Basic Ethernet YANG Models November 2016

Rob Wilton, Mahesh Jethanandani & Peter Jones - Cisco

Contents

Overview

Key 802.3 YANG models

Significant parts of the industry are pushing for automation via YANG.

The following 802.3 technology related YANG models seem to be most important to tackle first:

1. Basic Ethernet interface configuration
2. Operational state for effective monitoring of Ethernet interfaces
3. Link OAM – 802.3ah, Ethernet in the first mile.

Rob Wilton has offered to help authoring YANG models for (1) and (2).

There is already a proposed comprehensive model for (3) from another individual from Cisco. This is based on the standard Link OAM MIB and is ready for Task Force consideration.

Basic Ethernet Interface Configuration

Basic Ethernet Interface configuration

- Rob Wilton put a basic Ethernet configuration model forward about a year ago:
  - Github [YangModels/yang/experimental/ieee/802.3/ethernet.yang@3fe7a8c]
    - https://github.com/YangModels/yang/commit/3fe7a8c0
  - This model was structurally consistent, but incomplete.
  - It has been updated to pull in some MIB definitions but has lost some of its structural consistency in the process, and become less usable as a result:
    - https://github.com/YangModels/yang/commit/fbdd1977
- There have been further discussions in IETF on the best way to model settings for negotiated protocols:
  - Ethernet auto-negotiation is a particular case of this problem.
  - We should make sure that we align with IETF as much as possible to future proof the model.
- Hence, we propose to update the current poposed Ethernet configuration YANG model, and put that forward for review.

Basic Ethernet Configuration Coverage

The proposal is for the Ethernet model to definitely cover the following Ethernet related configuration properties:

- Speed
- Duplex
- Flow-control
- Auto-negotiation
- Priority flow control
- *Link quality monitoring (for further study)*

Interface config not covered in 802.3 model:

The proposal is that the following interface related properties are not covered in the 802.3 model because they are already addresses in the generic interfaces model:

- MTU – covered via draft-ietf-netmod-intf-ext-yang
- Loopback – covered via draft-ietf-netmod-intf-ext-yang
- MAC address - covered via draft-ietf-netmod-intf-ext-yang (because it applies to other Ethernet like interfaces)
- Inter-packet-gap – shouldn't need to be configurable
- Carrier-delay (also sometimes called holdtime) - covered via draft-ietf-netmod-intf-ext-yang

Optional Ethernet related configuration:

The following Ethernet interface related properties could be covered in an 802.3 model, or possibly in a more generic optics model? If it's in a more generic model, need to figure out where that would live.

- FEC
- Optical power monitoring

Feedback welcome

Ethernet Operational Data

What Cisco does today

Monitoring Ethernet interfaces

- Being able to monitor the behaviour of physical interfaces is critical to network operators.
- Cisco already has a XML based operational schema for Ethernet interfaces widely deployed.
- We propose to start from this deployed solution when building the 802.3 model:
  - Also pull in existing operational data from Etherlike MIB, and statistics from RMON & Etherlike MIB
  - Defer support for CSMA/CD related
  - Statistics should be split between general interface, PHY Ethernet, and Ethernet framing related Possibly some of the information may be related more generically to optics, and could potentially be defined in a separate optics YANG model?
  - We could still try and define it now and then figure out whether it should go in a generic optics module.
- The slides that follow use Cisco CLI show commands rather than XML for convenience of the reader.

Ethernet interface statistics

- IETF interface YANG model already defines some interface statistics (that all Ethernet interfaces will inherit)
  - There is no need to duplicate existing counters that are already available.
- Most additional Ethernet statistics are related to the physical layer (e.g. framing) and are tied to physical Ethernet interfaces.
  - All of these counters would be included.
  - E.g. the RMON etherStatsTable group, PAUSE frame counts, Symbol errors, unknown Opcodes, PFC statistics.
- Some counters relate to higher layers and would be covered by other modules in different projects (e.g. IETF flexible encaps draft):
  - E.g. packets dropped due to invalid VLAN, or invalid encapsulation, or invalid Dest MAC.
- Defer support for half-duplex collision statistics
- Include support for PFC and LPI

Cisco's Digital Optical Monitoring (DOM)

