Document Title	Time Synchronization Protocol Specification
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	897

Document Status	Final
Part of AUTOSAR Standard	Foundation
Part of Standard Release	1.5.1

Document Change History

Date	Release	Changed by	Description
2019-03-29	1.5.1	AUTOSAR Release Management	• Minor changes
2018-10-31	1.5.0	AUTOSAR Release Management	• Initial release

Disclaimer

Table of Contents

1 Introduction and overview

This protocol specification specifies the format, message sequences and semantics of the AUTOSAR Time synchronization Protocol.

The Time synchronization Protocol handles the distribution of time information over Ethernet. The Ethernet mechanism is based on existing PTP (Precision Time Protocol) mechanisms that are described in standards like IEEE1588 (IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems) and IEEE802.1AS (Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks). IEEE802.1AS, also known as gPTP (generalized Precision Time Protocol), can be seen as a profile (or subset) for using IEEE1588. However, neither IEEE1588 nor IEEE802.1AS have been developed considering automotive requirements. Therefore, the Time Synchronization over Ethernet uses the current mechanisms as defined in IEEE802.1AS with specific extensions and/or restrictions. Automotive Ethernet networks deviate from commercial Ethernet networks in terms of the following items:

- Role and functions of ECUs is known and defined a priori
- The network is static, i.e. components like ECUs, switches and characteristics like cable length, dont change during operation or even after switching off and switching on the vehicle. Components of course may be unavailable (due to failure situations or by purpose) but mostly only change when the vehicle is at a service facility.

Therefore, dynamic mechanisms like determining the Global Time Master (denoted as grandmaster in IEEE802.1AS) by the best master clock algorithm (BMCA) during operation are not required. It is also possible to omit the cyclic measurement of link delays on Ethernet links due to the static nature of the automotive network and restrict mechanisms that belonging to dynamic network topology.

1.1 Protocol purpose and objectives

The Time synchronization protocol is used to

- synchronize time bases and the corresponding Ethernet messages
- measure time differences between Ethernet frames

1.2 Applicability of the protocol

The concept is targeted at supporting time-critical and safety-related automotive applications such as airbag systems and braking systems. This doesn't mean that the concept has all that is required by such systems though, but crucial timing-related features that cannot be deferred to implementation are considered.

1.2.1 Constraints and assumptions

This document specifies the PRS_TimeSync protocol. It was created during elaboration of the AUTOSAR Foundation Standard 1.5.0 which took place in parallel to the development of the AUTOSAR Classic Standard 4.4.0. It already reflects all changes implied to TimeSyncOverEthernet by the work which was done for AUTOSAR Classic Platform.

1.2.2 Limitations

1. No support of BMCA protocol, like specified in [1, IEEE 802.1 AS]
2. No support of Announce and Signaling messages, like specified in [1, IEEE 802.1 AS].
3. The reception of a Pdelay_Req is not taken as a pre-condition to start with the transmission of Sync messages.
4. The Rate Correction will be performed by the Time synchronization protocol, which does not require the Pdelay mechanism. For some applications, e.g. for Audio/Video, it might be necessary to use Pdelay based Rate Correction performed by Time synchronization protocol itself, which is optional and not considered by this specification.
5. Because of (4), the Time synchronization protocol will not maintain the Ethernet HW clock but may use it as a source for the Virtual Local Time.
6. While IEEE 802.1AS states, that IEEE 802.1AS message shall not have a VLAN tag nor a priority tag, the Time synchronization protocol would allow Time Synchronization on VLANs under the condition, that the switch HW supports forwarding of reserved multicast addresses using the range of 01:80:C2:00:00:00 .. 0F

1.2.3 Accuracy

Time Master and Time Slave shall work with a Time Base reference clock accuracy as defined in [1, IEEE 802.1 AS], ANNEX B.1.2 "Time measurement granularity".

1.3 Dependencies

1.3.1 Dependencies to other protocol layers

There are no dependencies to other protocols.

