Contribution to IEEE 1588 and IEEE 802.1 AVB TG

TITLE: Initial Description of Possible Use of Sync and Followup Messages with Peer-to-Peer Transparent Clocks in AVB

Revision: 1.0

AUTHOR(S) Geoffrey M. Garner (gmgarner@comcast.net)
Samsung (Consultant)
Tel: +1 732 758 0335

DATE: 1 March 2006

1 Introduction

During the February 21, 2006 IEEE 1588/802.1 AVB Design meeting [3], there was initial discussion of a possible way that AVB could use Peer-to-Peer (P2P) Transparent Clocks (TCs) with Sync and Followup messages. It had been stated previously that the use of separate Sync and Followup messages is undesirable with the expected message rates in AVB because the resulting load on the processor would be too large (AVB is expected to use an inexpensive processor such as the 8051). In particular, the AVB processor may take as long as 10 ms to process a Followup message in a TC node. This means that for a path through 7 TCs, it would require on the order of 70 ms for the Followup message to travel from the master to the slave. Note that the Sync messages are time stamped at each TC node as they enter and exit, are not further processed; therefore, a Sync message will travel from the master to slave in much less than 70 ms (the time for the Sync to travel from the master to the slave will likely be on the order of the AVB latency requirement; values in the range of 2 – 6 ms have been discussed). If the master sends Sync every 10 ms, this means that each TC would have to maintain state information on the residence times for multiple Sync messages at any given time (the final TC in the chain would have to save information for as many as 7 Sync messages). As an alternative, TCs could maintain state information for fewer TCs if Followup messages were processed faster. As a second alternative, the Sync interval could be increased to 70 ms or more. The initial approach of having multiple outstanding Sync messages and the two alternatives were considered to be undesirable.

A possible approach for using Sync and Followup messages in AVB was discussed. In this approach, a Sync message is held at a TC until the Followup message arrives. When the Followup message arrives, its correction field is added to the correction field of the Sync, and the Sync is sent. The time the Sync is sent is noted; using this and the time the Sync arrived, a residence time for the Sync in the current TC node is computed. A new Followup message is generated, and the residence time for the current TC is placed in the its correction field.

With this approach, a Sync message will not get ahead of a Followup message for a previous Sync as long as the time between Sync messages is not shorter than the time required for processing a Followup message at a node (i.e., 10 ms). When a Sync message is sent, at least 10 ms will have elapsed since the previous Sync message, which means that the Followup message will have had enough time to be processed at the next downstream node.

Figure 1 illustrates the conventional case where Sync messages are sent frequently (e.g., every 10 ms) and not held, while Followup processing at P2P TC nodes takes on the order of one sync interval (every 10 ms in this example). It is seen that with each successive hop, the Followup message processing delay causes one additional Sync message to get ahead of the Followup message. Figure 2 illustrates the case where the Sync message is held at each successive P2P TC until the associated Followup message arrives. In this case, successive Sync messages do not get ahead of Followup messagews associated with previous Sync's. Note that in both cases the Followup message processing time occurs at each successive P2P TC but not at the BC at the beginning of the chain; it is assumed that the BC can send Followup almost immediately after sending Sync (i.e., in a time short compared to the sync interval). If the Followup processing time were also incurred at the BC, the Synch messages would be held longer at the first P2P TC in Figure 2.

It appears that this approach is fully consistent with the current IEEE 1588 P2P TC approach [1], [2]. The P2P TC Working Technical Description describes how the timestamp and correction fields of Sync and Followup are used so that a slave clock can compute its offset and both slave clocks and P2P TCs can compute path delays (i.e., the message semantics). However, the P2P TCs are free to do the processing in any way that the designer wishes as long as it is consistent with the semantics.

In addition to the above, the P2P TC will use the information contained in Sync and Followup to adjust its frequency. This will likely be done by comparing the amount of elapsed time that the free-running P2P TC oscillator measures between *N* successive Sync messages from the upstream master (*N* is to be determined; for other frequency adjustment schemes involving BCs but not TCs, it was suggested *N* would be on the order of 10; see Item 11 in Section 3 below). Also (see the Author's note in Item 9 of Section 3 below), it was indicated during the February 22 – 24, 2006 IEEE 1588 Face-to-Face Meeting that the P2P TC that issues Delay_Resp to the downstream OC or BC slave and the BC master that sends Sync to the downstream OC or BC must be synchronized in time. This is because the path delay (on the link between the P2P TC and BC or OC slave is computed in the normal manner by averaging the path delays in the two directions. The delay in the master-to-slave direction is obtained using the Sync and Followup messages that originate at the master. The delay in the slave-to-master direction is obtained using the Delay_Resp received from the P2P TC. If the Master and P2P TC are not synchronized in time, an error in the path delay will result. Note that this issue can be avoided in AVB if AVB were to require all BCs and OCs to process the ADelay messages. However, this also is likely not an issue for AVB because end devices that might require time synchronization can, in principle, be present at any AVB node, and therefore the P2P TC nodes would need to be time synchronized anyway. The issue arises in IEEE 1588 because in general 1588 does not require P2P TCs to be time synchronized or, for that matter, syntonized (i.e., frequency synchronized).

