

# Basic Ethernet YANG Models

Rob Wilton (Cisco)

[rwilton@cisco.com](mailto:rwilton@cisco.com)

28 Oct 2016

# Contents

Overview

Basic Ethernet Interface Configuration

Ethernet Operational Data

Capability Reporting

Generating Documentation

Summary & Next Steps

# Overview

# Key 802.3 YANG models

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.

I can help authoring YANG models for (1) and (2).

Another Cisco engineer has already put forward a comprehensive model for 3.

# Basic Ethernet Interface Configuration

# Basic Ethernet Interface configuration

- I 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, I propose that I update the current 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

# 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?:

- FEC
- Optical power monitoring

Feedback welcome

# Ethernet Operational Data

# 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
- We would like to use the data contained in that schema as a starting point for the 802.3 model:
  - Also pull in existing operational data from Etherlike MIB, and statistics from RMON MIB and Etherlike MIB
  - Defer support for CSMA/CD related statistics (could later be supported by a separate container, and possibly [sub] module).
  - Statistics should be split between general interface, PHY Ethernet, and Ethernet framing related (further details in next slide)
- 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 representations that follow use show commands rather than XML for convenience.

# 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

# 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 gradually have been to Ethernet interface
- 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?)

# Cisco XR Ethernet Oper Model

The operational Ethernet model used by Cisco XR broadly covers:

- Link state (admin state, oper state, led, reason down)
- Current speed, duplex, flow-control settings, auto-negotiation 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 Thresholds	Alarm High	Warning High	Warning Low	Alarm Low
Transceiver Temp (C):	<a1-h>	<wa-h>	<wa-l>	<a1-l>
Transceiver voltage (V):	<a1-h>	<wa-h>	<wa-l>	<a1-l>
Laser Bias (mA):	<a1-h>	<wa-h>	<wa-l>	<a1-l>
Transmit Power (mW):	<a1-h>	<wa-h>	<wa-l>	<a1-l>
Receive Power (mW):	<a1-h>	<wa-h>	<wa-l>	<a1-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>

# 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

# Generating Documentation

- The plan is to structure and write the models in YANG without requiring the overhead of generating UML models first.
- 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.
- The plan is to see what 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.
- Doxygen may be another alternative.

# Summary & Next Steps

# Summary & Next Steps

- I have proposed writing models covering basic Ethernet configuration and also operational state
- Request feedback on whether the general approach is acceptable
- If the approach is OK, then I'll produce the YANG models and send them out to the TF for review (along with the tree output, and perhaps some box diagrams to show the container hierarchy to make it easier to review)
- YANG author guidelines: [draft-ietf-netmod-rfc6087bis-09](#)
- Base IETF interfaces model: [RFC 7223](#)
- Producing a self documenting model:
  - Can we generate from the YANG itself?
  - Explore using Doxygen: <https://en.wikipedia.org/wiki/Doxygen>, or similar