

Joint IEEE-SA and ITU Workshop on Ethernet

IEEE 802.1 AS
gPTP - One Step Issues

Franz-Josef Goetz,
Member of IEEE 802.1 TSN TG,
Siemens AG

Draft PAR (P802.1 AS br) title & scope

■ Title

Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks Amendment: Enhancements and performance improvements

■ Scope(1)

➤ *The enhancements that are to be considered by this project include:*

- Support for link aggregation (IEEE 802.1AX, IEEE 802.1AXbk, and 802.1AXbq)
- Support for new media types, with corresponding media-dependent layers, e.g., IEEE Std 1901 and WiFi Direct
- Interoperability with one-step clocks on receive (but with no requirement to generate one-step Sync messages)

Draft PAR (P802.1 AS br) title & scope

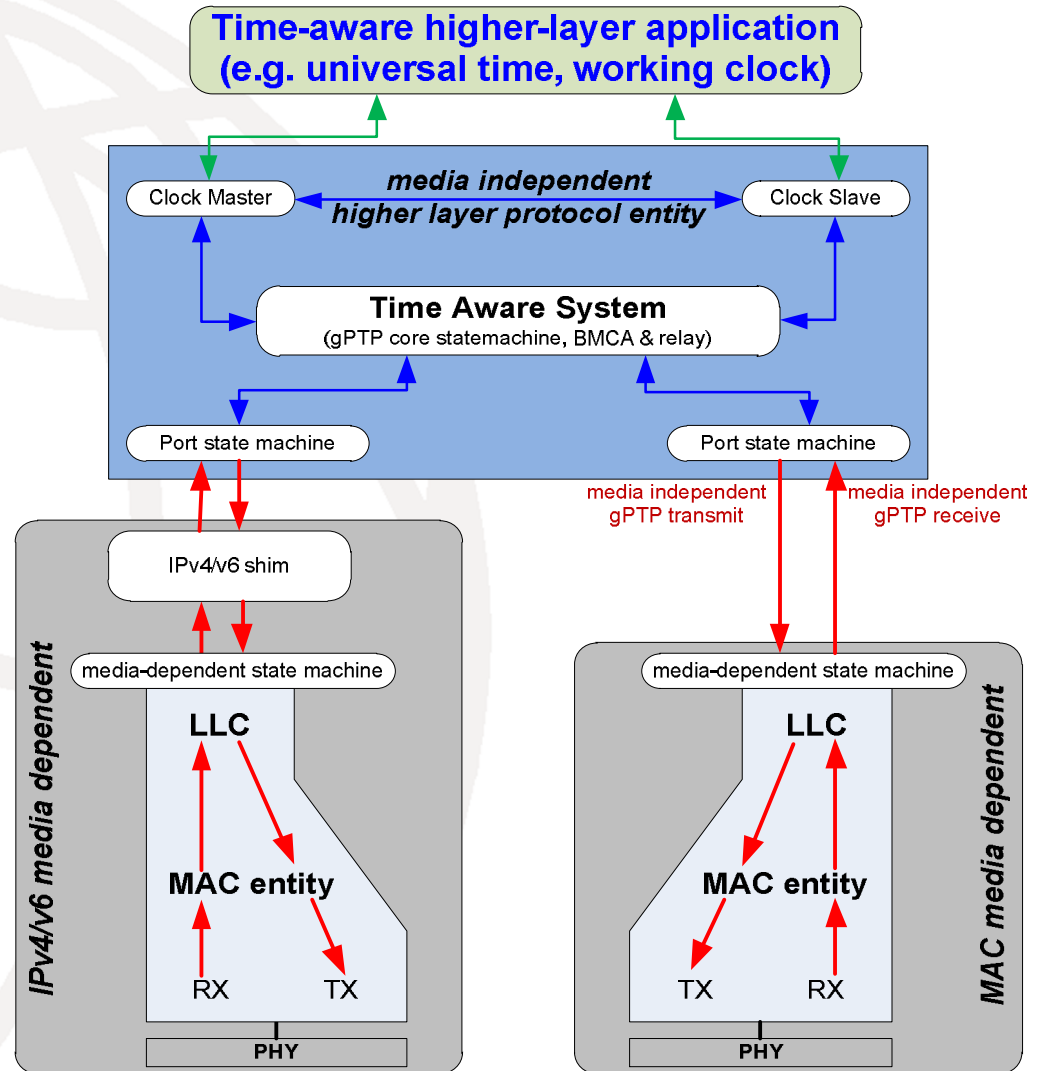
■ Scope(2)

