interfaces",
            "revision": "2014-05-08",
            "revision-label": "1.0.0",
            "datastore-xpath-filter": "ietf-interfaces:interfaces",
            "datastore-contents": {
                "ietf-interfaces:interface": {
                    "name": {"eth0": {
                        "oper-status": "up",
                    },
                    }
                }
            }
        }
    }
}
```

```xml
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2022-09-02T10:59:55.322</eventTime>
    <id>101</id>
    <module>ietf-interfaces</module>
    <revision>2014-05-08</revision>
    <revision-label>1.0.0</revision-label>
    <datastore-xpath-filter>ietf-interfaces:interfaces</datastore-xpath-filter>
    <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>
```
From Network Telemetry to Analytics

Next steps

- Do you realize the gaps and how it could be resolved?
  - By adding semantic reference in YANG push and in Data Mesh, an **automated data processing pipeline** which starts with Network Telemetry, consolidates at Data Mesh and ends at Network Analytics would become at reach.

- **Collaborate** with different network operators, network and analytic vendors and universities on bringing YANG semantics into Apache Kafka.

- Adding semantic reference into YANG push at IETF.

- -> What are your thoughts and comments?

- -> Interested to learn more? Join the IETF 115 public side meeting on Monday November 7\(^{th}\) at 13:30 or look at the project page:
  

thomas.graf@swisscom.com

05. November 2022