

# One-step for 802.1AS-REV

Michael Johas Teener  
[mikejt@broadcom.com](mailto:mikejt@broadcom.com)

# Problem

- There is a perception that two-step processing (sync/follow-up):
  - is less accurate than one-step
  - uses excessive processing in an intermediate system
  - reduces responsiveness because of longer residence times
- **Some IEEE 1588 profiles require one-step**
  - not providing one-step in 802.1AS is a potential obstacle in the “grand unification” of 1588 and 802.1AS

# Accuracy?

- **Potential problem if two-step adds significantly to residence time in an intermediate system**
  - and the clock used to measure residence time is not accurate
- **802.1AS provides good tools to minimize this effect**
  - intermediate system syntonization, neighbor rate ratio
- **Real-world implementations seem to work very well**
  - AVnu testing has validated

*note: this would be a great paper if someone would publish it*

# Processing

- **Two-step was chosen since the hardware is simple**
  - just a local time stamp
  - most processing can be done in “software”
- **for very large intermediate systems, this can be an issue**
  - receive two packets per slave port, compute and transmit two packets per master port, 8 times/sec (per domain)
  - for AS-REV =  $nPorts * 2 * 8 * 4 = nPorts * 64$  packets/sec
    - for a 16 port switch, this is 1k packets/sec
    - for a 1024 port switch, this 64k packets/sec
- **This does not seem excessive, but ...**
  - the author has received a number of complaints
  - some industrial networks will use 16-32 Hz sync rate

# Responsiveness

- **Two-step forces longer residence time**
  - at least the time it takes to receive follow up
  - usually includes software / cpu processing time
  - 802.1AS allows up to 10ms residence time
- **One-step has “just” queuing delays**
  - 100Mbit/sec -> 120 $\mu$ s delay behind 1.5k packet
  - plus some overhead
- **64-hop network**
  - up to 640ms delay for 802.1AS
  - up to 7.7ms delay for “one step”
- **Is this a problem?**
  - maybe, for some applications

# Electropolitics and grand unification

- **1588 defines one-step**
- **It's already used in both telecom and industrial applications**
- **Unification with 1588 in the future is desirable**
- **Not having it is an obstacle**

# Changes needed

- **Carry 802.1AS timing info TLV in Sync, not Follow\_Up**
- **Rate ratio calculation:**
  - change meaning of fields in Pdelay\_Response
  - use two step
- **Interoperation with two-step and 802.1AS-2011**

Geoff Garner outlined choices in <<http://www.ieee802.org/1/files/public/docs2012/as-garner-discussion-asbt-feature-assumptions-v02.pdf>>

# Move info TLV to sync

- 802.1AS uses the Follow\_Up to carry useful information
  - move it to the Sync

Table 11-10—Follow\_Up information TLV

Bits								Octets	Offset
8	7	6	5	4	3	2	1		
tlvType								2	0
lengthField								2	2
organizationId								3	4
organizationSubType								3	7
cumulativeScaledRateOffset								4	10
gmTimeBaseIndicator								2	14
lastGmPhaseChange								12	16
scaledLastGmFreqChange								4	28



# One-step pDelay

- Carry the responseOriginTimestamp (i.e., the timestamp of the sending of Pdelay\_Resp) in the requestReceiptTimestamp field of Pdelay\_Resp
  - This can be done because IEEE 1588 specifies that in the one-step case the requestReceiptTimestamp field is set to zero, and the difference  $t_3 - t_2$  is carried in the correctionField, but then we won't be able to carry the sub-ns part of the Pdelay\_Resp
  - Instead, carry the low-order 6 octets of the responseOriginTimestamp in the high order 6 octets of the field plus the 2 octet sub-ns part.

# Two-step pDelay?

- **pDelay is infrequent**
  - 1 per second, NOT duplicated for domains (or at least it shouldn't be)
  - low processing load
- **pDelay is NOT relayed**
  - processing is local anyway, hardly anything to be gained with one-step
- **Suggestion:**

***don't do one-step pDelay***

# "Legacy" compatibility

- One-step receive capability included in BMCA
- Use the twoStepFlag in the common header
  - If twoStepFlag is false in an announce message, then the port sending it can \*receive\* one-step sync
  - Current 802.1AS requires that twoStepFlag always be true, and ignored on reception

	announce transmitter announce receiver	twoStepFlag set (only accept two step)	twoStepFlag clear (can receive one step)
two step only (802.1AS-2011 or 802.1AS-REV two step only)		ignored, will send back only two step	ignored, will send back only two step
one step rx OK (802.1AS rev one step capable)		accepted, will send back only two step	<b>accepted, will send back one step ONLY if capable</b>

# Conclusion

- I think we ***must*** specify one-step operation
  - If for no other reason, then for marketing/ electropolitical purposes
- It's a straight-forward operation
  - I'll help Geoff get it in the draft
- There are **NO** requirements placed on systems that do not implement it