- Support for redundant paths
- Enhancements to the determination of asCapable (e.g., longer cable lengths, new media types)
- Incorporation of the interfaces specified in IEEE Std 802.3bf into the IEEE 802.3 full-duplex media-dependent layer model
- Improved performance
- Carrying information on alternate time scales (e.g., local time for a respective time zone)
- Automatic measurement of link delay asymmetry
- Additional parameter sets for non-Audio/Video applications, e.g., industrial control

gPTP – Architecture Model

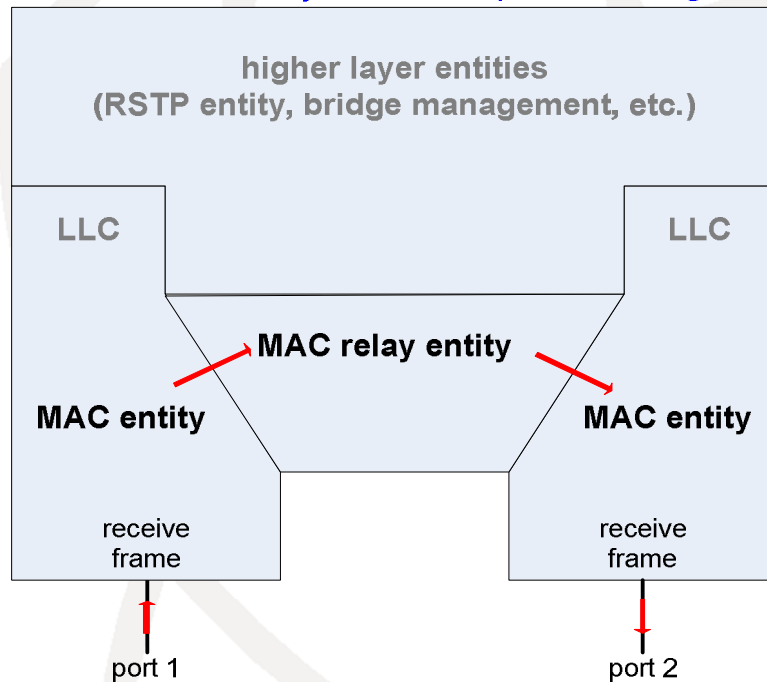
gPTP has introduced a layering model for IEEE 1588

- time application
(universal time, working clock)
- media independent entity
(BMCA algorithm, sync relay, ...)
- media dependent entity
 - time stamping
 - media (wired, wireless, ...)
 - transport (L2 MAC, IPv4, IPv6)



