Background for ER1 PTP Accuracy Issue (Comment #108)

David Ofelt – Juniper Networks

2024-04 802.3dj Interim

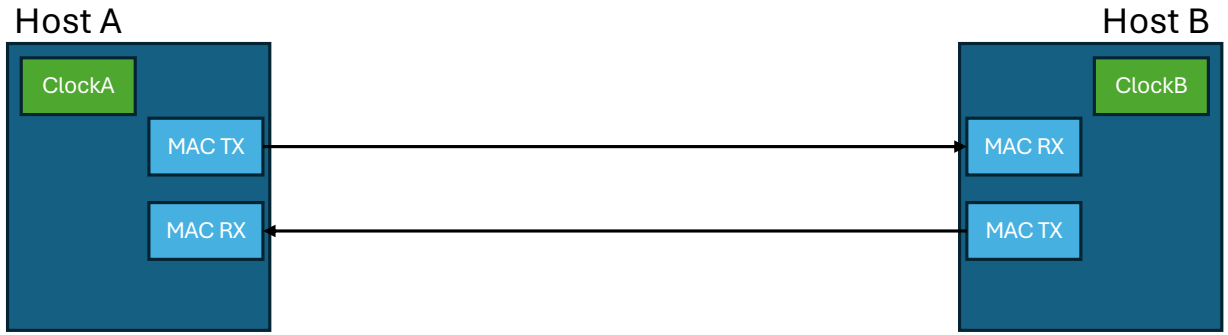
Contributers

- Gary Nicholl – Cisco
- John D’Ambrosia – Futurewei
- David Law – HPE
- Ulf Parkholm - Ericsson

Introduction

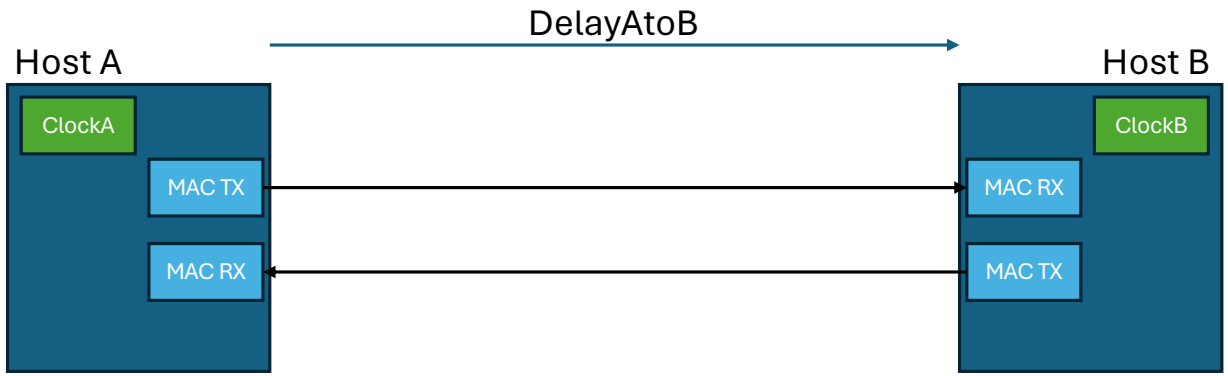
- High accuracy timing is an important feature for many networks
- Achievable accuracy can be limited by architectural issues
  - Even with perfect architecture- you need a **very** careful implementation
- The standard may define behaviors that make higher accuracy more difficult
  - Implementations will need to avoid these if timing accuracy matters
- The current 800GBASE-ER1 baseline has an architectural option that limits timing accuracy in common system configurations
  - Another presentation proposes changes to the baseline to fix this (sluyski_3dj_01_2405)
  - This presentation attempts to give background on what the issue is

