802.3 YANG
Base Interface Module

Rob Wilton

Cisco

2017 Jan 9 Pleminary - Remote

802.3 YANG Base Interface Module

Agenda:

- Module Overview
- Current Module Status
- Module Structure
- Key points of interest

Module Overview

- I'm writing a base Ethernet Interface model:
  - Aims to be sufficient to configure and manage the Ethernet settings for a "normal" Ethernet interface (e.g. as would be seen on a router, bridge, or host)
  - Also covers some deprecated stuff (CSMA-CD & dynamic rate control)
  - Other Ethernet models (e.g. for PON, POE, Ethernet protocols) are expected to augment the structure contained in this model
- Being developed in the github ***8023YangDesignTeam/EthernetYang*** project **on the *eth-intf* branch:**

  https://github.com/8023YangDesignTeam/EthernetYang/tree/eth-intf/experimental/ieee/802.3

Module Status

Covering:

ieee802-Ethernet-interface.yang, and

ieee802-Ethernet-interface-deprecated.yang

Not complete yet, but getting closer:

- Basic structure should be reviewable
  - E.g. where I have put configuration and the structure of the containers.
- Further refinement of “description” statements is required (marked in the model as XXX or TODO)
- Speed, duplex, basic flow-control, auto-neg config/state should be reviewable

Module Status - updates

Updates from previous version (that are on the **master** branch):

- Moved deprecated features (CSMA-CD and into a separate YANG file (ieee802-Ethernet-interface-deprecated.yang)
- Reformatted files to make them easier to work with and review.
- Changed speed to decimal64 in Gb/s to 3 dp
- Moved Ethernet stats under the Ethernet interface container
  - Clients retrieving the higher layer interface stats may not want the raw Ethernet counters)
- Rejigged auto-neg, duplex, speed, flow-control to work better with an operational state datastore
- Moved pause and priority flow-control together under a common container
- Added a capabilities container, and started adding elements.

Module Status – to do

Planned updates:

- Merge in the appropriate RMON counters into the statistics container *(see separate presentation)*
  - Should this include counters like etherStatsPkts65to127Octets?
- Move flow control counters under flow control operational state container (rather than being under *ethernet/statistics*).
  - Gives a clean place to put per class PFC counters.
- Move LPI counters? Where should these live?
- Model FEC (config + counters)
- Complete missing capabilities

Module Structure ieee802-ethernet-interface

Config

```
augment /if:interfaces/if:interface:
  +--rw ethernet
     +--rw auto-negotiation
     |  +--rw enable?           boolean
     +--rw duplex?              duplex-type
     +--rw speed?               eth-if-speed-type
     +--rw flow-control
        +--rw pause
        |  +--rw direction?     pause-fc-direction-type
        +--rw priority
        |  +--rw enable?        boolean
        +--rw force-flow-control?   empty
```

State

```
augment /if:interfaces-state/if:interface:
   +--ro ethernet
      +--ro auto-negotiation
      |  +--ro enable?                  boolean
      |  +--ro negotiation-status?      enumeration
      +--ro duplex?                     duplex-type
      +--ro speed?                      eth-if-speed-type
      +--ro flow-control
      |  +--ro pause
      |  |  +--ro direction?        pause-fc-direction-type
      |  +--ro priority
      |  |  +--ro enable?                boolean
      |  +--ro force-flow-control?       empty
      +--ro macc-extension-control?      boolean
      +--ro frame-limit-slow-protocol?   uint32
      +--ro capabilities
      |  +--ro auto-negotiation?         boolean
      +--ro statistics
         +--ro in-pkts-errors-fcs?          yang:counter64
         +--ro in-pkts-errors-alignment?    yang:counter64
         +--ro in-giant-pkts?               yang:counter64
         +--ro in-giant-threshold-pkts?     yang:counter64
         +--ro in-errors-symbol?            yang:counter64
         +--ro out-errors-MAC-internal?     yang:counter64
```

Module Structure
ieee802-ethernet-interface-deprecated

Config

```
augment /if:interfaces/if:interface/eth-if:ethernet:
    x--rw dynamic-rate-control? …
     … dynamic-rate-control-type {dynamic-rate-control}?
```

State

```
augment /if:interfaces-state/if:interface/…
                      … eth-if:ethernet/eth-if:capabilities:
  +--ro dynamic-rate-control-supported? …
                           … boolean {dynamic-rate-control}?
augment /if:interfaces-state/if:interface/eth-if:ethernet:
augment /if:interfaces-state/if:interface/eth-if:ethernet/…
                                          … eth-if:statistics:
  x--ro csma-cd {csma-cd}?
     x--ro in-errors-sqe-test?              yang:counter64
     x--ro in-errors-MAC-internal?          yang:counter64
     x--ro out-pkts-collision-single?       yang:counter64
     x--ro out-pkts-collision-multiple?     yang:counter64
     x--ro out-pkts-deferred?               yang:counter64
     x--ro out-pkts-collisions-excessive?   yang:counter64
     x--ro out-collisions-late?             yang:counter64
     x--ro out-errors-carrier-sense?        yang:counter64
     x--ro collision-histogram* [collision-count]
        x--ro collision-count         yang:counter64
        x--ro collision-count-pkts?   yang:counter64
```

Questions on Module Structure

- Does it look OK?
- Other Ethernet features/protocols would be expected to augment:

  `/if:interfaces/if:interface/eth-if:ethernet`
  `/if:interfaces-state/if:interface/eth-if:Ethernet`

  - So, by default a GET of `/if:interfaces/if:interface/eth-if:Ethernet` would return **both** interface settings and protocol settings as one structure
  - Should the Ethernet interface level configuration parameter and state go in a separate container?
- Is the split between current/deprecated OK?
- Comments on statistics perhaps best deferred to the separate presentation.

Thank you!