STATUS QUO OF DYNAMIC NETWORK MANAGEMENT WITH YANG-BASED CONFIGURATION MODELS

Dr. Christian Herber, Principal Software Architect
10TH IEEE SA ETHERNET & IP @ AUTOMOTIVE TECHNOLOGY WEEK (E&IP@ATD) 2021
Motivation

- **Zonalization**
  - Large Ethernet backbone networks
  - Increasing number of switches and hops
  - Co-existence of heterogeneous traffic
  - Run-time changes for networking configuration
  - Dynamic resource management
  - Fail-over handling
  - Predictive monitoring

- **Service-Orientation**
  - Trend to service-based communication
  - Field-upgradability of features

- **Fault Tolerance**
  - Reaction to faults at run-time
  - Heterogenous redundancy in the network

  ➔ Efficient programmatic interface for network configuration is needed

  ➔ YANG models can be used as a common language to describe devices and data
WHAT IS YANG?

• Modeling language designed for configuration, state, notifications and RPCs of a networking device

• Standardized by the IETF as RFC 7950 as a response to the heterogenous management interfaces in the market

• YANG models for various standards and devices are defined by
  - IETF
  - IEEE
  - MEF, ETSI, BBF, ODP, …
  - Vendors like Cisco, Fujitsu, Juniper, Huawei, Nokia, Ciena …

• Data instances of the YANG models can be used for dynamically managing switches, routers etc. using SDN protocols like NETCONF/RESTCONF
SDN ARCHITECTURE ACCORDING TO IEEE 802.1Qcc

Communication of talkers/listeners with CUC not in scope of standard. Could be SRP or SDN protocol.

TSN UNI – being defined in IEEE 802.1Qd.
CUC and CNC can communicate over e.g. RESTCONF.

Remote network management protocol.

1. CNC discovers topology and bridge capabilities.
2. CUC learns and accumulates talker/lister requirements.
3. CUC sends accumulated stream list to CNC.
4. CNC configures TSN features in each bridge.

CUC: Centralized User Configuration
CNC: Centralized Network Configuration
IEEE 802.1Qdj-d0-1 AND YANG: USER/NETWORK CONFIGURATION INFO (UNI)

- Module is using definitions (groupings) defined in Qcc
- Result is a big list of streams, with requests and status

Import TSN types model (defined in Qcc)

Actual configuration properties are reused from TSN types model
IEEE 802.1Qcc TSN TYPES (FOR REFERENCE)

Model is too extensive to visualize in a presentation → snapshot
IEEE 802.1QCC TSN TYPES – TALKER MODEL (PARTIAL)

Requirements for redundancy and latency

Selection strict priority/CBS

If this container exists, each bridge along the path must be time-aware
Used to configure the time-aware shapers

Bandwidth requirements from frame size and interval

Requirements for redundancy and latency
# Published IEEE YANG Models (Partial)

<table>
<thead>
<tr>
<th>IEEE project</th>
<th>Description</th>
<th>YANG Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.1Qcc-2018</td>
<td>Stream Reservation Protocol (SRP) Enhancements and Performance Improvements</td>
<td>• ieee802-dot1q-tsn-types.yang</td>
</tr>
</tbody>
</table>
| 802.1Qcp-2018 | YANG Data Model | • ieee802-dot1q-bridge.yang  
• ieee802-dot1q-types.yang (e.g. definition of a VLAN ID)  
• ieee802-types.yang (e.g. definition of MAC address)  
• ... |
| 802.1Qcr-2020 | Asynchronous Traffic Shaping | • ieee802-dot1q-ats.yang  
• ieee802-dot1q-stream-filters-gates.yang (part of Qci, but was needed for Qcr) |
| 802.1Qcx-2020 | YANG Data Model for Connectivity Fault Management | • Updates to all models of the Qcp project  
• ... |
| 802.1X-2020 | Port-Based Network Access Control | • ieee802-dot1x-types.yang  
• ieee802-dot1x.yang |
| 802.3.2-2019 | YANG Data Model Definitions | • ieee802-ethernet-interface.yang  
• ... |
## DRAFT IEEE YANG MODELS