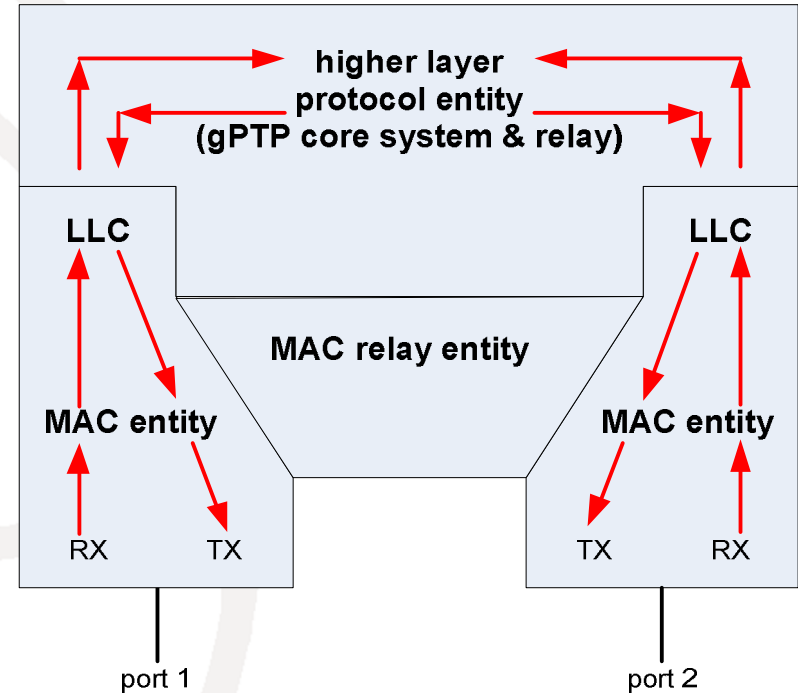
L2 Forwarding PTP/gPTP

PTP Transparent Clock (TC) Baggy pants Announce, Sync, FollowUp forwarding



- Announce, Sync and FollowUp are Multicast/Unicast messages
- Sync path depends on RSTP entity

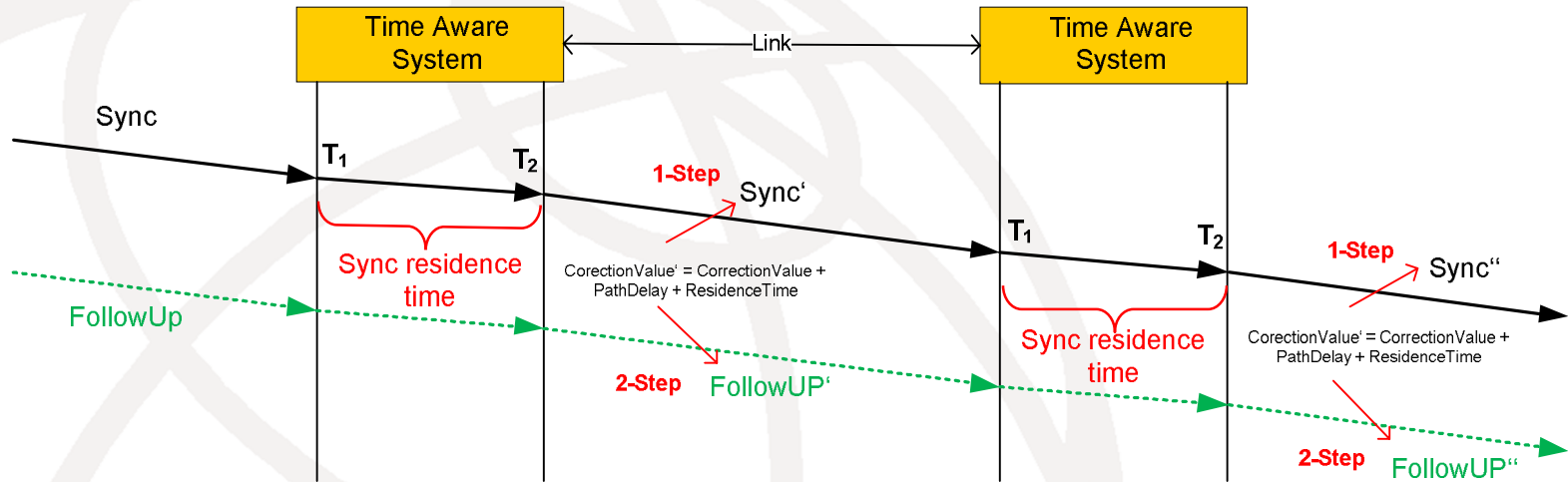
gPTP Time Aware System Baggy pants Announce, Sync, FollowUp forwarding