Goal



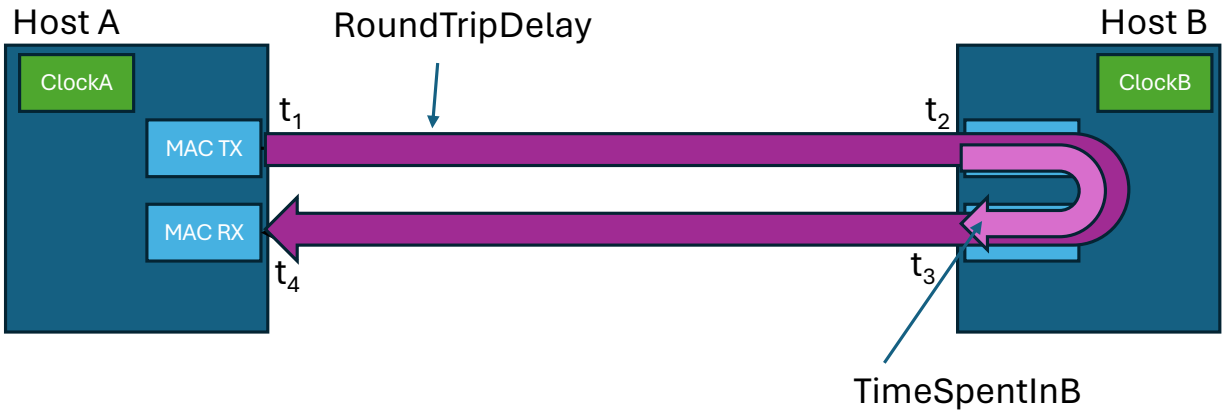
- Two hosts
- Both have clocks
  - assume clocks are counting at the exact same rate
  - clocks don't indicate the same time of day
- Goal is to synchronize the two clocks so that both read the exact same time
  - Until this happens, we can't compare timestamps between the two hosts
  - time deltas calculated using the same clock can be exchanged

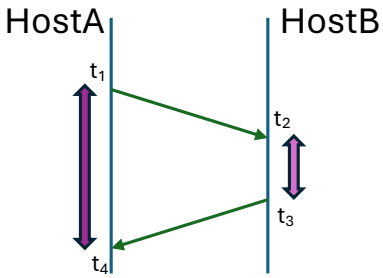
How? (**very** simplified)



- Measure one way delay between the two hosts
- Set ClockB to be (ClockA + DelayAtoB)
  - A can send B a message and B can update its clock

How?





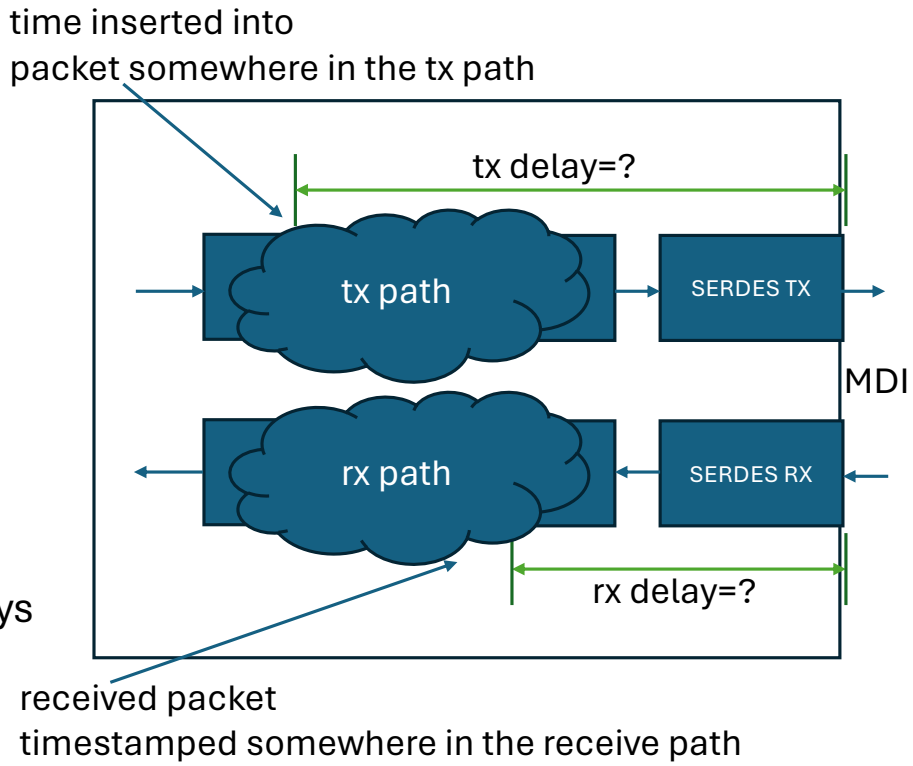
- Measure one way delay between the two hosts
  - Can't actually calculate that, but can estimate with:
    - DelayAtoB = (RoundTripDelay – TimeSpentInB) / 2
    - DelayAtoB = $((t_4 - t_1) - (t_3 - t_2)) / 2$
  - Assumes media is symmetric (hard assumption to avoid in many cases)
  - Assumes we are measuring from the MDI to the MDI
    - This is hard to do- more detail on subsequent slides
- The clocks on the two hosts can't be compared, but differences in time can be used
  - The one-way delay estimate only uses time deltas calculated using the local clock

