SDN and metrics from the SDOs:
Report from Benoit Claise (IETF OPS Area Director)

Dave Ward
dward@cisco.com
2017.04.06
Open Source and SDOs Landscape

Automation of Network + Infrastructure + Cloud + Apps + IOT
Why Data Model-Driven Management?

• APIs derived from the data models:
  • Data models = definitions and constraints
  • The protocol: NETCONF, RESTCONF, GRPC
  • The encoding: JSON, XML, protobuf
  • The programming language: Python, Ruby, Java, C, Erlang, ...

• Industry focusing on YANG as the data modeling language for services and devices
Data Models Driven Set of APIs
Data Model-Driven Management

• Scripting: easy to create, hard to maintain/clean-up
  => Data model-driven set of APIs

Data Models = APIs

• However,

Automation is as good as your data models and your toolchains
Generation of Model-Driven APIs Using YANG Development Kit (YDK)

YDK

Data Models (YANG) → YDK-gen → API Generator → YDK-Py

Python Docs
C++ Docs
Ruby Docs
go Docs
C# Docs
Data Model Language (schema language)

Data Modeling (schema)

Encoding (serialization)

Protocol

Application

Prog. Language

- YANG
- YANG Data Model
- XML
- JSON
- ProtoBuf
- Thrift
- NETCONF
- RESTCONF
- GRPC (HTTP/2)
- YANG Development Kit
- Python
- C++
- Any language

Non Standard Possible links
Data Model Driven Management: Example

Acting on resources

Module `my-interfaces` {
  {
    namespace "com.my-interfaces";
    container `interfaces` {
      list `interface` {
        key `name`;
        leaf `name` { type string; }
        leaf `admin-status` { type enum; }
      }
      rpc `flap-interface` {
        input {
          leaf `name` { type string; }
        }
        output {
          leaf `result` { type boolean; }
        }
      }
    }
  }
}

GET : Gets a resource
GET /restconf/data/my-interfaces:interfaces
GET /restconf/data/my-interfaces:interfaces/interface/<some name>

POST : Creates a resource or invoke operation
POST /restconf/operations/my-interfaces:flap-interface
+ JSON/XML Form Data (including name)
Response will have JSON/XML result

PUT : Replaces a resource
PUT /restconf/data/my-interfaces:interfaces/interface/<some name>
+ JSON/XML Form Data (name, admin-status)

DELETE : Removes a resource
DELETE /restconf/data/my-interfaces:interfaces/interface/<some name>
RESTCONF versus NETCONF: Summary

- RESTCONF: no notion of transaction
- RESTCONF: no notion of lock
- RESTCONF: no notion of candidate config and commit
- RESTCONF: so no notion of two phase commit
- RESTCONF: no <copy-config>
- RESTCONF: some more granularity for query => "config", "nonconfig", "all".
- RESTCONF: XML or JSON (while NETCONF is XML only)

NETCONF might be better for router and switches
RESTCONF might be better for controller north-bound interface
YANG Tsunami in the Industry
Data Model Location and Type

- Native models
- Standard models
- Proprietary extensions to standard models
- Operator defined services
IETF: Timeline of Important Specifications

<table>
<thead>
<tr>
<th>Year</th>
<th>Specification</th>
<th>RFC</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>YANG 1.0</td>
<td></td>
<td>October 2010</td>
</tr>
<tr>
<td></td>
<td>Interface and IP Modules</td>
<td>7223, 7277</td>
<td>May, June 2014</td>
</tr>
<tr>
<td></td>
<td>Common YANG Data Types</td>
<td>6991</td>
<td>July 2013</td>
</tr>
<tr>
<td></td>
<td>JSON Encoding</td>
<td>7951</td>
<td>August 2016</td>
</tr>
<tr>
<td></td>
<td>Routing Management</td>
<td>8022</td>
<td>November 2016</td>
</tr>
<tr>
<td></td>
<td>NETCONF 1.0, SSH Mapping</td>
<td>4741, 4742</td>
<td>December 2006</td>
</tr>
<tr>
<td></td>
<td>NETCONF 1.1</td>
<td>6241</td>
<td>June 2011</td>
</tr>
<tr>
<td></td>
<td>NETCONF Access Control</td>
<td>6536</td>
<td>March 2012</td>
</tr>
<tr>
<td></td>
<td>NETCONF over TLS + x.509</td>
<td>7589</td>
<td>October 2016</td>
</tr>
<tr>
<td></td>
<td>RESTCONF Protocol</td>
<td></td>
<td>January 2017</td>
</tr>
</tbody>
</table>

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Public
IETF: YANG Models Growth

New YANG validator
Active help to authors
pyang 1.6 (YANG 1.1)
Tool: pyang patch

http://claise.be/IETFYANGPageCompilation.png
# IETF Drafts with YANG Modules

## Vendors:
- Cisco: 74
- Huawei: 71
- Juniper: 46
- Ericsson: 25
- Ciena: 14
- ZTE: 8
- Brocade: 6
- Infinera: 6
- Metaswitch: 5
- Alcatel-Lucent: 4
- Intel: 1