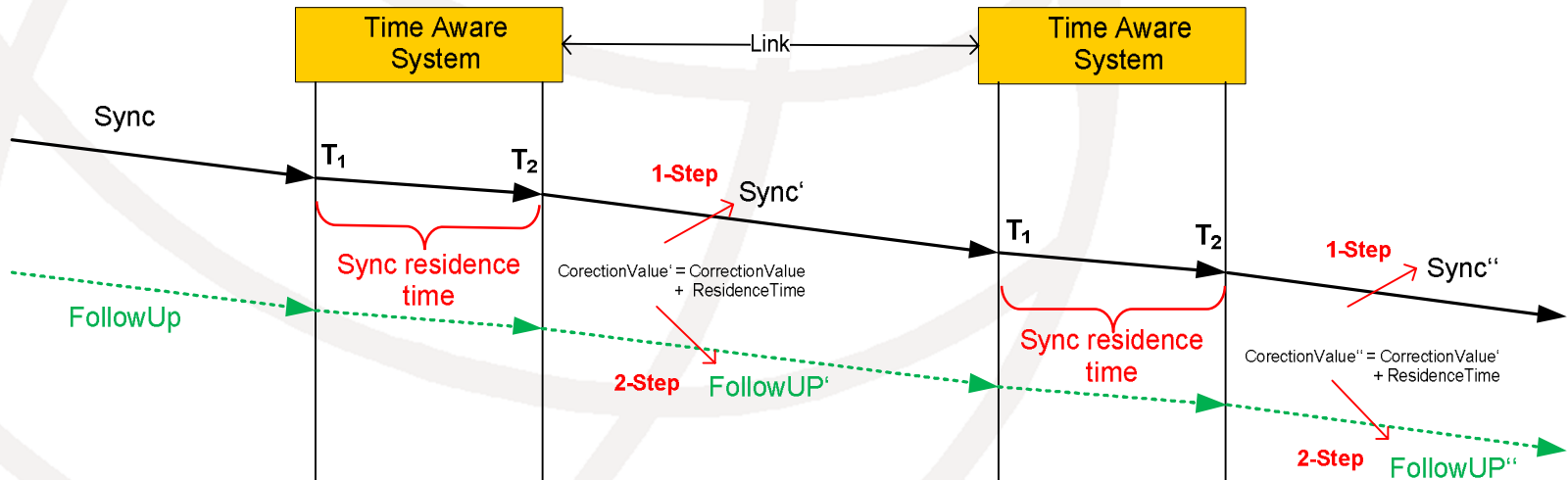
- Announce, Sync and FollowUp are peer-to-peer messages
- BMC algorithm is used to establish sync path

