Summary of New Framework for Time-Aware Systems in IEEE 802.1AS D1.1

Geoffrey M. Garner
SAMSUNG Electronics (Consultant)

IEEE 802.1 AVB TG
2008.01.28

gmgarner@comcast.net

Summary of Major Revisions for 802.1AS D1.1

- ❑Clause 10, *Media-independent layer specification,* is rewritten
- ❑Clause 11, *Layer specification for full-duplex, point-to-point links*, is rewritten
- ❑Subclause 6.4, *Data types and on-the-wire formats in an 802.1AS network*, is revised to describe the creation of PDUs and not explicitly describe on-the-wire formats
  - ▪This attempts to address several TG ballot comments against D1.0
- ❑These VGs focus on the new material in Clauses 10 and 11
- ❑New title for Clause 11 (previous title was specific to IEEE 802.3 full-duplex Ethernet) emphasizes that it is applicable to all full-duplex, point-to-point media
  - ▪IEEE 802.3 full-duplex Ethernet is a very important special case
  - ▪This change in emphasis was made to address a TG ballot comment against D1.0

New Framework for Time-Aware Systems - 1

- ❑D1.0 described boundary/ordinary clocks and peer-to-peer transparent clocks separately
- ❑D1.1 instead describes the various functions present in OCs, BCs, and P2P TCs
  - ▪Clauses 10 and 11 do not explicitly mention BCs, OCs, and P2P TCs
  - ▪Rather, an OC, BC, or P2P TC can be constructed by choosing the appropriate functions described in Clauses 10 and 11 for the time-aware system
- ❑Time-aware system functions
  - ▪ClockMaster – may or may not be present
  - ▪ClockSlave – may or may not be present
  - ▪Best master selection function (bmSelectionPresent) – may or may not be present
  - ▪Best master selection function is centralized in SyncSite entity (previously called PTPSynchronzation entity in D1.0) rather than decentralized in each TS entity (previously called BCPortI entity in D1.0)
    - This change was made to address a TG ballot comment against D1.0

New Framework for Time-Aware Systems - 2

- In addition, network synchronized time is transported by the combination of the time stamp field, correction field, and extraTime field
  - extraTime is a correction for the time error due to the frequency offset of the local free-running oscillator over the period since the previous reference time for the computation of network synchronized time
    - This reference time is generally taken to be a time instant that is between the receipt of the most recent synchronization information and the receipt of the previous most recent synchronization information before that
    - The scheme for computation of network-synchronized time is what has been referred to as backward interpolation
    - If the reference time is taken to be the time of receipt of the most recent synchronization information, then the period since that time (the duration of the period is referred to as backTime) is the residence time of a transparent clock
      - In this case the computation of network-synchronized time reduced to the split-syntonization scheme
    - However, in general this period may be a larger value; some analysis is needed to establish the optimal value
    - Previous analysis of the split syntonization scheme (March, 2007 IEEE 802.1 AVB TG meeting) showed the performance improvement of the split syntonization scheme compared to the case backTime = 0; that analysis suggests that worst-case performance will gradually degrade as backTime is decreased from the residence time to zero
    - However, so far there is no analysis for the case backTime > residence time
    - Note also that, while the worst case performance for backTime = residence time is better than that for backTime = 0, some non-worst cases for backTime = 0 will be better

New Framework for Time-Aware Systems - 3

- Define

$t^r_{i,j} = \text{equivalent local time of receipt of } j^{th} \text{ Sync message at node } i$

$\quad = \text{time of receipt minus measured propagation time on attached link}$

$t^s_{i,j} = \text{time of sending of } j^{th} \text{ Sync message at node } i$

(messages sent and received are numbered (indexed) separately)

$x^r_{i,j} = \text{sum of time stamp and correction field for } j^{th} \text{ Sync and Follow\_Up messages received at node } i$

$e^r_{i,j} = \text{cumulative extraTime field for } j^{th} \text{ Sync and Follow\_Up messages received at node } i$

