1-step for 802.1AS Details

(v1, 7-April-2015)

Michael Johas Teener
mikejt@broadcom.com

Agenda

- Review of current proposal
- New work
- Next steps

Review

- **Most changes in Clause 11 (full-duplex point-to-point) media dependent layer**
  - port can be “1step-capable”
  - if so capable, port can be in “1step-operation”
- **Intention is that media independent layer is almost the same**
  - existing path unchanged
  - need to propagate the received sequenceId
- **Notes and annex to explain implications of mixed 1step/2step networks**

Media independent

- **Clause 10 updates:**
  - PortSyncSync, MDSynchSend and MDSynchReceive new items
    - rxSequenceId - holds sequenceId from current slave port
    - 1stepSlave - set if slave port is operating in one step mode
    - rxRateRatio* - the rateRatio received on the current slave port
    - rxSourcePortIdentity* - the sourcePortIdentity received on the current slave port
  - *May want to have different PortSyncSend and PortSyncReceive structures
    - just to make things clear

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

MDSyncReceiveSM

- **No changes for 1step except:**
  - populating the MDSyncReceive structure from the Sync event message
    - including the TLV
  - set the 1stepOperation flag if appropriate
  - include the sequenceId value

MDSyncSendSM

- **If port is operating in 1Step mode:**
  - if MDSynchSend.1StepOperation set then
    - we know the slave port is 1Step and we want to operate like a 1588 transparent clock (1step TC)
  - if MDSynchSend.1Step not set then
    - we operate just like a 802.1AS port except we send a one step synch (1step master)
  - details follow
- **If a port is not operating in 1Step mode:**
  - no changes from 802.1AS-2011

MDSendSynchSM (1step TC)

- **Build sync from MDSendSync structure**
  - uses the upstreamTxTime and egress timestamp (and other latency info) to add the residence time to the correction field
  - uses rxSequenceId for the sequenceId
  - uses rxSourcePortIdentity for sourcePortIdentity
  - uses rxRateRatio for cumulativeScaledRateOffset
- **Requirement that send synch happen "as soon as possible after receive synch"**

MDSendSynchSM (1step master)

- **Slave port is 2step, so we need to synthesize the 1step**
- **Build sync from MDSendSync structure**
  - uses the upstreamTxTime and egress timestamp (and other latency info) to add the residence time to the correction field (same as before)
  - uses sequenceId for the sequenceId
  - uses sourcePortIdentity for sourcePortIdentity
  - uses rateRatio for cumulativeScaledRateOffset
- **In other words, same values as used in 2step**

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

"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 ***1stepOperation = false***	ignored, will send back only two step ***1stepOperation = false***
one step rx OK **(802.1AS rev one step capable)**	accepted, will send back only two step ***1stepOperation = false***	**accepted, will send back one step ONLY if capable** ***1stepOperation = true***

Notes on hybrid operation

- "Hybrid operation" means the path back to the GM includes both 1step and 2step links.
- There are three fields in sync/follow-up that now have possibly different meanings:
  - sequenceId
  - sourcePortIdentity
  - cumulativeScaledRateOffset

sequenceID

- **as far as I can tell, sequenceID is not relevant end-to-end, it's just a link parameter**
  - only used to correlate sync with corresponding follow-up
- **for a "TC path" (1step slave port, 1step master port), sequenceId is repeated …**
  - never tested or validated
  - but always incrementing at nearest upstream BC (2step port) or GM

cumulativeScaledRateOffset

- for a “TC path” the cumulative rate ratio is unchanged … the residence time is so short, this should not be significant
  - I’ve asked Geoff to do some simulations to test this hypothesis

sourcePortIdentity

- the sourcePortIdentity is the identifier of the closest upstream GM or BC
- for 802.1AS-2011, all TAS's are BCs
- for 802.1AS-rev, I propose that a "TC path" is NOT a BC
  - meaning that sourcePortIdentity is just like the 1588 meaning
- it's possible we could redefine sourcePortIdentity for 802.1AS-rev
  - it could be "grand master identity"
  - but that would be breaking 1588, perhaps

1588 implications

- **1588 has no concept of different ports doing different things**
  - like 1step and 2step in the same device
  - but then they don't define ethernet and wifi ports, either
- **Port capabilities in announce messages**
  - help their plug-and-play, they were thinking about things like this for profile interoperation
- **We will have to go to them with this idea as part of their new layered structure**
  - they might actually like the idea

All done!