One & Two –Step

Synchronization with peer-to-peer mechanism



Synchronization with end-to-end mechanism



Peer-To-Peer PathDelay

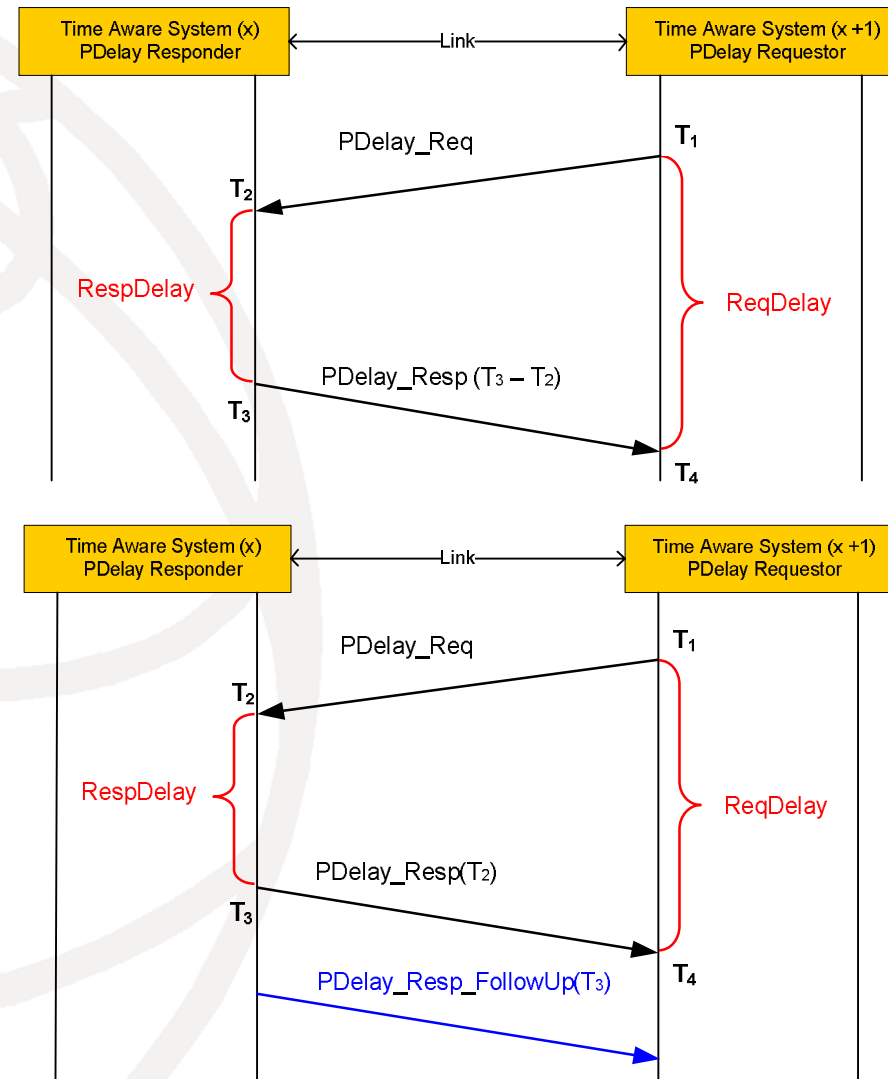
PDelay_Req, PDelay_Resp and PDelay_Resp_FollowUp messages are all peer-to-peer messages

➤ 1-Step PathDelay measurement

(PTP 1-Step path delay measurement do not carry T_2 back to the PDelay requester, for a accurate path delay measurement neighbor frequency offset is important)

=> another reason why gPTP use 2-Step Path Delay measurement

➤ 2-Step PathDealy measurement



Two Step for IEEE 802.1 AS

■ Reasons

- Measurements can be extremely accurate, since they are based on actual transmission/receipt time
- Packet must be detected, but not modified at a very low layer to get the actual transmission/receipt time
- No checksum is recomputed
- Protocol can be fully protected by 802.1AE MAC security
- 2-Step mechanism can achieve the same accuracy with hardware support (is possible)

■ But – Two Step without hardware support

(Sync processing needs a lot of computing power)

- What about multiples sync messages?
(two time scales, redundant path, one or two hot-standby grandmaster)
- What about higher sync rates?
- What about bridges with a huge number of port?

Two Step Sync Processing

■ Problem

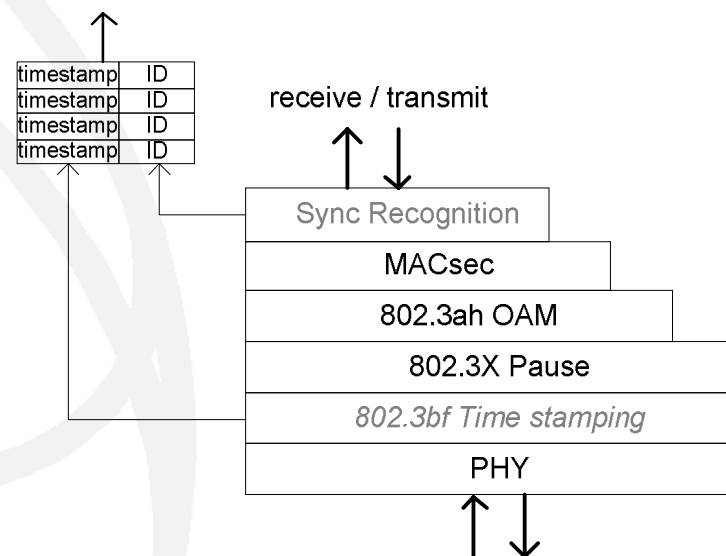
- How to match sync event messages to timestamps?