$x^s_{i,j} = \text{sum of time stamp and correction field for } j^{th} \text{ Sync and Follow\_Up messages sent from node } i$

$e^s_{i,j} = \text{cumulative extraTime field for } j^{th} \text{ Sync and Follow\_Up messages sent from node } i$

- Then

$$t^s_{i,j} = t^r_{i+1,j}$$

$$x^s_{i,j} = x^r_{i,j} + \left(t^s_{i,j} - b - t^r_{i,j}\right)r + b$$

$$e^s_{i,j} = (r-1)b$$

- where

$r = \text{measured rate ratio of local oscillator relative to grandmaster}$

$b = \text{backTime}$

New Framework for Time-Aware Systems - 4

❑Special case leads to split-syntonization scheme discussed previously

$$
\begin{aligned}
&b = t^s_{i,j} - t^r_{i,j} \\
&\text{Then} \\
&x^s_{i,j} = x^r_{i,j} + t^s_{i,j} - t^r_{i,j} = x^r_{i,j} + \text{residence time} \\
&e^s_{i,j} = (r-1)(t^s_{i,j} - t^r_{i,j}) = (r-1)(\text{residence time})
\end{aligned}
$$

❑Note: Setting $b = 0$ leads to traditional syntonization

$$
\begin{aligned}
&t^s_{i,j} = t^r_{i+1,j} \\
&x^s_{i,j} = x^r_{i,j} + \left(t^s_{i,j} - t^r_{i,j}\right)r \\
&e^s_{i,j} = 0
\end{aligned}
$$

New Framework for Time-Aware Systems - 5

❑ D1.1 allows the portion of the network synchronized time that excludes extraTime to be split among the time stamp field and correction field however the designer wishes

- Setting time stamp equal to time that upstream Sync message left closest upstream time-aware system that has a best master selection function present corresponds to P2P TC
- Setting time stamp equal to computed network-synchronized time, excluding contribution of extraTime and fractional ns (fractional ns must be carried in correction field) corresponds to OC/BC