<table>
<thead>
<tr>
<th>IEEE</th>
<th>Description</th>
<th>Stage</th>
</tr>
</thead>
</table>
| 802.1Qcw| YANG Data Models for Scheduled Traffic, Frame Preemption, and Per-Stream Filtering and Policing  
  • `ieee802-dot1q-preemption.yang`  
  • `ieee802-dot1q-psfp.yang`  
  • `ieee802-dot1q-sched.yang` | Draft  |
| 802.1Qdj| Configuration Enhancements for Time-Sensitive Networking  
  • `https://1.ieee802.org/tsn/802-1qdj/` | Draft  |
| 802.1ABcu| LLDP YANG Data Model  
  • `ieee802-dot1ab-types.yang`  
  • `ieee802-dot1ab-lldp.yang` | Draft  |
| 802.1AEdk| MAC Privacy protection  
  • `ieee802-dot1ae.yang`  
  • `ieee802-dot1ae-types.yang` | Draft  |
| 802.1ASdn| YANG Data Model (depends on YANG model from 1588e) | PAR    |
| 802.1CBcv| Information Model, YANG Data Model and Management Information Base Module  
  • `ieee802-dot1cb-frer.yang`  
  • `ieee802-dot1cb-stream-identification-types.yang`  
  • `ieee802-dot1cb-stream-identification.yang` | Draft  |
| 802.1CBdb| FRER Extended Stream Identification Functions  
  • `ieee802-dot1cb-mask-and-match.yang` | Draft  |
| 1588e   | MIB and YANG Data Models  
  • `ieee1588-tpm.yang` | Draft  |
YANG STATUS INTERMEDIATE SUMMARY

• IEEE standardization of YANG models is on a very good track
• As of today, it is possible to build a full TSN system with the existing (draft) models
• It will take a small number of years to reach full standardized situation

• However, there is more than just networking features based on IETF and IEEE
  - AUTOSAR, Avnu Alliance, and Open Alliance all specify additional networking features which are not covered by existing models

In addition to models, more is needed for a successful YANG ecosystem (tools, Software Development Kits, …)
  → For dynamic use cases, a transport mechanism for YANG data is needed
SDN ARCHITECTURE – YANG DATA TRANSPORT

SDN protocol needs to be used, if CUC and CNC are not on the same device

From 802.1Qcc: “One protocol explored for the UNI between CUC and CNC is RESTCONF”

From IEEE 802.1Qcc: “Examples of a remote network management protocol include Simple Network Management Protocol (SNMP), NETCONF, and RESTCONF”

SNMP is outdated and does not support YANG

⇒ Next slides show introduce the YANG capable SDN protocols
Models are the common denominator and root of SDN protocols discussed here.

Concrete data instances need to be described in a machine-readable format.

Provide mechanism to retrieve state and modify configuration.

Data transfer

Mechanism for (secure) transport.
**X-CONF COMPARISON**

<table>
<thead>
<tr>
<th></th>
<th>NETCONF</th>
<th>RESTCONF</th>
<th>CORECONF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year of Standardization</td>
<td>2006</td>
<td>2017</td>
<td>&gt;= 2021</td>
</tr>
<tr>
<td>Transport Layer</td>
<td>(SSH)/TLS</td>
<td>HTTPS</td>
<td>DTLS</td>
</tr>
<tr>
<td>Header format</td>
<td>XML</td>
<td>XML/JSON</td>
<td>Binary</td>
</tr>
<tr>
<td>Payload format</td>
<td>XML</td>
<td>XML/JSON</td>
<td>CBOR</td>
</tr>
<tr>
<td>Datastores</td>
<td>All</td>
<td>Running only</td>
<td>Running only</td>
</tr>
<tr>
<td>Distributed Configuration</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Locking</td>
<td>yes</td>
<td>no</td>
<td>No</td>
</tr>
<tr>
<td>Capability Discovery</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>REST interface</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

- Selection of NETCONF/RESTCONF is probably a question of preference
  - NETCONF has a longer history and better support
  - If you like web programming, REST, and json, then RESTCONF is better suited
- CORECONF enables SDN style of operations for constrained devices like microcontrollers
CONCLUSIONS AND SUMMARY

- **SDN**: Current trends like zonalization, service orientation, and fault-tolerance lead to the need for programmatic access to the network’s configuration and state
- **YANG**: Data modeling language perfectly fits this use case, and is widely adopted by standardization bodies
- **IEEE & YANG**: Great progress on TSN modelling. While not all models are finalized, full prototypes can be built with the existing draft models
- **SDN & YANG**: Good scale of SDN management protocols available from big application processors to constrained microcontrollers

Next steps:
- Definition of Automotive use-cases and requirement
- Closing gaps in models where necessary
- Build the infrastructure needed for YANG deployment