1-step for 802.1AS Details

(v4, 7-May-2015)

Michael Johas Teener

mikejt@broadcom.com

Agenda

- Review of current proposal
- Media independent (Clause 10)
- Media dependent for Ethernet (Clause 11)
- Coordination with 1588

Review

- **Most changes in Clause 11 (full-duplex point-to-point) media dependent layer**
  - port can be “oneStepCapable” (per port global variable array)
  - if so capable, a master port can be in “oneStepOperation” (port global)
    - slave port operation updated to support 1step, but no mode change, just a “oneStepSync” flag set in the MDSyncReceive data.
  - a master port in “oneStepOperation” can operate like a transparent clock (TC)
    - meaning not updating some sync fields
    - which can happen only if the current slave port is in “oneStepOperation” and the sync rate is the same
    - ***not required*** … it’s OK if a master port does update all fields
- **Intention is that media independent layer is almost the same**
  - existing path unchanged
  - use Signaling to communicate “oneStepCapable”
  - need to propagate some of the unprocessed received fields
- **Notes and annex to explain implications of mixed 1step/2step networks**

Media independent

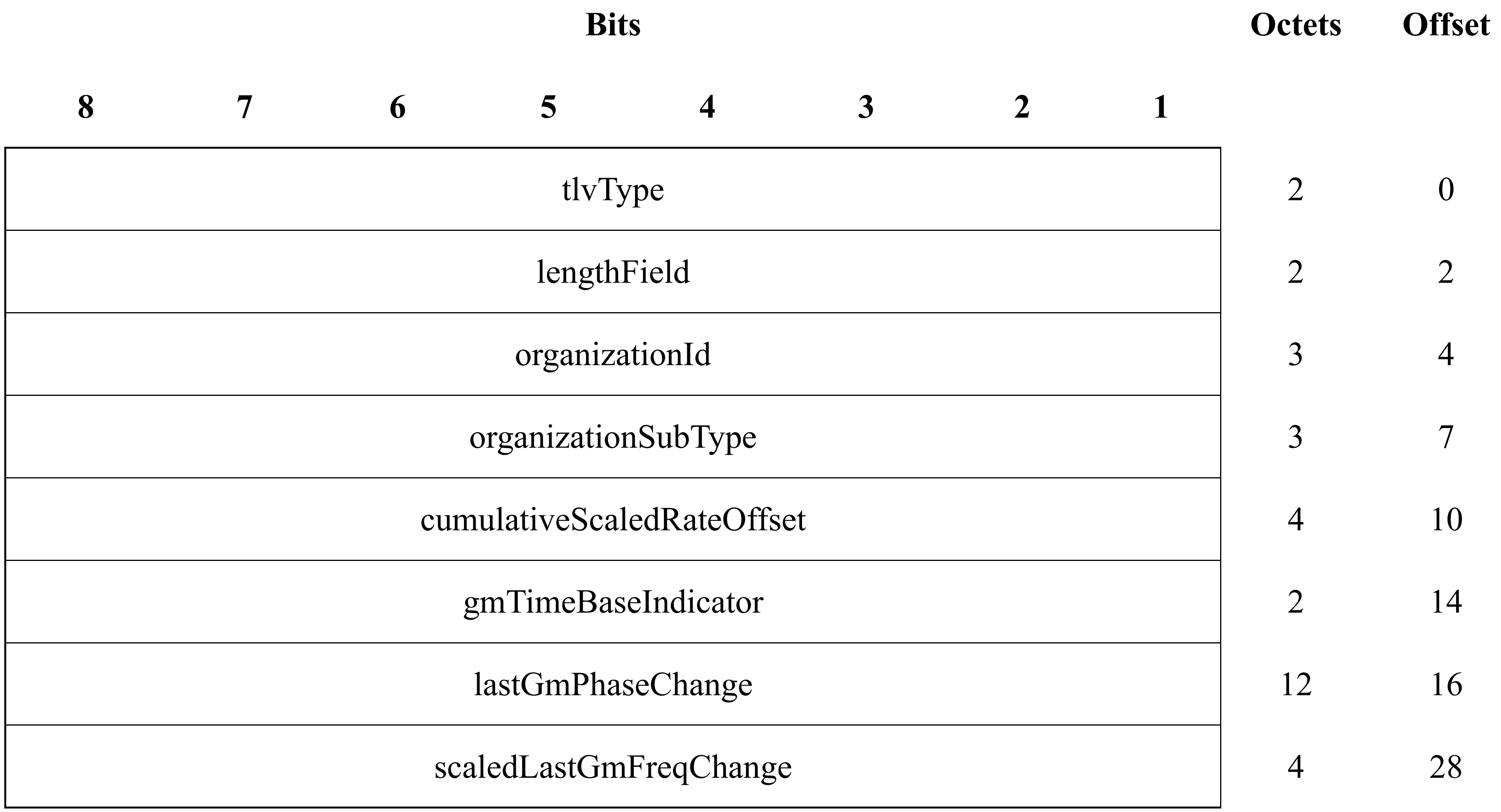
- **Clause 10 state machines unchanged except**
  - master ports (sending sync) get some extra information propagated from the slave port (receiving sync)
    - perhaps always propagate the received sourcePortIdentity
  - Add TLV to Signalling message to communicate “oneStepCapable”
- **Clause 10 data structures:**
  - Separate PortSyncSend and PortSyncReceive
  - MDSync<x> and PortSync<x> have new fields
    - rxSequenceId - holds sequenceId from the current slave port
    - oneStepSync - set if slave port received a one step sync
  - <y>SyncSend have new fields:
    - rxSourcePortIdentity - received on the current slave port
    - rxCumulativeRateOffset - received on the current slave port

<x> is “Send” or “Receive”, <y> is “MD” or “Port”

