Extending YANG for events, actions, and finite state machine
draft-sambo-opsawg-ccamp-supad-ext-yang-fsm-00

N. Sambo¹, M. Dallaglio¹, P. Castoldi¹, G. Fioccola², A. Di Giglio², F. Cugini³, G. Bernini⁴, P. Giardina⁴

1: Scuola Superiore Sant’Anna, Pisa, Italy
2: Telecom Italia, Torino, Italy
3: CNIT, Pisa, Italy
4: Nextworks, Pisa, Italy

IETF Meeting 99, Prague, July 2017
Proposal

• YANG models for events and finite state machine to program actions in YANG-modelled network devices
• Use case:
  • Flexible transponders in elastic optical networks: multiple rates, multiple modulation formats, multiple FECs
  • Format and FEC can be set based on optical physical layer (e.g., PM-QPSK more robust than PM-16QAM)
  • If physical conditions change (e.g., soft failure: BER increase), format or FEC can be adapted to get more robust transmission
Control scenario

- ABNO controller: governs the behavior of the network in response to changing network conditions
- SDN controller: for configuration and reconfiguration of the data plane
- OAM Handler: for receiving and processing alarms and triggering reconfiguration for maintenance

Figure 1: Assumed ABNO functional modules
State of the art

Active service connection

Degradation

OAM H

Alarm

Transmission parameter computation

SDN controller

Configuration

Recovery

Time consuming
Use case of application for FSM YANG model

Active service connection

Faster

Degradation

Fast reaction based on instructions
This draft

- YANG models for events, actions, and state machine
- the actions to be taken and the critical events can be re-programmed on the device by simply sending a new message configuration (NETCONF message) on the device controller with the new information
YANG model for events

```
+--rw events
    +--rw event [name type]
        +--rw name string
        +--rw type event-type
        +--rw description? string
        +--rw filters
            | +--rw filter [filter-id]
            |     +--rw filter-id yp:filter-id
        +--rw reaction
            +--rw operation [id]
                +--rw id event-id-type
                +--rw type enumeration
                +--rw conditional
                    | +--rw statement string
                    | +--rw true
                    |   +--rw execute
                    |   +--rw next-operation? event-id-type
                    | +--rw false
                    |   +--rw execute
                    |   +--rw next-operation? event-id-type
                +--rw simple
                    +--rw execute
                    +--rw next-operation? event-id-type
```

e.g., BER increase
to further express the event: e.g., BER>TH
e.g., change modulation format
an <rpc> is called
YANG model for finite state machine

This model extends the one of the events including state information and transition

```yang
topology
  +--rw current-state?  leafref
  +--rw states
    +--rw state [id]
      +--rw id           state-id-type
      +--rw name         string
      +--rw description? string
    +--rw events
      +--rw event [name type]
        +--rw name string
        +--rw type event-type
        +--rw description? string
        +--rw filters
          | +--rw filter [filter-id]
          |   +--rw filter-id   yq:filter-id
        +--rw reaction
          +--rw operation [id]
            +--rw id           event-id-type
            +--rw type         enumeration
            +--rw conditional
              | +--rw statement string
              |   +--rw true
              |     | +--rw execute
              |     |   | +--rw next-operation? event-id-type
              |     |   |   +--rw next-state? leafref
              |     | +--rw false
              |     |   | +--rw execute
              |     |   |   +--rw next-operation? event-id-type
              |     |   |   | +--rw next-state? leafref
            +--rw simple
              +--rw execute
              +--rw next-operation? event-id-type
              +--rw next-state? leafref
```

Implementation for the use case of application

- Event: BER > BER_{th}
- Reaction: e.g., FEC adaptation

- Event: BER < BER_{th}
- Reaction: FEC adaptation
Control plane set up

Three PCs:
• SDN controller (python implementation of a NETCONF client)
• TX controller (ConfD)
• RX controller (ConfD)

Scenario:
• $\text{BER}_{th} = 9 \times 10^{-4}$
• 100-Gbps net rate
• PM-QPSK

**Steady state:**
• 28 Gbaud
• 7% of FEC

**FEC-adapt state:**
• 31 Gbaud
• 20% of FEC
FSM installation on the device controller

Event BER increase with “Filter” expressing a threshold on the BER

“Steady” state

Reaction consists in increasing redundancy

Event BER decrease

“FEC-adapt” state

Reaction consists in decreasing redundancy
Summary

• YANG model for finite state machine
• Generic
• Use case: reconfig of optical transponders upon physical layer degradations
• OTHER USE CASES?

• Comments:
  - ‘Extending’ can be misleading due to ‘extension’ syntax in YANG
ACK: The work has been partially supported by the ORCHESTRA project.

email: nicola.sambo@sssup.it