

# Link delay protocol: ( $t_2, t_3$ ) vs ( $t_3 - t_2$ )

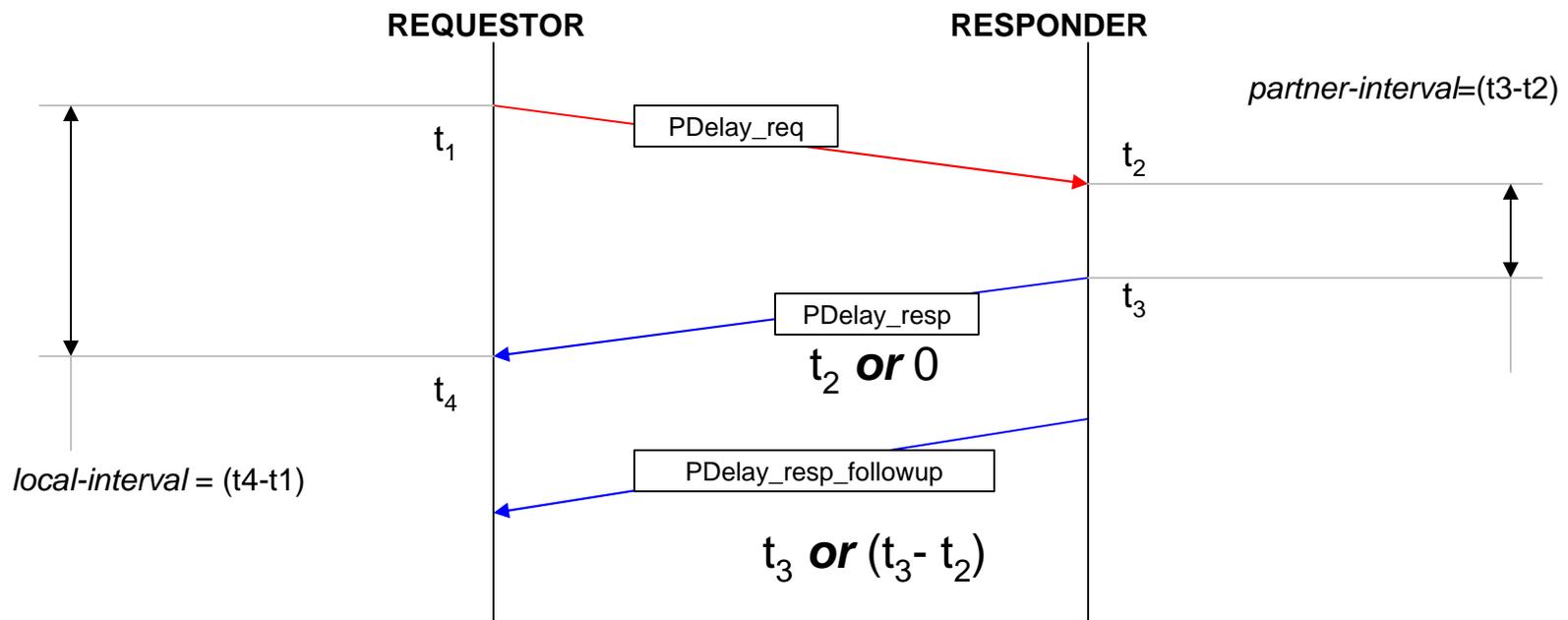
Chuck Harrison

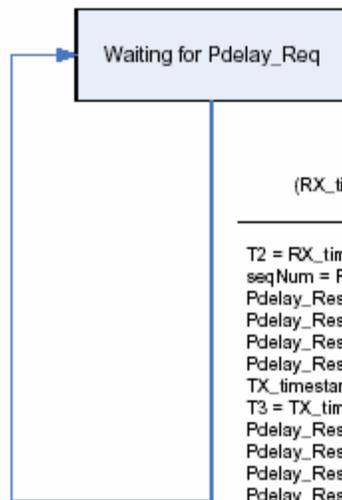
4 Apr 2007

# 1588 Link delay measurement feature: “Pdelay\_Req”

- Consists of a timestamped message exchange between requestor and responder
- Requestor knows his own egress and ingress timestamps ( $t_1$ ,  $t_4$  respectively)
  - $t_4 - t_1$  = gross roundtrip time
- Responder knows his own ingress and egress timestamps ( $t_2$ ,  $t_3$  respectively)
  - $t_3 - t_2$  = responder turnaround time
- Requestor learns  $t_2, t_3$  or  $(t_3 - t_2)$  from responder, computes link roundtrip:
  - $t_4 - t_1 - (t_3 - t_2)$  = net link roundtrip =  $2 * \text{link delay}$
- 1588 provides option of sending  $t_2$ ,  $t_3$  or  $(t_3 - t_2)$