Timestamp Details

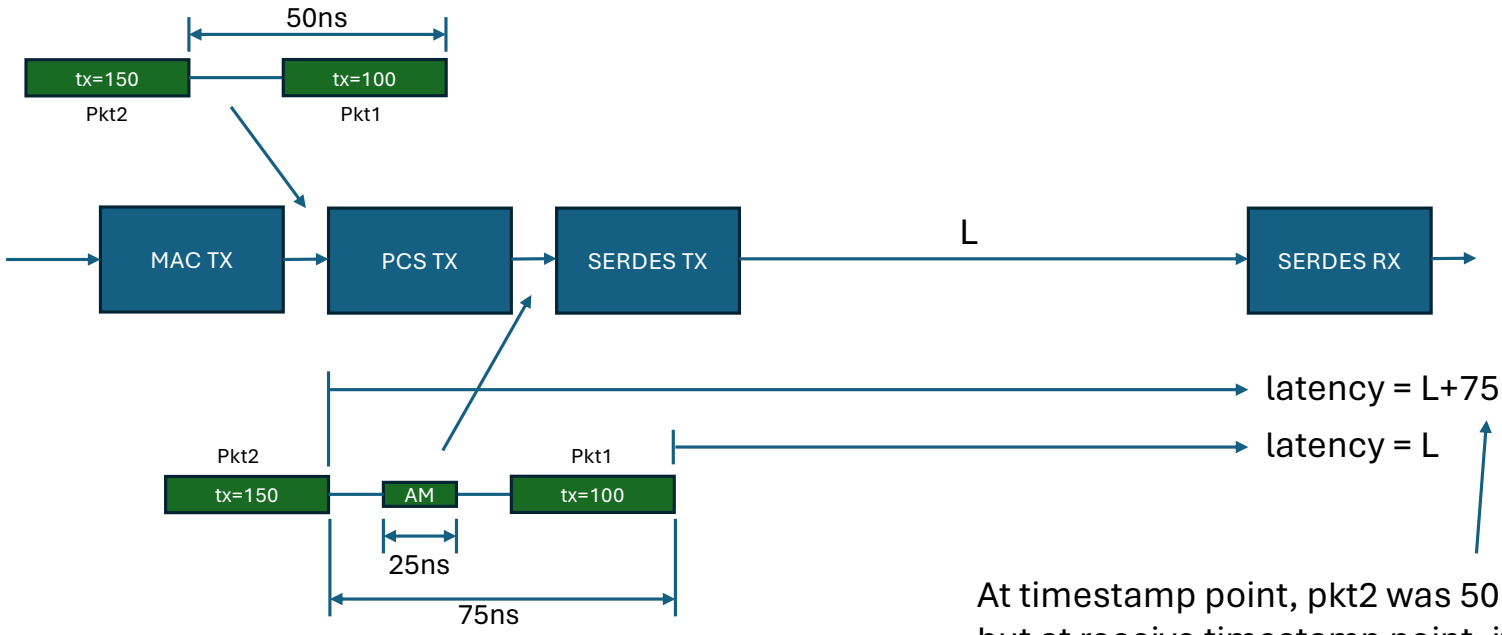
- TX side timestamps
  - these go in the packet
  - need to have a MAC to edit packet data
  - need to have an unencrypted packet (can be encrypted later)
- RX side timestamps
  - these are out-of-band with the packet
  - can’t be sent over the wire

