Data Modelling:
IEEE 802.1 Configuration and Control With NETCONF/YANG

Peter Jones

Cisco Systems

Version 1

May 11, 2014

Author Assumptions and Disclaimer

Assumptions

- **The goals of the Time-Sensitive Network Task Group (and other IEEE groups) should include**
  - **Define simply and clearly the primitives (aka building blocks) required to operate and manage implementations of our standards.**
  - **Simply support multiple ways to encode and transport this data based on system scale and environment.**

Disclaimer

- **Today the author knows a lot more about SNMP/SMI, than about NETCONF/YANG, but he knows where he needs to invest.**

Why is this important?

- IEEE 802.1 standards define a formal management interface using SNMP SMIv2 definitions (i.e. a MIB).
  - This is often a poorly understood part of the project, done towards the end and feels like guild knowledge, e.g. *All those who understand MIBs, please take one step forwards.*
- Many of our standards also define peer to peer communication protocols (e.g. SRP).
  - Because our standards are used all over the industry, these protocols can end up as an uncomfortable compromise between conflicting needs (e.g. scale, footprint, reach, simplicity, etc)
- This presentation is focused on TSN right now, but has broad applicability throughout IEEE 802.

What could we do different?

- Move away from specific syntax and transport, and focus on the semantics and operations.
- Said another way
  - spend more time on what we need to get done.
  - less time on the exact details of the encoding and transport of messages.
- How to tackle PICs and interoperability?
  - Briefly covered later in the deck, but this is not as hard as you might think.

TSN needs

What do we need to update/extend/replace?

- The MIB (formal programmatic interface) used by an NMS device (aka controller in today’s terms) to operate (e.g. configure/ monitor/ troubleshoot) TSN devices
  - (see FCAPS link for good background definitions)
- SRP for UNI /NNI signaling for stream control between TSN devices, including:
  - stream (or streams for in network protection) establishment
  - resource reservation
- PCEP (or PCEP like) protocol to/from PCE-like devices offering static or dynamic network planning services
  - This is an external function that maps service needs to network resources.
- All of these need to access and manipulate the same data store in the same network elements.

Past Practice?

- If we operated as normal, we would
  - ➢ Define/extend the MIB (SMI format) for configuration, operational state and statistics reporting.
  - ➢ Define/extend TLVs for service signaling in a set of protocols (e.g., SRP, IS-IS, PCEP, maybe RSVP-TE) depending on the network type and scope.
    - o It’s my belief that we would end up redefining the same basic data in a bunch of different formats.
  - ➢ Deal with MIB/TLV inconsistencies and mapping/translation between signaling protocols defined in various SDOs (e.g. IEEE, IETF, etc).

Best Practice?

- I propose that we investigate data modelling languages available today that meet the following requirements:
  - Allow high level description of data models.
  - Have (or are gaining) broad industry and standards organization support.
  - Allow for multiple data encoding schemes.
  - Allow for multiple transport protocol mappings.

Industry Movement

IETF moving from SNMP to NETCONF/YANG

- NETCONF/YANG replace SNMP/SMI.
- What's NETCONF? (wikipedia link)
  - NETCONF defines the operations, messaging and transport for managing network devices.
  - Defined in RFC 6241 (link)
  - RFC 6244 is an excellent primer (link)
- What's YANG? (wiki link)
  - YANG is a data modeling language used to model configuration and state data (includes operational state and statistics)
  - Defined in RFC 6020 (link)
  - Developed for NETCONF, but not limited to NETCONF
- Recent activity
  - All new IETF work that needs configuration is strongly encouraged to use NETCONF/YANG (IESG statement link)
  - The NETCONF Data Modeling Language (netmod) WG is actively defining YANG versions of the basic SNMP MIBs (e.g. system, interfaces, IP, routing, etc.).