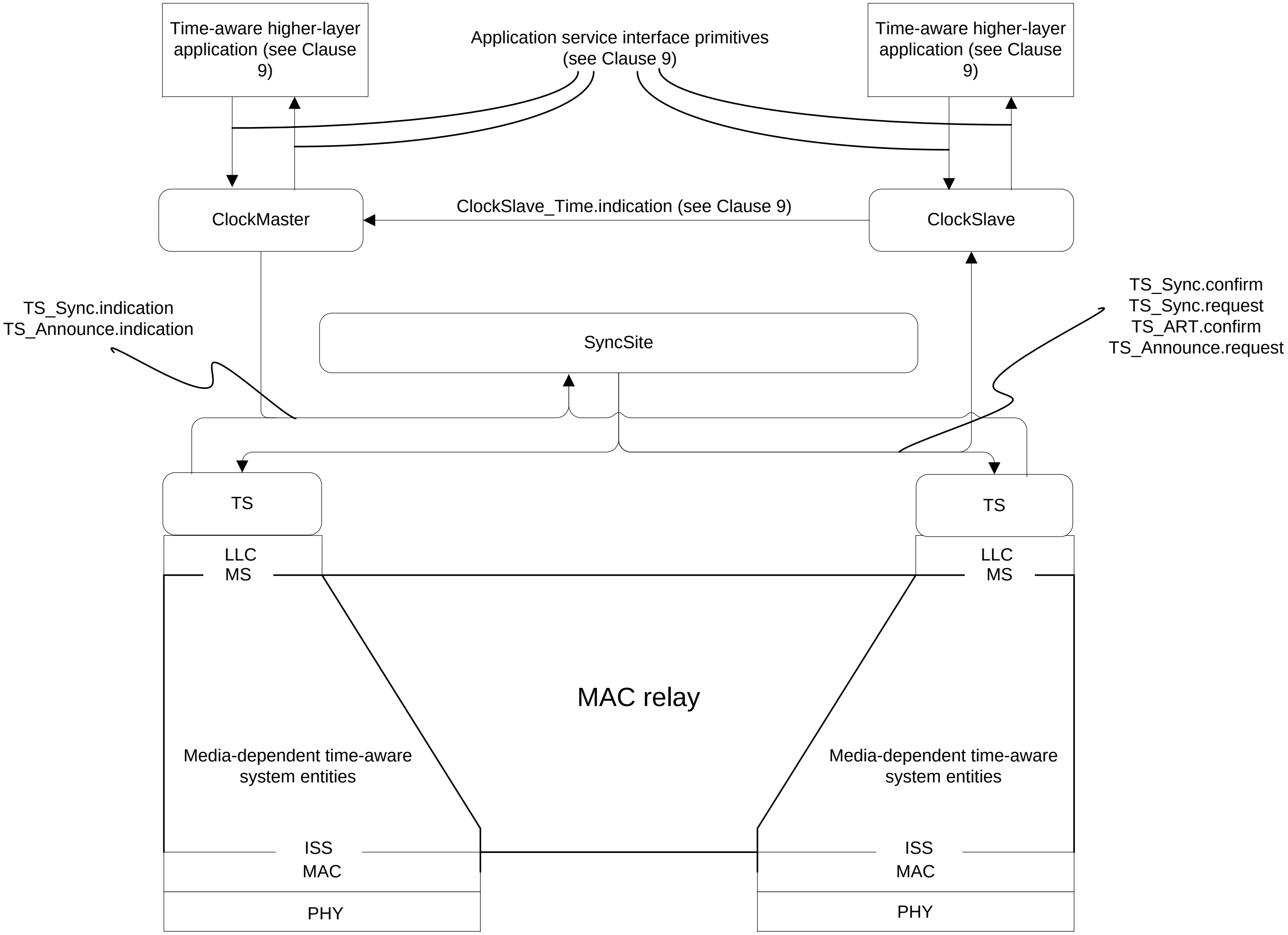
Possibilities for Time-Aware System - 1

Possibility	ClockMaster	ClockSlave	Best Master Selection	802.1ASD1.0/1588 terminology
1	Present	Present	Present	OC (1 port)/BC (>1 port)
2	Not present	Present	Present	Slave only OC (this case limited to 1 port)
3	Present	Not present	Present	No equivalence
4	Not present	Not present	Present	No equivalence
5	Present	Present	Not present	Note 1
6	Not present	Present	Not present	P2P TC with collocated slave-only OC
7	Present	Not present	Not present	Note 2
8	Not present	Not present	Not present	P2P TC