- DOM is a generic optics monitoring feature
- Operators need to monitor the quality of the optical signal
- SONET always had these features, and they have been gradually added to Ethernet interfaces.
- Different ways of monitoring: Ethernet alarms (LOS, RF, etc), BER monitoring, via protocols, DOM
- DOM monitors RX and TX signal power, and has warning and alarm threshold levels (exported by the PHY).
  - Some information is at the PHY level, other information is per optical lane.
- Alarms generate can warn a user an operator that a PHY is failing or likely to fail.
- We would like to include DOM support in the Ethernet operational data model, but it could go in a generic optics YANG model instead (but where to standardize?)
  - Maybe in IETF based on SFF-8472 - Specification for Diagnostic Monitoring Interface for Optical Transceivers

Cisco IOS-XR Ethernet Oper Model

The operational Ethernet model used by Cisco IOS-XR broadly covers:

- Link state (admin state, oper state, led, reason down)
- Current speed, duplex, flow-control settings, auto-negation settings
- L2 MTU, MRU
- Ethernet L2 related statistics
- Optics related information:
  - Media type, (e.g. 100GBASE-LR4)
  - Vendor, part number, serial number
- DOM information:
  - Transceiver temp, voltage, current alarms, previous alarms
  - Per lane information (wavelength, TX power, RX power, Laser Bias), alarms
  - Thresholds for alarms
- PHY statistics:
  - Sync header count, PCS BIP errors
  - FEC (corrected control word count, uncorrected control word count)
- PFC

Monitoring Ethernet interfaces – PHY (except DOM - to follow)

```
Phy:
    Media type: <Media Type>,
    [No optics present|Optics: <Optics>]
        Vendor: <Optics Vendor>
        Part number: <Optics vendor part number>
        Serial number: <Optics vendor serial number>

 Alarms:
        Current:
            [No alarms|<list of alarms>
        Previous:
            [No alarms|<list of alarms>
    Statistics:
        Sync Header Error Count: <count>
        PCS BIP Error count: <count>
        FEC:
            Corrected Codeword Count: <count>
            Uncorrected Codeword Count: <count>
```

Monitoring Ethernet interfaces – PHY DOM

```
Digital Optical Monitoring:
        Transceiver Temp: <temp> C
        Transceiver Voltage: <voltage> V

        Alarms key: (H) Alarm high, (h) Warning high
                    (L) Alarm low, (l) warning low
           Wavelength    Tx Power          Rx Power      Laser Bias
        Lane  (nm)    (dBm)    (mW)     (dBm)     (mW)      (mA)
        --   -----   ------   ------    ------   ------    ------
        <values per lane , including alarms>

        DOM alarms:
            [No alarms|<list of alarms>|Not available]


        Alarm                     Alarm    Warning   Warning    Alarm
        Thresholds                High      High       Low       Low
                                 -------   -------   -------   -------
        Transceiver Temp (C):     <al-h>    <wa-h>    <wa-l>    <al-l>
        Transceiver Voltage (V):  <al-h>    <wa-h>    <wa-l>    <al-l>
        Laser Bias (mA):          <al-h>    <wa-h>    <wa-l>    <al-l>
        Transmit Power (mW):      <al-h>    <wa-h>    <wa-l>    <al-l>
        Receive Power (mW):       <al-h>    <wa-h>    <wa-l>    <al-l>
```

Monitoring Ethernet interfaces – PFC

```
Priority Flow Control:
    Total Rx PFC Frames: <Total Rx Frames>
    Total Tx PFC Frames: <Total Tx Frames>
    CoS  Status   Rx Frames   Tx Frames
    ---  -------  ----------  ----------
      0  <off|on> <Rx Count>  <Tx Count>
      1  <off|on> <Rx Count>  <Tx Count>
      2  <off|on> <Rx Count>  <Tx Count>
      3  <off|on> <Rx Count>  <Tx Count>
      4  <off|on> <Rx Count>  <Tx Count>
      5  <off|on> <Rx Count>  <Tx Count>
      6  <off|on> <Rx Count>  <Tx Count>
      7  <off|on> <Rx Count>  <Tx Count>
```