Move info TLV to sync

- 802.1AS uses the Follow_Up to carry useful information
  - move it to the Sync if oneStepOperation is true

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 one step except:
  - populating the MDSyncReceive structure from the Sync event message
    - including the TLV
  - set the oneStepSync flag if appropriate
  - include the sequenceId value

MDSyncSendSM

- **If port is operating in one step mode:**
  - if oneStepOperation[rcvdPSSyncPtr] && (syncInterval[rcvdPSSyncPtr] == syncInterval[txPSSyncPtr]) && TCOperation[[txPSSyncPtr] then “transmit like a TC”
    - we know the slave port is one step and using the same synch rate, so we can operate like a 1588 transparent clock
    - ***not required***, unless the group decides that it should be
  - if !!oneStepOperation[rcvdPSSyncPtr] || (syncInterval[rcvdPSSyncPtr] != syncInterval[txPSSyncPtr]) || !TCOperation[[txPSSyncPtr] then “one step, not TC”
    - we operate just like a 802.1AS port except we send a one step synch (one step master)
  - details follow
- **If a port is not operating in one step mode:**
  - no changes from 802.1AS-2011

MDSendSynchSM #1

(“transmit like a 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
    - rate ratio adjustment optional (see “residence time error” in a later slide)
  - uses rxSequenceId for the sequenceId
  - uses rxSourcePortIdentity for sourcePortIdentity
  - cumulativeRateOffset must be updated ***unless*** the device on a slave port does not compute the neighbor rate ratio calculation (sets it to 1.0)
    - using signaling to set “computeNeighborRateRatio” to false
    - then the rxCumulativeRateOffset could be repeated (I believe)

MDSendSynchSM #2

(“transmit like a TC”)

- Requirement that send synch happen “as soon as possible after receive synch”
    - not certain how this can be specified
    - worst case residence time is one variable for a “time fidelity report”

MDSendSynchSM

(“one step, 802.1AS-2011”)

- Slave port is two step or different sync rate, so we need to synthesize the one step event message
  - or we just don’t want TC-like operation
- 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)
    - rate ratio adjustment optional (see “residence time error” in a later slide)
  - uses sequenceId for the sequenceId
  - uses sourcePortIdentity for sourcePortIdentity
- In other words, same values as used in two step

residence time error

- If we don’t adjust residence time using rateRatio, there is an error
  - (ratio error)*(residence time) or:

ratio error residence time	200ppm	100ppm	50ppm
250 *μs*	50ns	25ns	10ns
100 *μs*	20ns	10ns	5ns
25 *μs*	5ns	2.5ns	1ns
10 *μs*	2ns	1ns	0.5ns

- this might be OK, it would be a <u>*cost*</u> of using TC mode
  - reported using new clock fidelity impairment reporting facility

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

Backwards compatibility

- Signaling is currently used to communicate with peer port
  - rate for pDelay, sync and announce, computation of neighbor rate ratio, etc.
- Add one-step ***receive*** capability in the Signaling message
  - Hmm. I notice that we never define when Signaling messages are sent.
    - I also note that sometimes it's "Signaling" and sometimes it's "Signalling"
  - Should be sent about the same time pDelay process starts
- New TLV in Signaling message
  - one field in TLV is "oneStepCapable"
  - If oneStepCapable is true, then the port sending it can ***receive*** one-step sync

announce transmitter announce receiver	oneStepCapable false (only accept two step)	oneStepCapable true (can receive one step)
two step only **(802.1AS-2011 or 802.1AS-REV two step only)**	ignored, will send back only two step ***oneStepOperation = false***	ignored, will send back only two step ***oneStepOperation = false***
one step rx OK **(802.1AS rev one step capable)**	accepted, will send back only two step ***oneStepOperation = false***	**accepted, will send back one step ONLY if capable** ***oneStepOperation = true***

Notes on hybrid operation

- "Hybrid operation" means the path back to the GM includes both one step and two step 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” through a TAS, sequenceId is repeated …
  - never tested or validated
    - consider the case of transition from TC-like to non-TC-like and vice-versa
  - but always incrementing at nearest upstream TAS (non-TC path) or GM
- perhaps just require that each sequenceID is different than the previous “n” sync messages
  - where “n” is TBD, maybe 4

cumulativeScaledRateOffset

- for a “TC path” the cumulative rate ratio *may* be unchanged
  - requires that the downstream device not compute neighbor rate ratio
    - use signaling message to do that
  - need to understand the effect on downstream TAS operation … need the “clock fidelity distortion”
    - off-topic, but important, is to validate clock accuracy
- clearly, it’s OK if the cumulative rate ratio is updated
  - that’s just 802.1AS-2011

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"
    - ***I think this would be really useful!***
  - but that would be breaking 1588, but perhaps if we ask 1588 to allow profiles to make this change

1588 implications

- 1588 TCs don’t necessarily wait for a follow up on a master port
  - so they don’t/can’t convert a two step sync into a one step synch
- 1588 has no concept of different ports in a TC doing different things
  - like one step and two step in the same device
  - but then they don’t define ethernet and wifi ports, either
- Port capabilities in announce or signaling messages?
  - help their plug-and-play, they were thinking about things like this for profile interoperation
- Requests to 1588
  - Allow these features to be included in a profile spec
  - Perhaps part of the new layered architecture

All done!

document history		
v1	2015-04-07	initial version, TSN call 2015-04-08
v2	2015-04-08	separate out “TC” mode, fix names, agenda
v3	2015-05-03	updates to for final discussion on AS call 2015-05-04
v4	2015-05-07	effect of not computing cumulative rate ratio, requirements for that, and note about loss of accuracy if rate ratio not used residence time correction
v5	2015-05-19	additional notes, change names to be consistent, using signaling for port capabilities