Discussion of Assumptions for 802.1ASbt Features

Revision 2

Geoffrey M. Garner
Consultant

IEEE 802.1 AVB TG
2012.07.19

gmgarner@alum.mit.edu

Introduction

- This presentation provides discussion for potential assumptions for 802.1ASbt
  - The objective is to clarify the assumptions sufficiently so that an initial draft for 802.1ASbt can be produced
- This revision is prepared to indicate what would be needed if feature 1 were to include full one-step support (for both transmit and receive)
  - This material was requested during the presentation of Revision 1
  - As a result, the feature 1 title is changed to reflect this
  - To avoid confusion with feature 1 as described previously, we label this "feature 1a"
  - We also have added another possible solution to measuring rate ratio using one-step Pdelay messages

List of Features - 1

- The following 802.1ASbt features are in the PAR [1], and have been discussed at various times in the 802.1 AVB TG (see, for example, [2]) (table is continued on next slide) (except that feature 1 is replaced by a more generalized “feature 1a”, whose description was requested)
- The order of the features here is chosen for convenience; those feature perceived by the editor to be more clearly defined right now are listed first (the order differs from that in [1] and [2])

Feature	Description
1a	Support for one-step clock
2	Incorporation of the interfaces specified in IEEE Std 802.3bf into the IEEE 802.3 full-duplex media-dependent layer model.
3	Support for new media types, with corresponding media-dependent layers, e.g., IEEE Std 1901 and WiFi direct
4	Management support for automatic measurement of link delay asymmetry.
5	Support for link aggregation (IEEE 802.AX, IEEE 802.1AXbk, IEEE 802.1AXbq
6	Support for redundant paths.

List of Features - 2

- The following 802.1ASbt features are in the PAR [1], and have been discussed at various times in the 802.1 AVB TG (see, for example, [2]) (table is continued on next slide)
- The order of the features here is chosen for convenience; those feature perceived by the editor to be more clearly defined right now are listed first (the order differs from that in [1] and [2])

Feature	Description
7	Carrying information on alternate time scales (e.g., local time for a respective time zone).
8	Improved performance (e.g., improved grandmaster changeover time, longer chains of time-aware systems).
9	Enhancements to the determination of asCapable (e.g., longer cable lengths, new media types).
10	Additional parameter sets for non-Audio/Video applications, e.g., industrial control

Feature 1 – Interoperability with One-Step Clock on Receive – 1

- This feature is relatively well defined, at least for Sync
- For Sync
  - Allow twoStepFlag to be FALSE in Table 11-4
  - Also in Table 11-4, now must pay attention to twoStepFlag (and not ignore it on receive)
  - Since there is no Follow_Up message, the Follow_Up TLV must be carried in the Sync message (we now refer to it as the “Sync/Follow_Up TLV)
  - In MDSyncReceiveSM state machine (11.2.13 and Figure 11-6), need to add logic for case where twoStepFlag is FALSE
    - If twoStepFlag is FALSE, do not wait for Follow_Up, and process correctionField and Sync/Follow_Up TLV in Sync as they would be processed if they were in Follow_Up message
  - In MDSyncReceiveSM state machine, if twoStepFlag is TRUE, followUpCorrectionField in the MDSyncReceive structure is set to the sum of the most recently received Sync and associated Follow_Up message correction fields, because, since now one-step operation is possible, the Sync correctionField may have timing information

Feature 1 – Interoperability with One-Step Clock on Receive – 2

❑For Sync (cont.)

- In MDSyncSendSM state machine, change the name of the SEND_FOLLOW_UP block to SET_CORRECTION_FIELD
  - The current setFollowUp function is invoked in the SEND_SYNC block; it does all the current items, but for the Sync message instead of the Follow_Up message, except the processing of the correction field (item (a) in its description)
  - The boolean notification variable rcvdMDTimestampReceive is received in between the SEND_SYNC and SET_CORRECTION_FIELD blocks
  - The SET_CORRECTION_FIELD block processes the Sync correctionField in the same manner as step (a) of the current setFollowUp function. This block also resets the rcvdMDTimestampReceive boolean.
    - This processing is done as the Sync message is being transmitted (this is the essence of one-step behavior)