IEEE and OpenConfig Models

Comparison

IEEE

```
module: ethernet
augment /if:interfaces/if:interface:
 +--rw ethernet
  +--rw (transmission-mode)?
   +--:(auto-negotiation)
   | +--rw auto-negotiation
   |  +--rw status?                          enumeration
   |  +--rw duplex?                          enumeration
   |  +--rw speed?                           identityref
   |  +--rw advertised-flow-control?         enumeration
   |  +--rw forced-flow-control?             flow-control-settings
   +--:(manual)
    +--rw manual
     +--rw duplex?                          enumeration
     +--rw speed?                           identityref
     +--rw flow-control?                    flow-control-settings

augment /if:interfaces-state/if:interface:
 +--ro ethernet
  +--ro auto-negotiation!
  | +--ro status?                           enumeration
  +--ro duplex?                             enumeration
  +--ro speed?                              uint64
  +--ro flow-control?                       flow-control-settings
```

OpenConfig

```
module: openconfig-if-ethernet
augment /oc-if:interfaces/oc-if:interface:
 +--rw ethernet
  +--rw config
  | +--rw mac-address?                      yang:mac-address
  | +--rw auto-negoLate?                    boolean
  | +--rw duplex-mode?                      enumeration
  | +--rw port-speed?                       identityref
  | +--rw enable-flow-control?              boolean
  +--ro state
   +--ro mac-address?                       yang:mac-address
   +--ro auto-negoLate?                     boolean
   +--ro duplex-mode?                       enumeration
   +--ro port-speed?                        identityref
   +--ro enable-flow-control?               boolean
   +--ro hw-mac-address?                    yang:mac-address
   +--ro effective-speed?                   uint32
   +--ro counters
    +--ro in-mac-control-frames?            yang:counter64
    +--ro in-mac-pause-frames?              yang:counter64
    +--ro in-oversize-frames?               yang:counter64
    +--ro in-jabber-frames?                 yang:counter64
    +--ro in-fragment-frames?               yang:counter64
    +--ro in-8021q-frames?                  yang:counter64
    +--ro in-crc-errors?                    yang:counter64
    +--ro out-mac-control-frames?           yang:counter64
    +--ro out-mac-pause-frames?             yang:counter64
    +--ro out-8021q-frames?                 yang:counter64
```

- OpenConfig attempts to address common cases, not full specification.
- There is some cross-pollination between people working on IETF, OpenConfig and IEEE YANG models.

Capability Reporting

Capability Reporting

- Ethernet configuration depends on the type of interface and PHY capabilities.
- YANG supports two ways of constraining a model:
  - "when" statements (hides part of the model if constraint isn't met)
  - "must" statements (constraints check before configuration can be applied)
  - For configuration, neither of these statements can depend on operational state.
  - So, neither statement help ensure that configuration is consistent with the hardware capabilities.
- However, it is still useful for a operator to be able to determine what configuration is allowed for particular Ethernet hardware
- IETF may come up with a formal solution for this, but in the meantime propose that we expose capabilities using a separate tree of config false leaves:
  - Indicates which configuration items are supported
  - Indicates which configuration item values will be accepted.

Generating Documentation

Model Development Process

Generating Documentation

- We propose to structure and write the models in YANG, without requiring the generation of UML models first.
  - The YANG model is the formal, normative specification
- YANG can be quite readable, but the overall structure isn't always readily apparent.
- Being able to see and review the overall tree structure and relationships is key to being able to review the models effectively.
- IETF makes use of textual tree output in RFCs that contain YANG models
- 802.3 documents have more flexibility, so a graphic solution may be better.
- We will investigate output formats the existing YANG tools can generate and choose the best of those (which might be UML, or something else) to represent the structure of the YANG models.
  - There are a number of tools that may be appropriate e.g., pyang, YDK-Py, Doxygen, etc.

Summary & Next Steps

Summary & Next Steps

- We propose the TF should start by writing models covering basic Ethernet configuration and also operational state
  - Do we have support for this approach?
- If so, then Rob Wilton volunteers to produce draft YANG models and send them out to the TF for review
  - Along with the tree output, and other generated documentation (e.g., box diagrams) to show the hierarchy to make it easier to review
- We propose to follow:
  - YANG author guidelines: draft-ietf-netmod-rfc6087bis-09
  - Base IETF interfaces model: RFC 7223
- Our goal is a self documenting model:
  - Generate documentation from the YANG source, including extra hints as needed.
  - Look at Doxogen: https://en.wikipedia.org/wiki/Doxygen as one example of a tool of this type.

Thank You!

Backup

Thank You!