# PDelay\_Response options





```

Integer64 T2, T3, turnaroundTime;
Integer64 TX_timestamp, RX_timestamp;

Integer64 TCreceive(); /* returns timestamp as a
64 bit signed integer, in units of 2^(-16) ns */

Integer64 TCtransmit(); /* returns timestamp as a
64 bit signed integer, in units of 2^(-16) ns */

/* Since TC clock is syntonized and not
synchronized, a full timestamp (seconds plus ns)
is not needed; instead, timestamp can be taken
using a 64-bit syntonized timer that counts in
units of 2^(-16) ns); note that the basic timer
interval can be an integer multiple of this basic
unit if desired */

(RX_timestamp = TCreceive (Pdelay_Req) ) !=
NULL

T2 = RX_timestamp;
seqNum = Pdelay_Req.seqNum;
Pdelay_Resp.seqNum = seqNum;
Pdelay_Resp.correctionField = 0;
Pdelay_Resp.requestingPortIdentity = Pdelay_Req.sourcePortIdentity;
Pdelay_Resp.requestReceiptTimestamp = 0;
TX_timestamp = TCtransmit (Pdelay_Resp);
T3 = TX_timestamp;
Pdelay_Resp_Follow_Up.seqNum = seqNum;
Pdelay_Resp_Follow_Up.correctionField = Pdelay_Req.correctionField;
Pdelay_Resp_Follow_Up.requestingPortIdentity = Pdelay_Req.sourcePortIdentity
Pdelay_Resp_Follow_Up.responseOriginTimestamp = 0;
turnaroundTime = T3 - T2;
Pdelay_Resp_Follow_Up.correctionField += turnaroundTime;
TCtransmit ( Pdelay_Resp_Follow_Up);

... = subnanosec(T2); // if subnanosecs used
... = T2;
... = T3;
... = subnanosec(T3) // if used
  
```

Figure 2-1. State machine, and corresponding C code, for Pdelay responder P2P TC port

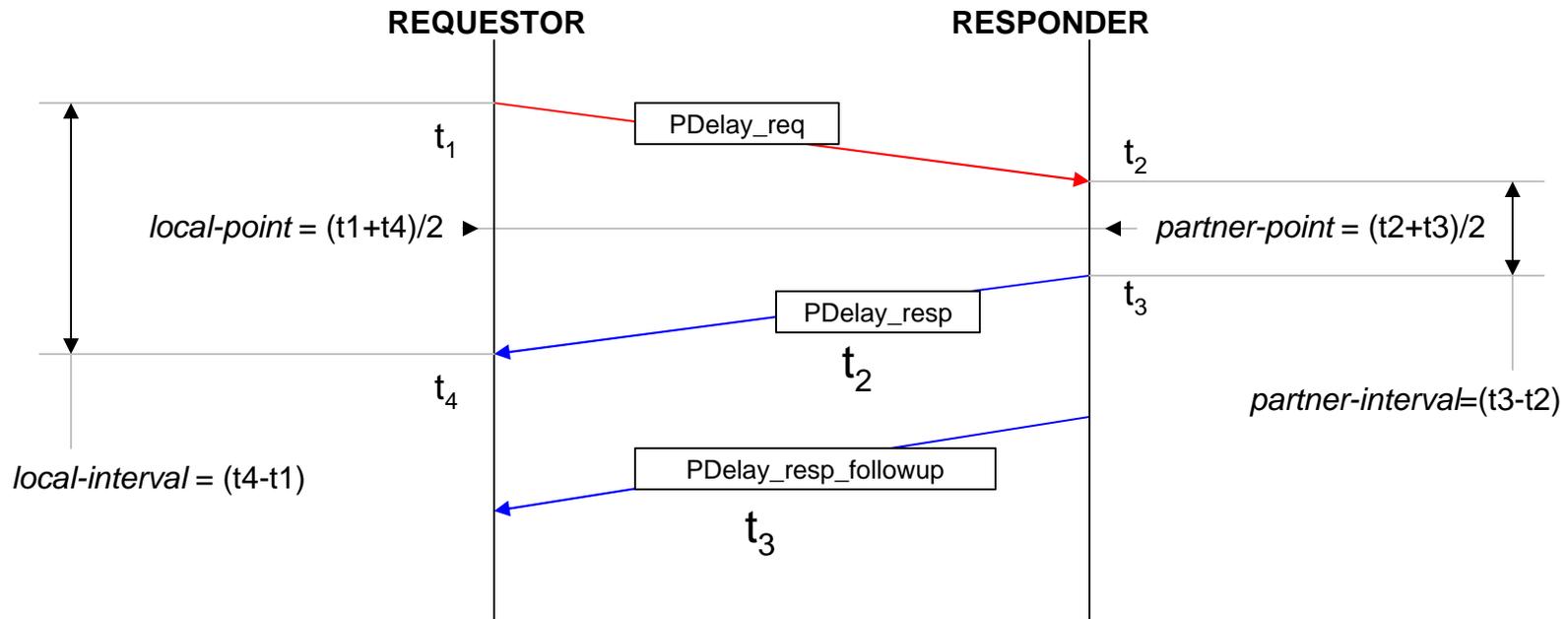
Ref: as-garner-protocol-state-machines-frame-formats-0307.pdf

# Why?

- Transfer more information at no increase in bandwidth or implementation complexity
- Extra information allows correlation of device epochs and tracking of freq offset
  - Turnaround time can be compensated for freq offset
  - Additional non-1588 services (e.g. Network Event, Proximity Control) can be implemented

# Epoch correlation

(supports frequency offset compensation)



# Summary

- 1588 allows two alternative modes of operation for PDelay\_Response
- We should choose just one for AVB profile compliance
- Choosing  $(t_2, t_3)$  rather than  $(t_3 - t_2)$  costs almost nothing and provides more functionality
- Let's choose  $(t_2, t_3)$  as standard.