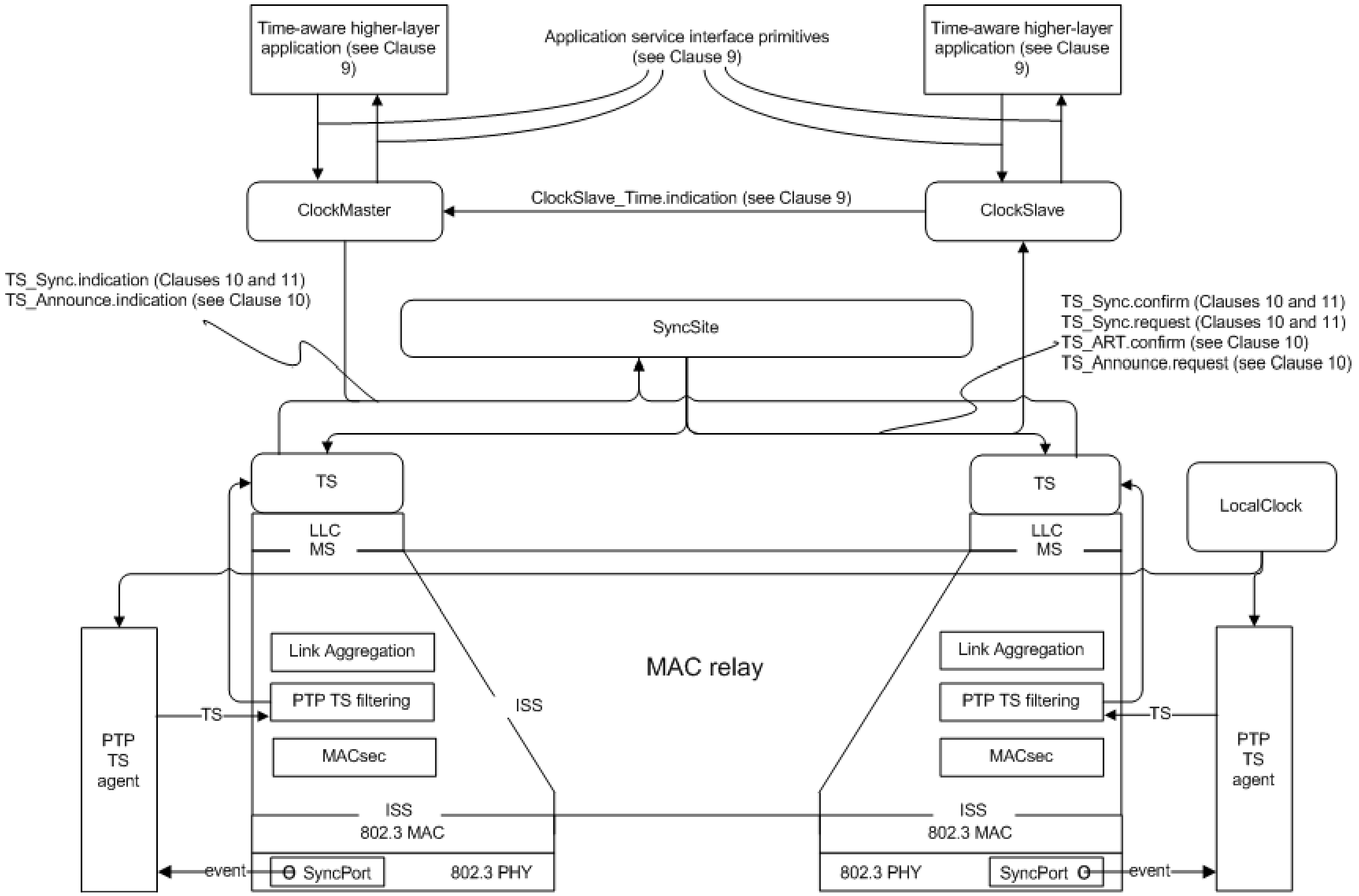
Possibilities for Time-Aware System - 2

- Note 1 – 802.1AS D1.0/1588 allow a P2P TC with collocated OC
  - This would be equivalent in D1.1 to merging in one box a multiport time-aware system with no ClockMaster, ClockSlave, or best master selection function and another time-aware system with one port and ClockMaster, ClockSlave, and best master selection functions present
    - In this system, Announce messages would be reflected by former time-aware system; latter time aware system would determine best clock and also send Announce messages
    - Seemed more efficient to simply assume best master selection operates for entire time-aware system, as in possibility 1
- Note 2 – This case is not present in 802.1AS D1.0/1588
  - In 802.1AS D1.1, a best master selection function must be present in order to use ClockMaster information; if a best master selection function is not present, information from any ClockMaster is ignored
    - This was necessary in order that state machines would behave sensibly

Media-Independent Layering (Clause 10)


Time-aware higher-layer application (see Clause 9)
Application service interface primitives (see Clause 9)
Time-aware higher-layer application (see Clause 9)
ClockMaster
ClockSlave_Time.indication (see Clause 9)
ClockSlave
TS_Sync.indication
TS_Announce.indication
SyncSite
TS_Sync.confirm
TS_Sync.request
TS_ART.confirm
TS_Announce.request
TS
TS
LLC
MS
LLC
MS
MAC relay
Media-dependent time-aware system entities
Media-dependent time-aware system entities
ISS
MAC
ISS
MAC
PHY
PHY

Layering for Time-Aware System with Full-Duplex, Point-to-Point Links (Clause 11)