YANG and SMI mapping: Some RFC6021 examples

YANG type	Equivalent SMIv2 type (module)
counter32	Counter32 (SNMPv2-SMI)
zero-based-counter32	ZeroBasedCounter32 (RMON2-MIB)
counter64	Counter64 (SNMPv2-SMI)
zero-based-counter64	ZeroBasedCounter64 (HCNUM-TC)
gauge32	Gauge32 (SNMPv2-SMI)
gauge64	CounterBasedGauge64 (HCNUM-TC)
object-identifier	-
object-identifier-128	OBJECT IDENTIFIER
date-and-time	-
timeticks	TimeTicks (SNMPv2-SMI)
timestamp	TimeStamp (SNMPv2-TC)
phys-address	PhysAddress (SNMPv2-TC)
mac-address	MacAddress (SNMPv2-TC)
xpath1.0	-

YANG type	Equivalent SMIv2 type (module)
ip-version	InetVersion (INET-ADDRESS-MIB)
dscp	Dscp (DIFFSERV-DSCP-TC)
ipv6-flow-label	IPv6FlowLabel (IPV6-FLOW-LABEL-MIB)
port-number	InetPortNumber (INET-ADDRESS-MIB)
as-number	InetAutonomousSystemNumber (INET-ADDRESS-MIB)

<snip>

uri	Uri (URI-TC-MIB)

IF-MIB SMI and YANG extracts

```
IfEntry ::=
   SEQUENCE {
      ifIndex            InterfaceIndex,
      …
      ifType             IANAifType,
      …
   }

IfXEntry ::=
   SEQUENCE {
      ifName             DisplayString,
      ….
   }

ifName
   OBJECT-TYPE
   SYNTAX       DisplayString
   MAX-ACCESS   read-only
   STATUS       current
   DESCRIPTION
      The textual name of the interface.“
   ::= { ifXEntry 1 }

ifType OBJECT-TYPE
   SYNTAX       IANAifType
   MAX-ACCESS   read-only
   STATUS       current
   DESCRIPTION
      "The type of interface.“
   ::= { ifEntry 3 }
```

```
container interfaces-state {
   config false;
   description
      "Data nodes for the operational state of
      interfaces.";

list interface {
   key "name";
   description
      "The list of interfaces on the device.  “

leaf name {
   type string;
   description
      "The name of the interface.”
   reference
      “RFC 2863: The Interfaces Group MIB - ifName";
   }

leaf type {
   type identityref {
     base interface-type;
   }
   mandatory true;
   description
      "The type of the interface.";
   reference
      “RFC 2863: The Interfaces Group MIB - IfType";
   }
```

OpenDaylight & Yang

- OpenDaylight is a Linux Foundation collaborative project for building SDN infrastructure.
- They gave a presentation to IETF 88 on
  - their overall goals
  - Model Driven Service Abstraction Layer
  - A set of asks for the netconf and netmod WGs.
- I believe that moving towards defining functionality based on models is one of their key elements.
- I believe that we can use this trend to our advantage and head down the same path

YANG/NETCONF options today.
(standard or draft)

- Transports (not including Historic)
  - SSL
  - TLS
  - RESTCONF
- Encodings
  - XML
  - JSON
  - EXI (Efficient XML Interchange)
- It seems like there will be more to come.

What’s EXI - Efficient XML Interchange?

- It’s a World Wide Web Consortium (W3C) standard designed to be a
  - Very compact representation for XML intended to simultaneously optimize encode/decode time and performance and message size (link)
- Main design goals (link)
  - General:
  - Minimal:
  - Efficient:
  - Flexible:
  - Interoperable:
- EXI as an encoding may be a good fit for smaller systems.
  - There are number of open source implementations available, including EXIficient, OpenEXI and EXIP.

Moving Forward?

Next Steps – High Level

- My proposal is that we:
  - Adopt a consistent way to define data models (objects + operations) for our standards.
  - Spend most effort deciding what needs to get done, not how to encode or transport it.
- If we agree on this, or decide that it's worth pursuing further then:
  - Find the set of people most interested in following up.
  - Start to work on the tasks proposed on the next slide.

Next Steps - Details

- Push for 802.1 as a whole to move to replace SNMP/SMI with NETCONF/YANG
  - There are mechanical translators (e.g., web libsmi) for converting SMI MIB to YANG model
  - Many common SNMP tools/companies (e.g., EMENATE, WebNMS, MG-SOFT) already have NETCONF/YANG support
- Pick at least one preferred transport and encoding, SSL/XML align to IETF.
  - Investigate other options as needed (e.g., L2 transport protocol, EXI)
- Develop YANG models for
  - Interface to NMS/controller
  - Interface to PCE
  - Peer to Peer signaling (I think this is the biggest stretch, but ATM ILMI using SNMP is one existence proof)

Thank you.