1.3.2 Dependencies to other standards and norms

The AUTOSAR Time Synchronization protocol is derived from [1, IEEE 802.1 AS]. For VLAN characteristics refer to [2, IEEE 802.1Q-2011].

1.3.3 Dependencies to the Application Layer

There are no dependencies to the application layer.

2 Use Cases

ID	*Name*	*Description*
0001	Pdelay measurement	Measuring of delays between Ethernet messages
0002	Time Synchronization	Time synchronization of different time bases.

3 Protocol Requirements

3.1 Requirements Traceability

Requirement	Description	Satisfied by
[RS_TS_20047]	The Timesync over Ethernet module shall trigger Time Base Synchronization transmission	[PRS_TS_00016] [PRS_TS_00050] [PRS_TS_00186]
[RS_TS_20048]	The Timesync over Ethernet module shall support IEEE 802.1AS as well as AUTOSAR extensions	[PRS_TS_00002] [PRS_TS_00003] [PRS_TS_00004] [PRS_TS_00005] [PRS_TS_00011] [PRS_TS_00012] [PRS_TS_00016] [PRS_TS_00018] [PRS_TS_00023] [PRS_TS_00025] [PRS_TS_00028] [PRS_TS_00050] [PRS_TS_00053] [PRS_TS_00054] [PRS_TS_00055] [PRS_TS_00056] [PRS_TS_00057] [PRS_TS_00058] [PRS_TS_00059] [PRS_TS_00060] [PRS_TS_00061] [PRS_TS_00062] [PRS_TS_00063] [PRS_TS_00066]

		[PRS_TS_00067] [PRS_TS_00068] [PRS_TS_00069] [PRS_TS_00070] [PRS_TS_00071] [PRS_TS_00072] [PRS_TS_00075] [PRS_TS_00077] [PRS_TS_00079] [PRS_TS_00086] [PRS_TS_00141] [PRS_TS_00142] [PRS_TS_00149] [PRS_TS_00154] [PRS_TS_00163] [PRS_TS_00164] [PRS_TS_00166] [PRS_TS_00167] [PRS_TS_00168] [PRS_TS_00169] [PRS_TS_00170] [PRS_TS_00171] [PRS_TS_00181] [PRS_TS_00206] [PRS_TS_00207] [PRS_TS_00208] [PRS_TS_00209] [PRS_TS_00210]
[RS_TS_20051]	The Timesync over Ethernet module shall detect and handle errors in synchronization protocol / communication	[PRS_TS_00004] [PRS_TS_00025] [PRS_TS_00164] [PRS_TS_00210]
[RS_TS_20052]	The configuration of the Time Synchronization over Ethernet module shall allow the module to work as a Time Master	[PRS_TS_00094] [PRS_TS_00207] [PRS_TS_00208] [PRS_TS_00209]
[RS_TS_20053]	The configuration of the Time Synchronization over Ethernet module shall allow the module to work as a Time Slave	[PRS_TS_00156]
[RS_TS_20054]	The Implementation of the Time Synchronization shall evaluate and propagate Time Gateway relevant information	[PRS_TS_00094] [PRS_TS_00156]
[RS_TS_20059]	The Timesync over Ethernet module shall access all communication ports belonging to Time Synchronization	[PRS_TS_00053] [PRS_TS_00054] [PRS_TS_00055] [PRS_TS_00056] [PRS_TS_00057] [PRS_TS_00058] [PRS_TS_00059] [PRS_TS_00060] [PRS_TS_00166] [PRS_TS_00167] [PRS_TS_00168] [PRS_TS_00169] [PRS_TS_00170] [PRS_TS_00171] [PRS_TS_00207] [PRS_TS_00208] [PRS_TS_00209]
[RS_TS_20061]	The Timesync over Ethernet module shall support means to protect the Time Synchronization protocol	[PRS_TS_000104] [PRS_TS_00028] [PRS_TS_00062] [PRS_TS_00063] [PRS_TS_00066] [PRS_TS_00067] [PRS_TS_00068] [PRS_TS_00069] [PRS_TS_00070] [PRS_TS_00071] [PRS_TS_00072] [PRS_TS_00074] [PRS_TS_00075] [PRS_TS_00076] [PRS_TS_00077] [PRS_TS_00078] [PRS_TS_00079] [PRS_TS_00081] [PRS_TS_00082] [PRS_TS_00084] [PRS_TS_00085] [PRS_TS_00086] [PRS_TS_00088] [PRS_TS_00089]