Time-aware higher-layer application (see Clause 9)
Application service interface primitives (see Clause 9)
Time-aware higher-layer application (see Clause 9)
ClockMaster
ClockSlave_Time.indication (see Clause 9)
ClockSlave
TS_Sync.indication (Clauses 10 and 11)
TS_Announce.indication (see Clause 10)
SyncSite
TS_Sync.confirm (Clauses 10 and 11)
TS_Sync.request (Clauses 10 and 11)
TS_ART.confirm (see Clause 10)
TS_Announce.request (see Clause 10)
TS
TS
LocalClock
LLC
MS
LLC
MS
PTP TS agent
Link Aggregation
PTP TS filtering
MACsec
MAC relay
ISS
Link Aggregation
PTP TS filtering
MACsec
PTP TS agent
TS
TS
ISS
802.3 MAC
ISS
802.3 MAC
event
SyncPort
802.3 PHY
802.3 PHY
SyncPort
event

802.1AS D1.1 Issues that Needed Consideration - 1

- Since an arbitrary time-aware system may or may not have a ClockMaster, ClockSlave, and best master selection function present, the SyncSite and TS state machines must handle all combinations of these cases (except those that might be disallowed)
- Examples
  - Announce messages arrive, and information is sent to SyncSite, regardless of whether best master selection function is present
    - SyncSite must handle the Announce information properly, and TS entities that SyncSite sends the information to must handle it properly
  - sync_receipt_timeout must be handled properly whether or not a best master selection function is present
  - SyncSite must handle time synchronization information and potential best clock information it receives from ClockMaster, if one is present, whether or not best master selection function is present

802.1AS D1.1 Issues that Needed Consideration - 2

- ❑State machines for SyncSite and TS entities need to behave sensibly in all cases
  - ▪If best master selection function is present, want ports to change state when SyncSite sends new best clock information to TS entities after executing BMCA
    - BMCA is executed whenever new potential best master information is received by SyncSite via TS_Announce.indication primitive (either when Announce message is received or ClockMaster sends the primitive)
  - ▪If best master selection function is not present, the time-aware system pays attention to whatever port is currently receiving Sync messages (i.e., is in the SLAVE state); if stop receiving Sync messages, all ports go to LISTENING state and then first port to receive Sync message goes to slave state (and other ports go to MASTER state)
    - In D1.0, this was the P2P TC behavior; was separate from BMCA in OC/BC
    - In D1.1, where part of time-aware system includes whether best master selection function is present or not, this behavior occurs whether or not best master selection is present; it corresponds to sync_receipt_timeout. If best master selection function is present, then may get subsequent port state changes when next Announce is received.

802.1AS D1.1 Issues that Needed Consideration - 2

- In D1.1, a Sync message is sent on MASTER ports in response to having received a Sync message on SLAVE port
  - But, this is subject to constraint that time between successive Sync messages is not less than 50% of Sync interval, nor more than 125% of Sync interval
  - To achieve this behavior, a TS_Sync.request primitive is sent from SyncSite to TS entities whenever Sync is received provided the time since last TS_Sync.request was not less than 0.5*mean_sync_interval
    - In addition, send TS_Sync.request if time since last TS_Sync.request is 1.25*mean_sync_interval, even if have not received TS_Sync.indication from SLAVE port

Issues for 802.11 Links

❑Currently, 802.11 initiates MLME_PRESENCE_REQUEST.request from slave end

- This means that when SyncSite at bridge that contains wireless AP sends TS_Sync.request to TS entities, the TS entity for the 802.11 links will ignore it
- Instead, requests will come separately from wireless stations
- Need to decide if this behavior is desirable (i.e., should master initiate request when it receives the TS_Sync.request
  - Note that it would not send Sync message; the TS_Sync.request would initiate the sending of synchronization information via the presence request/response mechanism, but initiated from master end

❑Currently, 802.11 does not have field for accumulatedExtraTime

- This would be needed if we want to use interpolation on 802.11 links