Scalable Link Peer Validation for “baseT”

Michael Johas Teener
mikejt@broadcom.com

Agenda

- Need for link peer validation
  - and how it's addressed in 802.1
- Problems in the real world
  - and how THOSE are addressed in 802.1 AVB
  - and why it's not good enough
- (Sidenote: how long does autonegotiation take?)
- Proposal: "The last bit defined for autonegotiation"
  - do the *minimum* necessary to bring up a link
  - run LLDP for all future status exchanges
- Conclusion

Why link peer validation?

- For AVB:
  - all time-sensitive protocols are dependent on knowing the characteristics of a LAN
  - for 802.3 LANs, AVB will only accept a full duplex connection, since CSMA-CD introduces non-deterministic delays ... autonegotiation tells us that
  - for 802.1 bridges, AVB requires peers to understand the protocols, since non-AVB bridges also introduce non-deterministic delays ... LLDP/802.1AS/802.1Qat do peer discovery
- For DCB:
  - for 802.1 bridges, DCB requires peers to understand the protocols, since non-DCB bridges have unmanageable congestion

Problems in the real world

- Current 802.1 assumptions are that frames with the 802.1-reserved addresses are NOT repeated through bridges
  - in particular those with the 01-80-C2-00-00-0E LLDP address
- There are these things called “switches” or “unmanaged bridges” or “buffered repeaters”
  - which are, unfortunately, roughly 100% of the home market (maybe more)
  - these devices almost always repeat the “0E” address
- They make the use of LLDP almost impossible in a home environment

AVB solutions

- One of the AVB projects is 802.1AS time synch
- 802.1AS is required for all the “802.1BA AVB Systems” profiles
- 802.1AS provides (as a byproduct of the path delay - “PDelay” - process) the delay between two adjacent 802.1AS entities
  - the assumption being that 802.1AS/802.3 entities are directly connected
- if the delay is greater than that expected for 100m of UTP, then the 802.1BA home network profile will specify that the link is “non-participating”
  - e.g., there is a buffer of some sort in the way

… but that's not good enough

- Some 802.1BA profiles want to support longer distance links
  - professional A/V, industrial both may use optical for >100m links
- Some 802.1BA profiles will require 10G and up
  - professional A/V has 1.4Gbit per stream (and up)
  - 10G buffered repeater could fit underneath the time delay for 100m UTP
- … and then there's DCB
  - where links can be anything from a few inches to several kilometers
  - and there is no requirement for 802.1AS

Sidenote:
how long does autonegotiation take?

- Somebody tell me how long does a typical autoneg sequence take for:
  - 100BASE-T?
  - 1G BASE-T?
  - 10G BASE-T?
- Remember these numbers!

Proposal: the last autonegotiation

- Use just one more code somewhere in the autonegotiation sequence to indicate that:
  - use existing information to bring up the link
  - there is an LLDP-capable entity attached to the PHY that can take over the information exchange
  - LLDP will handle all further information / state exchange that will feed into all future link bring-up protocols
- 802.3 will never again define another physical layer startup protocol that does anything other than establish a baseline communication capability that supports LLDP

… and maybe …

- At some future point, 802.3 could deprecate most of autonegotiation …
  - how long would it take to bring up 10G links if there was an intermediate 100BASE-T stage where LLDP did the information exchange?
  - maybe 40G links?
  - maybe 100G links?
  - maybe link aggregation?
    - oops, that's a 802.1 thing now, but still …

Conclusion

- LLDP is a very useful lightweight protocol useful for a vast number of applications:
  - AVB, DCB, (EEE?, PoE?)
- we need to guarantee that it really is a LINK discovery protocol
- autonegotiation can do that
  - and then we can stop using autonegotiation for anything other than bare-bones link bring-up