Feature 1 – Interoperability with One-Step Clock on Receive – 3

❑Need to decide what to do regarding the Pdelay messages

- Both Pdelay_Resp and Pdelay_Resp_Follow_Up messages are used in nearest neighbor rate ratio measurement
- In one-step Pdelay, only Pdelay_Resp is sent, and it carries difference between its send time and the Pdelay_Req receipt time
  - This is sufficient information for propagation delay measurement, but not for neighbor rate ratio measurement
- Some possible solutions are
  - Don't handle one-step Pdelay messages on receive (i.e., only handle one-step Sync)
  - 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 t3 – t2 is carried in the correctionField
    - –But this would not allow any sub-ns component of the timestamp of sending Pdelay_Resp to be carried
    - –Also, this would be a specification in 802.1AS; it would be necessary that the one-step system that sends Pdelay_Resp complies with this (probably would want to request this be added to 1588v3)

Feature 1 – Interoperability with One-Step Clock on Receive – 4

❑Need to decide what to do regarding the Pdelay messages (cont.)

- Some possible solutions (cont.)
  - Carry the responseOriginTimestamp in a TLV attached to Pdelay_Resp
    - In this approach, a TLV of the same length must be added to Pdelay_Req so that Pdelay_Req and Pdelay_Resp have the same length
    - This is done to ensure that there is no error in measured propagation delay in the event there is an unknown or undetected asymmetry that depends on message length
    - This Pdelay_Req TLV would likely not carry any useful information; its purpose would be only to ensure that Pdelay_Req and Pdelay_Resp have the same length
  - Invent a new mechanism for neighbor rate ratio measurement for this case (i.e., other than the new mechanisms above)
  - Others?
- The intent is that the AVB TG would pick a solution for measuring rate ratio (either one of the above or another solution)
  - It is **not** intended to have multiple options

Feature 2 – Incorporation of IEEE 802.3bf Interface

- This feature is relatively well defined
- Replace MDTimestampReceive primitive (11.2.9) with appropriate 802.3bf primitives
- Modify MDSyncSendSM, MDSyncReceiveSM, MDPdelayReq, and MDPdelayResp state machines to use the 802.3bf primitives and mechanisms
- May be able to re-use aspects of early v1 drafts (D2.0 and earlier)
  - These used older interface model that had some similarity to 802.3bf model

Feature 3 – Support for New Media Types

- In accordance with the discussion in the November, 2011 IEEE 802.1 AVB TG meeting, WiFi direct will not be included
- In accordance with the discussion in the November, 2011 IEEE 802.1 AVB TG meeting, IEEE Std 1901 can be handled as a type of CSN
  - Add entry for IEEE Std 1901 in Table E.2
  - Add a subclause E.6.3 entitled “IEEE Std 1901 behavior”
  - Add appropriate reference(s) to clause 2
- The above items must be supplied to the Editor, but at least placeholders can be included in the initial draft

Feature 4 – Mgt Support for Automatic Measurement of Link Asymmetry

- Some approaches for automatic measurement of link asymmetry are described in [6] and [7]
  - as-huang-compensation-for-physical-line-asymmetry-0311.pdf
  - asbt-huang-measurement-of-link-delay-asymmetry-1031-v00.pdf
- The editor will go through these documents to determine what managed objects and associated MIB variables are needed to support these features
- It will also be considered whether signaling message(s) are needed (and, if so, associated state machines)
- The author of [6] and [7] will be consulted on these items as well
- A presentation will be generated to summarize what is needed, before adding the material to the initial draft
- However, a placeholder and/or Editor's Note will be included in the draft to indicate this work is ongoing

Feature 5 – Support for Link Aggregation

- Link aggregation is mentioned in 11.2.5 of 802.1AS
- Indicates that link aggregation is not specified; however, if it is used:
  - Sync and Pdelay_Req messages must be part of the same conversation
  - There can be error in time synchronization if Pdelay_Resp uses a different physical link than Sync and Pdelay_Req
    - Error is equal to the absolute value of one-half the difference in the delays in the two directions
- To run the 802.1AS protocol over a link that consists of two or more aggregated physical links, we need a mechanism to ensure that Sync and Pdelay_Req are part of the same conversation, and that Pdelay_Resp uses same physical link as Sync and Pdelay_Req
- Presentation(s) needed on this
  - i.e., do there exist mechanisms that would enable this to be done?
  - If information on this is not obtained in time for the first draft, a placeholder or Editor's note will be included indicating that information is needed

