802.3 YANG
Base Interface Statistics Update

Rob Wilton

Cisco

2017 Jan 30 TF Ad Hoc

Overview

- Recap of statistics conclusion from Interim
- Explanation of statistics issues
- Overview of current proposed statistics structure
  - Latest committed to eth-intf branch on github

Conclusion from Interim

- Leave Clause 30 counters as they are.
- All counters in YANG model can be 64bit.
- Histogram counters should be generic and go to Ethernet-like module in IETF. Please refer to page 12 in “wilton_3cf_03_0117.pdf”.
- Don’t redefine attributes but refer to definitions in Clause 30.
- *But want as much granularity as possible (note – this was not formally stated in minutes)*

Statistics Issues (1): FCS & Alignment

- RMON counters combine FCS and Alignment errors into one error counter.
  - I would like to do the same for the YANG
- Clause 30 and MIB separates them into two counters:
  - But 30.3.1.1.7 aAlignmentErrors": *This counter will not increment for group encoding schemes encoding greater than 4 bits per group.*
  - *Implies that it doesn't increment for 1Gb/s (8b/10b encoding) or faster?*
  - *If so, seems pointless to have a separate counter, it will nearly always be 0. (We could possibly put it in legacy module, …)*

=> Combining them into one counter seems sensible.

Statistics Issues (2): Giants/Runts with/without FCS

- RMON counts splits oversized and undersized pkts into separate counters depending on whether they pass FCS:

```
etherStatsUndersizePkts          Counter32,  // pkt < 64, good CRC
etherStatsOversizePkts           Counter32,  // pkt > 1518, good CRC
etherStatsFragments              Counter32,  // pkt < 64, bad CRC
etherStatsJabbers                Counter32,  // pkt > 1518, bad CRC
```

- Clause 30 (and Etherlike MIB) don't support this split, only provide combined counters regardless of FCS check.
- I propose that we follow the existing Clause 30, and don't introduce new split FCS good/bad clause 30 registers for giants/runts. Proposed:

```
in-errors-giant-pkts             Counter64,  // pkt > max-frame-length
in-errors-runt-pkts              Counter64,  // pkt < 64
```

802.3 Ethernet Frame/Phy Counters

(Combined Etherlike MIB and RMON MIB)

```
This counters are in addition to the ietf-interfaces statistics.
interfaces-state/interface/ethernet/frame-statistics:
  in-total-octets           counter64, // Total received bytes (good + bad)
  in-total-pkts             counter64, // Total received pkts (good + bad)
  in-pkts-errors-fcs        counter64, // 64 <= pkt <= 1518, bad CRC or alignment
  in-pkts-errors-runt       counter64, // pkt < 64
  in-pkts-errors-giant      counter64, // pkt > MRU
  out-total-octets          counter64, // Total transmitted bytes (good + bad)
  out-total=pkts            counter64, // Total transmitted pkts (good + bad)
  // May still be some generic input/output errors missing.

interfaces-state/interface/ethernet/phy-statistics:
  in-errors-symbol            counter64, // symbol errors
  lpi {   <- TODO, make LPI a feature.
    in-lpi-transitions        counter64, // lpi transitions
    in-lpi-time               decimal64, // lpi time (seconds, 6 d.p.)
    out-lpi-transitions       counter64, // lpi transitions
    out-lpi-time              decimal64, // lpi time (seconds, 6 d.p.)
  }
```

802.3 Ethernet Flow Ctrl Counters + Other MAC Control counters

```
augment /if:interfaces-state/if:interface:
   +--ro ethernet
      <snipped>
      +--ro flow-control
      |  +--ro pause
      |  |  +--ro direction?    pause-fc-direction-type
      |  |  +--ro statistics
      |  |     +--ro in-pkts-pause?    yang:counter64
      |  |     +--ro out-pkts-pause?   yang:counter64
      |  +--ro pfc  <- TODO Make PFC a feature
      |  |  +--ro enable?       boolean
      |  |  +--ro statistics
      |  |     +--ro in-pkts-pfc?    yang:counter64   <- Can we add per class PFC counters here?
      |  |     +--ro out-pkts-pfc?   yang:counter64
      |  +--ro force-flow-control?   Empty
      +--ro mac-control-statistics                    <- Generic counters.
         +--ro in-pkts-macc-unknown?      yang:counter64
         +--ro in-pkts-macc-extension?    yang:counter64
         +--ro out-pkts-macc-extension?   yang:counter64
```

Any Questions?
Thank you!

Backup Slides

IETF interface YANG statistics

(For reference. Every Ethernet interface always has these)

```
+--ro statistics
   +--ro discontinuity-time    yang:date-and-time

   +--ro in-octets?            yang:counter64 = (total good bytes, inc fcs chars)
   +--ro in-unicast-pkts?      yang:counter64 = good uni pkts    (not drop/error/
   +--ro in-broadcast-pkts?    yang:counter64 = good bcast pkts           unknown)
   +--ro in-multicast-pkts?    yang:counter64 = good mcast pkts      "
   +--ro in-discards?          yang:counter32 = e.g. QoS/ACL drops
   +--ro in-errors?            yang:counter32 = e.g. Frame errors
   +--ro in-unknown-protos?    yang:counter32 = e.g. Unknown proto drops.

   +--ro out-octets?           yang:counter64
   +--ro out-unicast-pkts?     yang:counter64
   +--ro out-broadcast-pkts?   yang:counter64
   +--ro out-multicast-pkts?   yang:counter64
   +--ro out-discards?         yang:counter32
   +--ro out-errors?           yang:counter32
```

Existing RMON MIB Ethernet counters

(For reference purposes only, defined in RFC 2819)

```
etherStatsDropEvents              Counter32,  // Drop due to lack of resources
etherStatsOctets                  Counter32,  // Total bytes (good + bad)
etherStatsPkts                    Counter32,  // Total pkts (good + bad)
etherStatsBroadcastPkts           Counter32,  // Total good bcast pkts
etherStatsMulticastPkts           Counter32,  // Total good mcast pkts
etherStatsCRCAlignErrors          Counter32,  // 64 <= pkt <= 1518, bad CRC/align
etherStatsUndersizePkts           Counter32,  // pkt < 64, good CRC
etherStatsOversizePkts            Counter32,  // pkt > 1518, good CRC
etherStatsFragments               Counter32,  // pkt < 64, bad CRC
etherStatsJabbers                 Counter32,  // pkt > 1518, bad CRC
etherStatsCollisions              Counter32,  // Collision estimate
etherStatsPkts64Octets            Counter32,  // 64 byte pkts
etherStatsPkts65to127Octets       Counter32,  // 65 – 127 byte pkts
etherStatsPkts128to255Octets      Counter32,  // 128 – 255 byte pkts
etherStatsPkts256to511Octets      Counter32,  // 256 – 511 byte pkts
etherStatsPkts512to1023Octets     Counter32,  // 512 – 1023 byte pkts
etherStatsPkts1024to1518Octets    Counter32,  // 1024 – 1518 byte pkts
```