■ Solution

- Sync recognition above the MACsec (ID from the packet to record with timestamp)
- Time stamping at lowest level to get maximum accuracy

■ Method

- Every received PTP event messages carries timestamp and ID up
- Every transmitted PTP event messages carries timestamp and ID up



One Step Sync Processing

■ Proposal

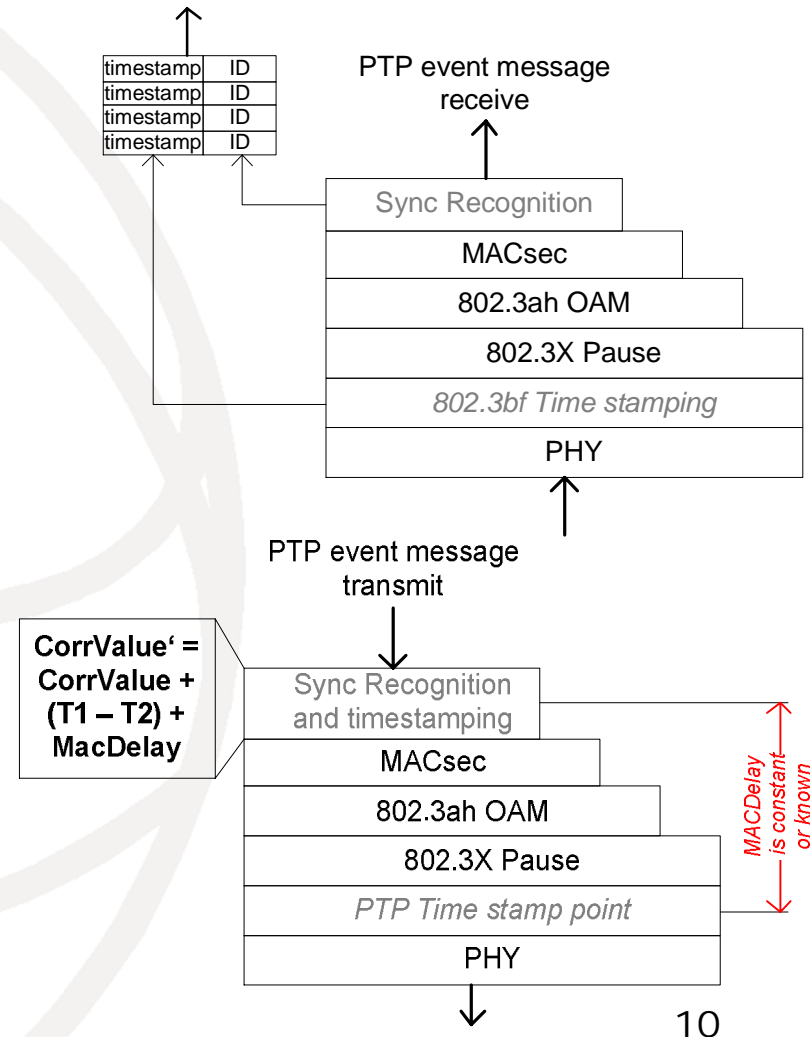
- Receive: time stamping at PHY
- Transmit: time stamping below MACsec

■ Disadvantage

- Putting MACsec below time stamping can sacrifice accuracy
- Guaranteed or known MACDelay

■ To discuss:

- Is MACsec for PTP required?
- Is this a solution also for synchronization over IPv4/v6?



*THANK YOU
for your attention*

Questions?