The purpose of this document is to provide an initial written description of this approach. The description has been generated based on the verbal discussion in the Design Meeting [3]. It is expected that this written description will be modified after discussion and review.

2 Ordinary or Boundary Clock

1. The master clock (i.e., port in the master state) sends Sync on the respective ports with the followup flag set and the correction field initialized to zero, and measures the time of departure of Sync on each port
2. The master (i.e., port in the master state) sends Followup on each respective port with the correction field initialized to zero and the preciseOrigin timestamp equal to the measured time of departure of the Sync message on that port
3. Each slave (i.e., port in the slave state) measures the time of arrival of the Sync message
4. When the slave (i.e., port in the slave state) receives the Followup message corresponding to a Sync message, it adds the correction fields in the Sync and Followup messages to the preciseOrigin Timestamp in the Followup message). [**Author's Note: TC Working Technical Description Version 12 indicates that the port in the slave state also adds the result of the upstream path delay calculation (Section 1, Items 4a and 4b of [1]). It is not clear which path delay calculation this is referring to. The path delay on the link from the slave port to TC is obtained using the regular Delay_Req/Delay_Resp mechanism, and appears as a separate term in the slave clock offset calculation in IEEE 1588, version 1, clause 7.8.1.1. The regular Delay_Req/Delay_Resp mechanism is described in Section 6 of TC Working Technical Description, Version 12 [1] (referred to as the Default Delay Measurement there). Discussion during the Feburary 22 – 24, 2006 IEEE 1588 Face-to-Face meeting [2] indicated that the equation for the one-way delay in**

Section 6, Item 3 of [1] is incorrect, and that the descriptions in Section 1 (Items 4a and 4b) and Section 6 of [1] need to be consistent with each other and also with IEEE 1588, Version 1.]

3 P2P Transparent Clock

Notes: (1) In AVB, all TCs are Followup TCs (i.e., on-the-fly TCs are assumed not to be used)
(2) Items 6, 8 and 9 below are taken from the current TC Working Technical Description (Version 12; 17 Feb 2006; a portion of item 9 referring to End-to-End (E2E) TCs has been omitted because AVB will not use E2E TCs

1. Measures the time of arrival of a Sync message
2. Holds the Sync message until the corresponding Followup message arrives
3. On arrival of the Followup message, adds the correction field of the Followup Message to the Sync message
4. Sends the Sync message on the respective master ports, and measures its departure time on each port
5. Computes the residence time for the Sync message on each port, and places it in the correction field of a respective new Followup message generated for each port
6. Adds the path delay on the path from which the Sync message came to the correction field of the followup message
7. Sends the new Followup message
8. Performs path delay measurements to peer nodes using the ADelay mechanism.
9. Responds to Delay_Req received from a boundary clock or ordinary clock with Delay_Resp, in the same manner that a boundary or ordinary clock would respond (i.e., the Delay_Resp contains the delay_receipt timestamp for the corresponding received Delay_Req). **[Author's Note: It was indicated during the February 22 – 24, 2006 IEEE 1588 Face-to-Face Meeting that the P2P TC that issues Delay_Resp to the downstream OC or BC slave and the BC master that sends Sync to the downstream OC or BC must be synchronized in time. This is because the path delay (on the link between the P2P TC and BC or OC slave is computed in the normal manner by averaging the path delays in the two directions. The delay in the master-to-slave direction is obtained using the Sync and Followup messages that originate at the master. The delay in the slave-to-master direction is obtained using the Delay_Resp received from the P2P TC. If the Master and P2P TC are not synchronized in time, an error in the path delay will result. Note that this issue can be avoided in AVB if AVB were to require all BCs and OCs to process the ADelay messages. However, this also is likely not an issue for AVB because end devices that might require time synchronization can, in principle, be present at any AVB node, and therefore the P2P TC nodes would need to be time synchronized anyway. The issue arises in IEEE 1588 because in general 1588 does not require P2P TCs to be time synchronized or, for that matter, syntonized (i.e., frequency synchronized).]**
10. Issues Delay_Req to the master port of an attached upstream OC or BC, and uses the Delay_Resp message returned by the master OC or BC to compute the pathe delay on the link to the master. This path delay is added to the correction field of the Followup message that is generated.
11. Uses information in the Sync and Followup messages from the upstream BC or OC master (which is also the Grandmaster) to adjust its frequency. This will likely be done by comparing the amount of elapsed time that the free-running P2P TC oscillator measures between *N* successive Sync messages from the upstream master (*N* is to be determined; for other frequency adjustment schemes involving BCs but not TCs, it was suggested *N* would be on the order of 10).