Feature 6 – Support for Redundant Paths – 1

- There was some discussion of this in the May, 2012 AVB TG meeting, but in the more general context of overall redundancy in AVB Gen 2
- Based on the discussions, redundancy with respect to synchronization seems to relate to having a hot standby GM (or, in general, multiple hot standby GMs, though whether the number is two or more than 2 is not of central importance to this feature)
  - In the current meeting, there also was discussion of redundant paths from one GM to any time-aware system [8] (more on this shortly)
- In the event of failure of the current GM, the hot standby GM would immediately take over as GM
  - The new GM would supply synchronization to the network and, *presumably*, there would be negligible impairment (i.e., jitter, wander, phase transient) during the changeover)
    - The reason we say *presumably* is that there would at least be a phase change due to any difference in the frequency and phase noise of the old and new GMs (see below)

Feature 6 – Support for Redundant Paths – 2

- Based on the discussion in the May, 2012 meeting and the general notion of a "hot standby", we assume that:
  - The hot standby GM is chosen when the current GM is chosen, as well as the synchronization hierarchy (i.e., synchronization spanning tree) for the hot standby GM
  - The hot standby GM is always sending Sync messages, and each time-aware system is computing the synchronized time relative to the hot standby GM in addition to the synchronized time relative to the current GM
  - If the current GM goes away, the hot standby GM becomes the new GM, and a new hot standby GM is chosen (along with its associated synchronization spanning tree)
    - In general, if there are *N* – 1 hot standby GMs and either the current GM or one of the hot standby GMs goes away, all the GMs of priority lower than the one that goes away increase their priority by one level, and a new lowest priority GM hot standby GM is chosen

Feature 6 – Support for Redundant Paths – 3

- Note that, while the current and hot standby GMs each send Sync and Announce messages, the Pdelay mechanism and the associated computation of link delays and neighbor rate ratios need only be done once per link (for each direction)
  - This is because neighbor rate ratios are based on free-running local clock frequencies, and the difference in propagation time measured in the time bases of the different local clocks or different GMs is approximately the same
- In addition, the discussion of SPB and IS-IS in the current meeting [8] discussed the idea of alternate redundant paths between two nodes, with duplicates of each frame taking each path (and where a single failure cannot cause both paths to fail)
  - In the context of Synchronization, this would correspond to Sync messages being sent from the active GM to each time-aware system over multiple paths
  - Note that, unlike the case of traffic, the redundant Sync messages can be used to improve the time estimate (or, might choose not to use a redundant message with an earlier timestamp from the latest message used)

Feature 6 – Support for Redundant Paths – 4

- However, assuming the above basic assumptions are correct, there are still many details to be worked out, e.g.,
  - How are the multiple spanning trees created? The discussion in the May, 2012 meeting seemed to favor using IS-IS with the Dijkstra algorithm to determine the “shortest” (i.e., best) paths
  - Presumably this alternate BMCA (IEEE 1588 allows an *alternate* BMCA to be used instead of the default BMCA describe in clause 9; the current 802.1AS BMCA actually is an alternate BMCA, though it is very similar to the default BMCA) would be run regularly, because it would have to react not only to the current or hot standby GM going away, but to any topology change (e.g., loss of link or node)

Feature 6 – Support for Redundant Paths – 5

❑Details (cont.)

- Presumably, each time-aware system would need to process separately the Sync messages received from each GM, and maintain the time relative to each GM
  - It appears that this would be a straightforward extension of the current 802.1AS state machines related to synchronization (i.e., as opposed to best master selection)
  - It would be necessary to have a means of distinguishing which GM a Sync message's information pertains to
    - –In a sense, the maintaining of time relative to the current and hot standby GMs is similar to maintaining time relative to different domains; this is because each domain has a GM whose time must be maintained (and, carrying this further, the GM and synchronization spanning tree must be chosen for each domain)
    - –It appears that the same amount of computation per GM is done, whether the multiple GMs are for redundancy or for different domains
    - –To distinguish the GMs, the domain number field could be used
    - –Alternatively, the GM clockIdentity could be added in a TLV for Sync or Follow_Up (creating a new field in the Sync and Follow_Up messages would require a change to 1588 itself; note that it already is present in the Announce message)

Feature 6 – Support for Redundant Paths – 6

- Details (cont.)
  - If multiple, redundant paths are set up, could have redundant paths from each GM (but, as indicated above, could use all the Sync information, from the redundant paths, if desired)
  - But, even if using all the Sync information, must still detect syncReceiptTimeout for the Sync messages on each path because, if lose a path, need to run BMCA to set up a new redundant path

Feature 6 – Support for Redundant Paths – 7

- Developing the above in detail and adding it to the 802.1ASbt draft requires a fair amount of effort
- Therefore, before developing the above in detail, some initial agreement is needed that this approach is the desired one
  - Or, if this is not the desired approach, what the desired approach is
- In the initial 802.1ASbt draft, an editor's note and/or placeholder for this feature will at least be included