## Operators:
- Orange: 13
- Telefonica: 6
- DT: 4
- Level 3: 4
- Google: 2
- Verizon: 2
- AT&T: 2
- China Mobile: 2
- China Unicom: 1
- Softbank: 1

## “YANG” companies:
- Yumarkworks: 6
- Tail-f: 5
IETF Focus

• YANG language definition

• Protocols (NETCONF, RESTCONF) and encoding (JSON, XML)

• Data Model-driven Telemetry with Pub Sub

• Network elements / technology YANG models
  • Main focus: routing

• Only two service YANG models (by choice):
  • L3VPN, with RFC 8049
  • L2VPN chartered

• Location
  • RFC-based YANG modules
  • Draft-based YANG modules (extracted from drafts!)
IETF Focus: Pub/Sub (Telemetry)

<table>
<thead>
<tr>
<th><strong>Subscribed Notifications</strong></th>
<th><strong>YANG Datastore Push</strong></th>
<th><strong>YANG Notifications2</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Dynamic &amp; Configured subscriptions</td>
<td>• Datastore on-change and periodic triggers</td>
<td>• Encapsulation Headers objects: Signature, de-duplication, severity, originator</td>
</tr>
<tr>
<td>• Multiple subscriptions / transport</td>
<td>• Filtering objects within a notification</td>
<td>• Bundled records and record types</td>
</tr>
<tr>
<td>• Multiple configured receivers</td>
<td>• Authorization model per object</td>
<td></td>
</tr>
<tr>
<td>• Establish, modify, delete, kill RPC</td>
<td>• Sending of full YANG trees or yang-patch</td>
<td></td>
</tr>
<tr>
<td>• State change notifications</td>
<td>• Tagging of partial updates</td>
<td></td>
</tr>
<tr>
<td>• Suspend/resume</td>
<td>• Tagging of on-change object support</td>
<td></td>
</tr>
<tr>
<td>• Filtering full notifications</td>
<td>• Negotiation of filters and period lengths</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• More error responses</td>
<td></td>
</tr>
</tbody>
</table>

- Stream discovery
- Replay (and start time negotiation)
- Prioritization
- Monitoring / reporting
- QoS
- Error responses

**NETCONF Transport for Subscribed Notifications**
- Transport mapping

**RESTCONF & HTTP2 Transport for Subscribed Notifications**
- Transport mappings (including HTTP2 QoS)
- Heartbeats and clean-up
IEEE

• **802.1**
  - Two active YANG projects in progress: 802.1Xck (Port-Based Network Access Control Amendment: YANG Data Model) and 802.1Qcp (Bridges and Bridged Networks Amendment: YANG Data Model)
  - Initial work on a LAG (802.1AX) YANG model has been started as part of the next amendment to 802.1AX.

• **802.3**:  
  - Approved a study group: Standard for Ethernet YANG Data Model Definitions

• Regular IETF/IEEE coordination calls

• Location  
  - https://github.com/YangModels/yang/tree/master/standard/ieee
Broadband Forum

• 192 YANG modules
  • Common: subscriber, interfaces, ethernet, PPPoE, etc.
  • Fiber to the Distribution Point (FTTdp) Distribution Point Unit (DPU): FastDSL (FAST/VDSL container), G.hs (FAST/VDSL selection), G.fast, VDSL, MELT+SELT (line tests)
  • Passive Optical Networking (PON):
    • SDN access node

• Participation to the IETF hackathon
  • Integration with the YANG catalog
  • Updated their licensing terms: all work-in-progress YANG modules now in github, to be picked up this toolchain

• Location: https://github.com/YangModels/yang/tree/master/standard/bbf
CableLabs

- CCAPv3.1 system configuration and state
- RPHY Configuration
- Location:
  - http://mibs.cablelabs.com/YANG/DOCSIS/
- The new YANG Models developed by the VPI WG will also be published at the same location in a couple of weeks.
OPENCONFIG

• Operators-led YANG models
  • Google, AT&T, British Telecom, Microsoft, Facebook, Comcast, Verizon, Level3, Cox Communications, Yahoo!, Apple, Jive Communications, Deutsche Telekom / TeraStream, Bell Canada

• Focus: 93 network elements YANG models
  • Routing (BGP, ISIS, rib, network-instance), routing policy, interfaces
  • Layer2 (vlan, spanning tree), ACL, optical transport, mpls, etc.

• Some YANG models not completely aligned with the IETF

• Location:
  • https://github.com/openconfig/public
OPENCONFIG

- Streaming Telemetry specifications and configuration
- gRPC Network Management Interface (gNMI)
  - Protocol: gRPC
  - Encoding: protobuf
DMTF

• Redfish: goal of addressing all the components in the data center with a consistent API

• Goal: use the networking YANG models developed in the IETF, convert them to CSDL (common schema definition language: a JSON encoding of Odata)
  • YANG to CSDL specifications, to convert the IETF networking YANG models.