4 References

[1] *Transparent Clock – Working Technical Description*, Version 12, IEEE 1588 TC Subcommittee, February 17, 2006.

[2] Revisions made to *Transparent Clock – Working Technical Description*, Version 12 at February 22 – 24, 2006 IEEE 1588 Face-to-Face Meeting (see meeting minutes).

[3] IEEE 1588/802.1 AVB Design Meeting, February 21, 2006 (see meeting minutes).

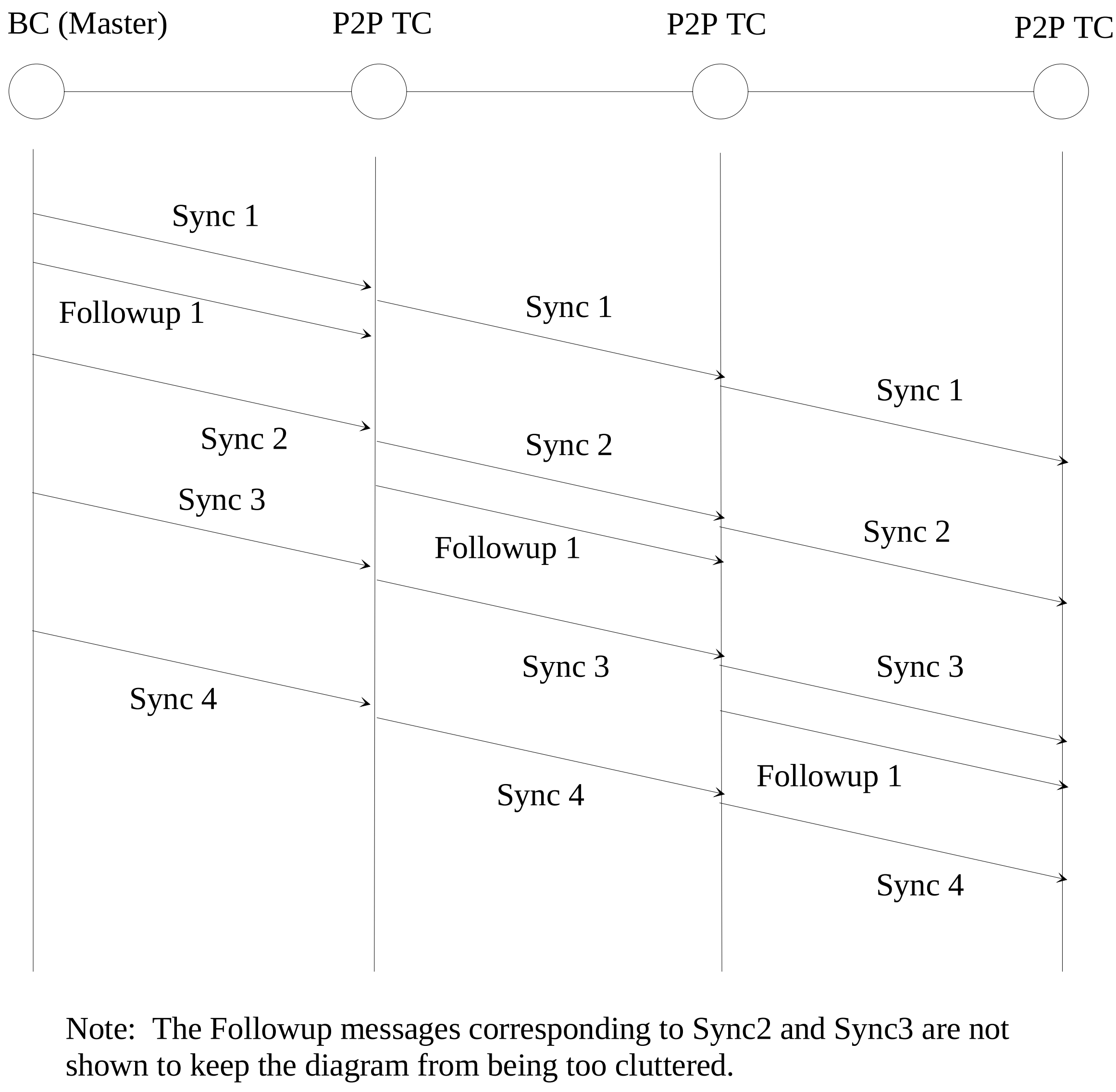

Figure 1. Sync and Followup sent in conventional manner (Sync not held at P2P TC until corresponding Followup arrives). Followup processing time and sync interval are of the same order. Sync processing time is much less than Followup processing time.