Feature 7 -Alternate Time Scales

- Several approaches for this feature are described in References [3], [4], and [5]
- If a decision on which approach to use is not reached in time for the initial draft of 802.1ASbt, an editor's note and/or placeholder for this feature will at least be included

Feature 8 – Improved Performance

- E.g., improved GM changeover time, longer chains (taken from the PAR [1])
- The improved GM changeover time is addressed by the "Support for Redundant Paths" feature (feature 6 above)
- The support for long chains would be needed by the *Reference Clock* approach for alternate timescales (see [2], [3], and [4])
  - This is because the source (talker) and destination (listener) nodes in this approach might not necessarily be separated by a small number of hops in the synchronization spanning tree (either the tree for the current or hot standby GM)
  - However, before adding any new features to 802.1ASbt, the achievable performance with the current 802.1AS needs to be determined, i.e.,
    - What is the maximum time error, jitter, and wander for a sufficiently large number of hops
    - The "sufficiently large number of hops" pertains to the entire network; in [1], 128 is mentioned for the entire network and 64 for the working clock domain

Feature 8 - Improved Performance

- Initially, new simulations should be performed to determine the performance over a large number of hops (simulate at least 128, probably more for completeness)
- Note that the 802.1AS simulations give only the component of time error due to timestamp granularity, local noise phase noise, local node frequency offset measurement error, and local node frequency stability
- We also need components that account for
  - Uncompensated timestamp error, uncompensated PHY latency, and uncompensated link asymmetry
  - Error in the GM time, if this is important to the applications
  - Time change when changing from current to hot standby GM
  - The above means that a budgeting exercise is necessary

Feature 9 – Enhancements to Determine asCapable

- In P802.1AS-Cor-1, a value of 800 ns for neighborPropDelayThresh for 100BASE-TX and 1000BASE-T links, for full-duplex Ethernet, has been added
- However, for fiber links, i.e., 100BASE-FX and 1000BASE-F, neighborPropDelayThresh is set to the maximum possible value
- For full-duplex IEEE 802.3 fiber links, need information (presentation) on the full range of desired cable lengths
- Need presentations to define the mechanism for other media, i.e., 802.11, 802.3EPON, MoCA, and any new media added by Feature 2; However, note that
  - For EPON, the default value of asCapable is TRUE (802.1AS/13.4)
  - For CSN, for the case of native path delay measurement, asCapable is set to TRUE
  - Therefore, may only need to address 802.11 and CSN for case without CSN network clock reference
- Without the above information, an Editor's note and/or placeholder will be added for this feature to the initial 802.1ASbt draft

Feature 10 – Additional Parameter Sets for Non-A/V Applications

- Need to decide if these parameters should be in 802.1ASbt, or in a new version of 802.1BA
- If the former, would we have multiple sets of default values, for the respective applications?
- In either case, we would eventually need presentations that describe the respective parameter sets for the respective applications
  - Industrial control
  - Automotive
  - Others??
- If there is no immediate decision on whether the multiple parameter sets should go in 802.1ASbt, an editor's note on this will be included

References - 1

[1] IEEE P802.1ASbt PAR, September 10, 2011 (available at http://standards.ieee.org/develop/project/802.1ASbt.html)

[2] Geoffrey M. Garner, *Initial Summary of Work Items for 802.1ASbt (802.1AS Amendment)*, presentation for IEEE 802.1 AVB TG, November 7, 2011.

[3] Don Pannell, *What Makes Talkers & Listeners AVB Compliant?*, presentation for IEEE 802.1 AVB TG, May, 2012.

[4] Franz-Josef Goetz, *Dual Time Scale in Factory & Energy Automation, V1.01, White Paper about Industrial Requirements and Concepts @ Time Synchronization (IEEE 802.1ASbt)*, White Paper submitted to IEEE 802.1 AVB TG, July, 2012.

[5] Franz-Josef Goetz, *Which Mechanism for two Time Scales @ Industry*, presentation for IEEE 802.1 AVB TG, July, 2012.

[6] Lu Huang, *Compensation for Asymmetry of Physical Line*, presentation for IEEE 802.1 AVB TG, March, 2011.

References - 2

[7] Lu Huang, *Management Support for Automatic Measurement of Link Delay Asymmetry*, presentation for IEEE 802.1 AVB TG, November, 2011.

[8] Nigel Bragg and Paul Unbehagen, *Traffic Engineering for 802.1 Unicast/Multicast Networks*, presentation for IEEE 802.1 AVB TG, July, 2012.