• Goal: YANG device profile for data center network switches - top-of-rack switches
MEF

- two standard device-level YANG Models (quite old, proposal to update them)
  - MEF 38 is Service OAM Fault Monitoring (CFM)
  - MEF 39 is Service OAM Performance Measurement (G.8013/Y.1731 PM)

- Open projects to create YANG models for:
  - Carrier Ethernet services, i.e. API from business apps to service orchestrator ("legato"), for EVC based services as defined in MEF 10.3 and OVC based services as defined in MEF 26.2
  - Network Resource Provisioning, i.e. API from Service Orchestrator to Domain Controller ("presto")
  - IP Services expected to eventually augment IETF L3SM model
  - Note: YANG for inter-provider APIs also expected.

- Location (for MEF members):
  - https://wiki.mef.net/display/MTA/YANG+Modules+for+MEF+Services
  - Github
Open ROADM

• The **Open ROADM Multi-Source Agreement (MSA)** defines interoperability specifications for Reconfigurable Optical Add/Drop Multiplexers (ROADM). Included are the ROADM switch as well as transponders and pluggable optics. Specifications consist of both Optical interoperability as well as YANG data models.

• [Open ROADM MSA - Home](#)

• YANG modules: [GitHub - OpenROADM/OpenROADM_MSA_Public](#)
sysrepo

• Open-source framework that brings NETCONF/YANG management to any UNIX/Linux application

• Ex: dnsmasq management via NETCONF using Sysrepo: http://www.sysrepo.org/dnsmasq-demo

• Location:
  • https://github.com/sysrepo/sysrepo
SDOs Alignement and Trajectory

Network Service YANG data models
- VPWS - L2VPN
- VPLS - L2VPN
L3VPN

Network Element YANG data models
MPLS  BGP  IPv4 & IPv6  Ethernet
Open Source and SDOs Landscape

Automation of Network + Infrastructure + Cloud + Apps + IOT
Coordination is Really Required, now!
Coordination is Really Required, Now!

• Previous picture is about the IETF YANG models
  • New dimensions: different SDOs/Opensource projects
  • New dimension: versioning

• These YANG models must work together to create services
• Good problem to have: All YANG models arrive at the same time
  • As opposed to MIB modules in the past

• Standard Development Organizations (SDOs) can't work in isolation: industry wide coordination is required
How to Organize the industry?

• With a YANG catalog!

• The related metadata regarding implementation, maturity level, model type, etc (based on the openconfig catalog as a starting point)

• The inventory of all YANG modules, cross SDOs, cross vendors
  • SDOs on board: IETF, BBF, IEEE, etc.
  • Some vendors on board

• Just started: Output of the IETF hackathon, two weeks ago
YANG DB Search

Enter your search term(s) below:

vrf

Search Options

- Case-Sensitive
- Regular Expression
- Include MIBs

Schema Types

- All
  - Typedef
  - Identity
  - Container
  - Leaf
  - Grouping
  - Extension
  - List
  - Notification
  - Feature
  - RPC
  - Leaf-List

Search! Reset
# YANG Catalog: Search

## YANG DB Search Results for 'vrf'

<table>
<thead>
<tr>
<th>Name</th>
<th>Revision</th>
<th>Schema Type</th>
<th>Path</th>
<th>Module</th>
<th>Origin</th>
<th>Organization</th>
<th>Maturity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vrf-routing-instance</td>
<td>2015-05-25</td>
<td>identity</td>
<td>/rt.vrf-routing-instance</td>
<td>ietf-routing</td>
<td>Industry Standard</td>
<td>ietf</td>
<td>This identity represents a VRF routing instance. The type is distinct from the default-routing-instance. There may be multiple vrf-routing-interfaces.</td>
<td></td>
</tr>
</tbody>
</table>
YANG Catalog: Dependency
YANG Catalog: Code Generation

- **NetConf**
- **RestConf**

```
import lxml.etree as ET
from argcomplete import import ArgumentParser
from noolclient import import RPCError

payload = ""
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:urn="urn:ietf:params:xml:ns:yang:ietf-interfaces"><interfaces>
  <interface>
    <name>eth0/0</name>
    <description>new interface</description>
    <type>ietf-ethernet-MCLag</type>
    <admin-status>true</admin-status>
    <status>true</status>
  </interface>
</interfaces>
</config>
```

---

RPC | Script | Capabilities
Open Source Networking

- Application Layer / App Server
- Network Data Analytics
- Orchestration, Management, Policy
- Cloud & Virtual Management
- Network Control
- Operating Systems
- IO Abstraction & Data Path
- Disaggregated Hardware

Automation of Network + Infrastructure + Cloud + Apps + IOT
ETSI NFV
Summary and Key Messages

• Automation and programmability are required these days

• Data Modeling-driven set of APIs is key for automation
  • The key is the data models

• YANG is the data modeling language for configuration and monitoring: a full, formal contract language with rich syntax and semantics to build applications on.

• Many YANG data model developments
  • In different standard development organizations (but primarily at the IETF),
  • In opensource

• The SDOs start to align

• Time to synchronize to reduce the OPEX
SDN and metrics from the SDOs: Report from Benoit Claise (IETF OPS Area Director)

Dave Ward
dward@cisco.com
2017.04.06