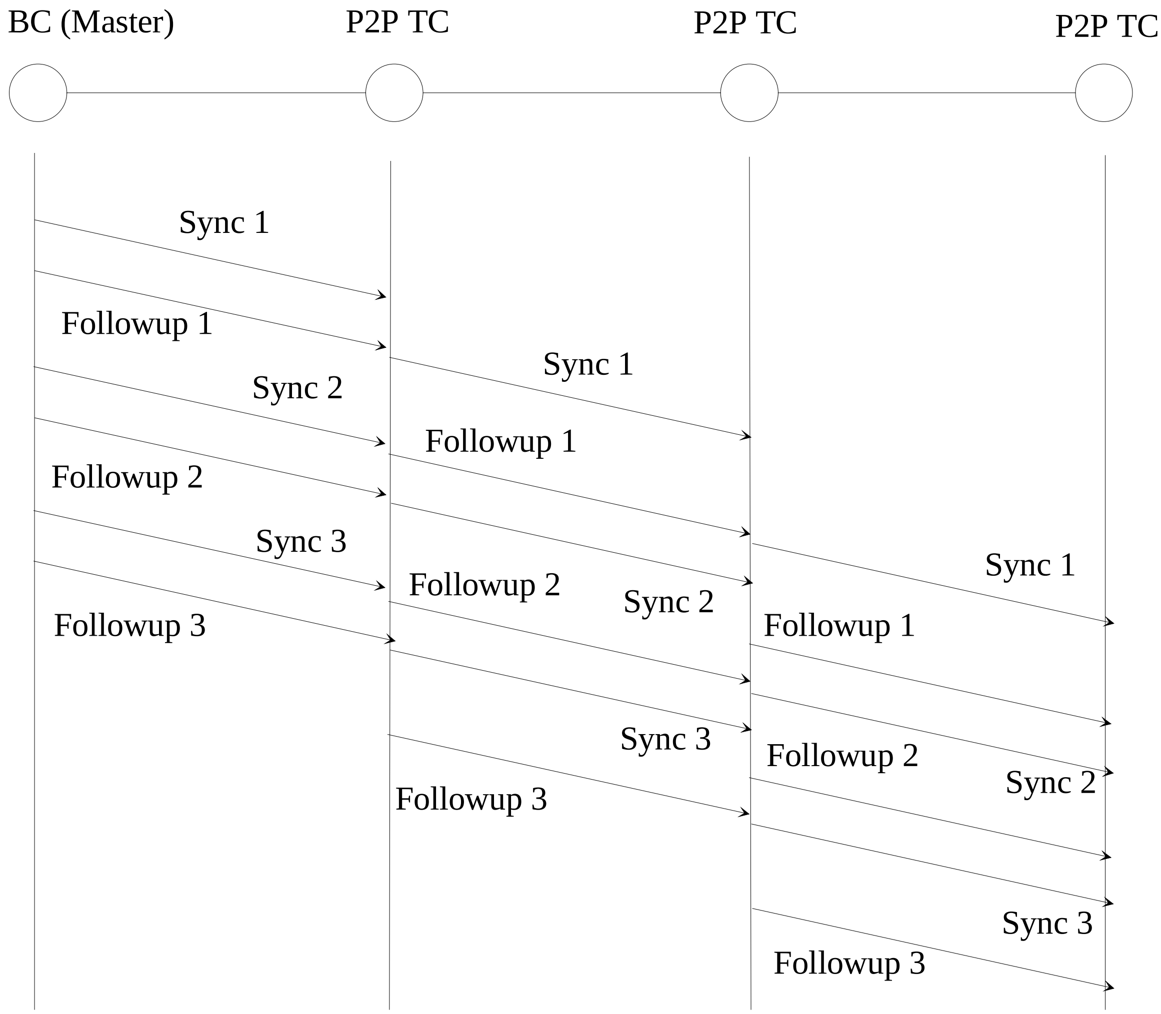

Figure 2. Sync and Followup sent with Sync held at P2P TC until corresponding Followup arrives. Followup processing time and sync interval are of the same order. Sync processing time is much less than Followup processing time.