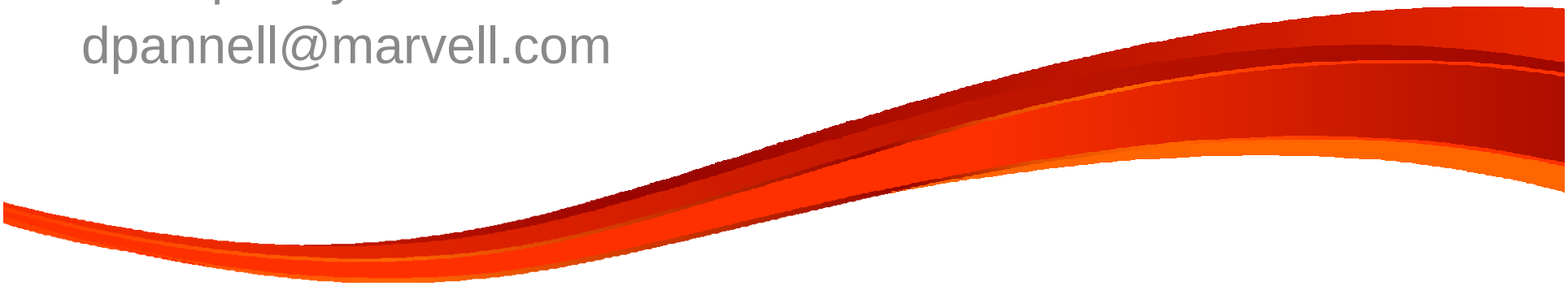
Interoperable Pdelays in IEEE 802.1AS-Rev

Don Pannell
Principal Systems Architect
dpannell@marvell.com



IEEE 802.1 Face to Face
March 2016 - Macau

Problem & Goals

- If IEEE 802.1AS-Rev adopts the proposed Common Mean Path Delay Service Option (http://www.ieee802.org/1/files/public/docs2016/as-cummings-1588-common-p2p-0316-v1.pdf), how can this co-exist with IEEE 802.1AS-2011's Pdelay?
- The goals here is to make sure "new" systems can interoperate with an IEEE 802.1AS-2011 neighbor at that standard's level of features
- And to detect neighbors that are also running the "new" version and IEEE 802.1AS-Rev
- And to do this with minimal delays and minimal network overhead

Base Proposal for IEEE 802.1AS-Rev

- At the prescribed Pdelay interval, new device's transmit 2 back-to-back (or as soon as it can) Pdelay_Req frames. The 1st frame is the "new" Common Mean Path Delay Service Option frame and the 2nd frame is the "older" 802.1AS-2011 pDelayReq frame.
- Responders will respond to the 1st Pdelay_Req frame that it understands & once a port responds to a "newer" frame type it will ignore (i.e., not respond to) "older" frame types.
  - Error conditions & corner cases are covered later in this presentation
- The 1st bullet adds a small amount of network overhead, while the 2nd bullet maintains the same network overhead.
- This base proposal can be further optimized to reduce network overhead.

AS-Rev device to non-AS device

- Both Pdelays per interval are transmitted
- No responses are ever received

- Other requirements:
- Transmit Pdelay_Req's for N Pdelay cycles per link up then stop transmitting Pdelay_Req's
- What about devices that don't launch support for gPTP until they are running an application that needed gPTP?
  - This could be solved if they start sending Pdelay_Req's when they need gPTP services and any station that is currently not sending Pdelay_Req's shall start doing so as soon as a Pdelay_Req is received (the same type is used too).

AS-Rev device to AS-2011 device

- Both Pdelays per interval are transmitted
- “Older” Pdelay_Req’s will be responded to & transmitter knows its link partner is AS-2011 capable

- Other requirements:
- Stop transmitting “new” Pdelay_Req’s after N Pdelay cycles per link up

- Can’t stop transmitting “new” Pdelay_Req’s right after receiving an “older” Pdelay_Resp due to possible errors (this is covered later)

AS-Rev device to AS-Rev device

- Both Pdelays per interval are transmitted
- “New” Pdelay_Req’s will be responded to & the “older” Pdelay_Req’s will be ignored - transmitter knows its link partner is AS-Rev capable

- Other requirements:
- Stop transmitting “older” Pdelay_Req’s after receiving first “new” Pdelay_Resp

- Error condition handling is next…

AS-Rev device to AS-Rev device w/Errors

- Both Pdelays per interval are transmitted
- “Older” Pdelay_Req’s was be responded to because of a CRC (or other) error on the “new” Pdelay_Req
- Requester assumes responder is AS-2011 only capable but continues for N more cycles to transmit “new” Pdelay_Req’s in case this is an error
- The “older” Pdelay cable measurement data can still be used even if the protocol switches to the “new” format on the next cycle

- Once a requestor sees its responder is AS-Rev capable it can stop transmitting the “older” Pdelay_Req

Summary

- This approach starts up just as fast a IEEE 802.1AS-2011 does
- Very little extra overhead is needed and with optimizations this is a start-up condition only

- Possible Issues:
- Will IEEE 802.1AS-2011 devices properly ignore the “new” Common Mean Path Delay Service Option frames? This is not a standards issue, but an implementation issue – but this could still impact our decisions

Questions?

This presentation is intended as a discussion point & possibly a place to start. More thought and verification by this group on this topic is needed.