Realistic Implementation Details

- Can't measure at the MDI, so we need to measure from within the chip and account for any error between that point and the MDI.
- If tx/rx delay are constant
  - MDI is effectively at the timestamp point
- If tx/rx delays are not constant
  - Equivalent to media randomly changing length
- Implementation techniques and clause 90 allow designs to adjust for both static and dynamic delays

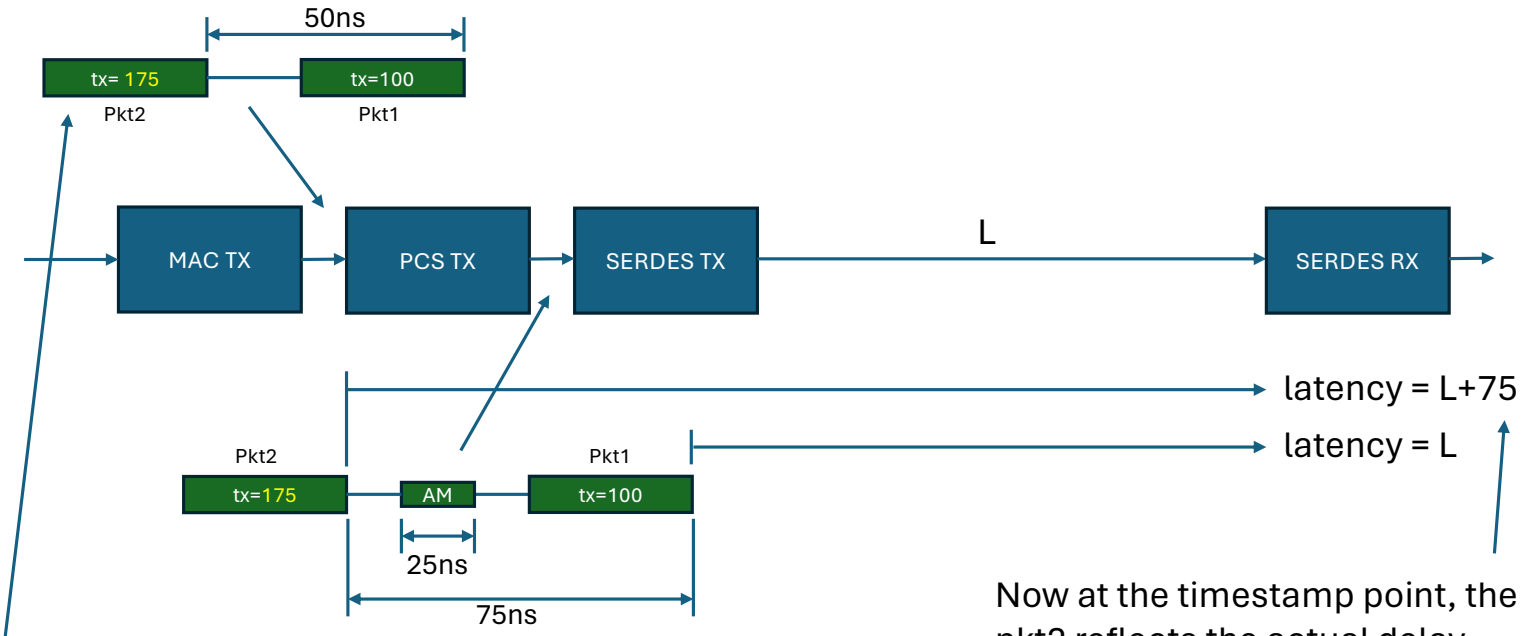


AM insertion picture showing dynamic error



At timestamp point, pkt2 was 50ns behind pkt1, but at receive timestamp point, it is 75ns.

AM insertion picture showing adjustment



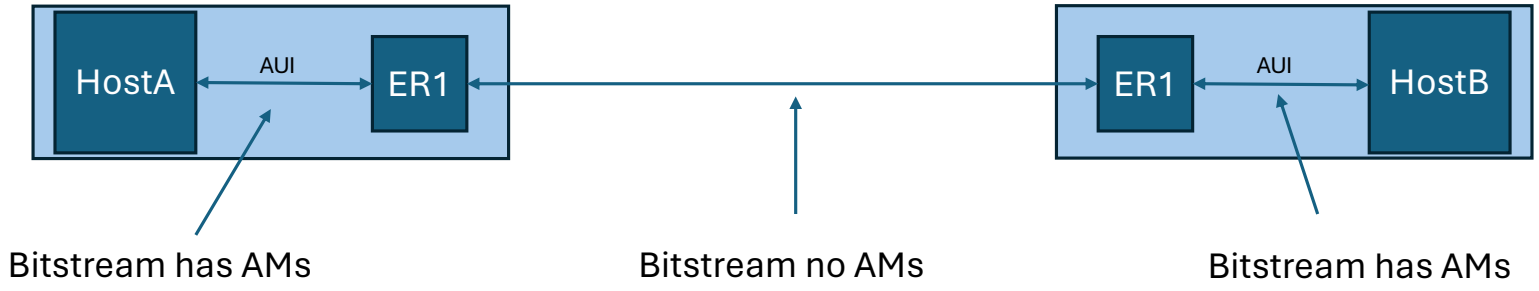
Clause90 and other implementation techniques can pre-adjust for the AM insertion (or idle insert/delete)

Now at the timestamp point, the timestamp in pkt2 reflects the actual delay

Extender Sublayer

- Extender Sublayer is two PCSs separated by an MII
- Standard is written to delete and then re-add alignment markers
  - Assuming the PCSs on either side use them
- Implementations are not required to do this, can leave them in across the MII in between the two PCSs in the XS
  - 400ZR Implementation Agreement takes this approach
- Timestamp adjustments in an XS are tricky
  - XS doesn’t have a MAC so can’t edit packets
  - No other channel to pass information

ER1 Problem



- ER1 encoding doesn't contain AMs
  - payload bitrate is lower than AUI rate
- If ER1 has AUI & XS (normal module case)
  - AMs are deleted when going from AUI->ER1
  - AMs are re-inserted at arbitrary point when going from ER1->AUI
- No guarantee that AMs are re-inserted where they were deleted
  - This changes in the relationship of packets with each other messes with timing
- No tools available to correct for this

ER1 fix

- Two easy fixes
  - Can just map the AMs along with the rest of the bitstream
  - Can remember where the AMs are deleted and then re-insert them exactly in the same spot
- These are functionally equivalent
  - The second one is believed to be an easier change and is backward compatible with the current baseline.
  - Proposal in sluyski_3dj_01_2405

Summary

- The current 800GBASE-ER1 baseline has an issue that limits timing accuracy
- The changes proposed in sluyski_3dj_01_2405 bring the timing accuracy for ER1 in line with the rest of the PMDs defined in 802.3dj