		[PRS_TS_00091] [PRS_TS_00092] [PRS_TS_00093] [PRS_TS_00097] [PRS_TS_00098] [PRS_TS_00099] [PRS_TS_00100] [PRS_TS_00101] [PRS_TS_00102] [PRS_TS_00103] [PRS_TS_00105] [PRS_TS_00106] [PRS_TS_00107] [PRS_TS_00108] [PRS_TS_00109] [PRS_TS_00112] [PRS_TS_00113] [PRS_TS_00114] [PRS_TS_00115] [PRS_TS_00116] [PRS_TS_00117] [PRS_TS_00118] [PRS_TS_00119] [PRS_TS_00120] [PRS_TS_00157] [PRS_TS_00181] [PRS_TS_00182] [PRS_TS_00183] [PRS_TS_00184] [PRS_TS_00185]
[RS_TS_20062]	The Timesync over Ethernet module shall support user specific data within the time measurement and synchronization protocol	[PRS_TS_000104] [PRS_TS_00028] [PRS_TS_00062] [PRS_TS_00063] [PRS_TS_00066] [PRS_TS_00067] [PRS_TS_00068] [PRS_TS_00069] [PRS_TS_00070] [PRS_TS_00071] [PRS_TS_00072] [PRS_TS_00074] [PRS_TS_00075] [PRS_TS_00076] [PRS_TS_00077] [PRS_TS_00078] [PRS_TS_00079] [PRS_TS_00081] [PRS_TS_00082] [PRS_TS_00084] [PRS_TS_00085] [PRS_TS_00086] [PRS_TS_00088] [PRS_TS_00089] [PRS_TS_00092] [PRS_TS_00103] [PRS_TS_00105] [PRS_TS_00106] [PRS_TS_00118] [PRS_TS_00119] [PRS_TS_00120] [PRS_TS_00181]
[RS_TS_20063]	The Timesync over Ethernet module shall use the Time Synchronization protocol for Synchronized Time Bases to transmit and receive Offset Time Bases	[PRS_TS_000104] [PRS_TS_00092] [PRS_TS_00095] [PRS_TS_00103] [PRS_TS_00105] [PRS_TS_00106] [PRS_TS_00110] [PRS_TS_00117] [PRS_TS_00118] [PRS_TS_00119] [PRS_TS_00120]
[RS_TS_20066]	The Timesync over Ethernet module shall support a static (pre)configuration of IEEE 802.1AS Pdelay	[PRS_TS_00003] [PRS_TS_00011] [PRS_TS_00012] [PRS_TS_00140] [PRS_TS_00141] [PRS_TS_00142] [PRS_TS_00143] [PRS_TS_00149]

4 Definition of terms and acronyms

4.1 Acronyms and abbreviations

Abbreviation / Acronym:	Description:
(G)TD	(Global) Time Domain
(G)TM	(Global)Time Master
<Bus>TSyn	A bus specific Time Synchronization module

AVB	Audio Video Bridging
BMCA	Best Master Clock Algorithm
CID	Company ID (IEEE)
CRC	Cyclic Redundancy Checksum
Debounce Time	Minimum gap between two Tx messages with the same PDU.
ETH	Ethernet
EthTSyn	Time Synchronization Provider module for Ethernet
Follow_Up	Time transport message (Follow-Up)
GM(C)	Grand Master (Clock)
OFS	Offset synchronization
Pdelay	Propagation / path delay as given in IEEE 802.1AS
Pdelay_Req	Propagation / path delay request message
Pdelay_Resp	Propagation / path delay response message
Pdelay_Resp_Follow_Up	Propagation / path delay Follow-Up message
PDU	Protocol Data Unit
PTP	Precision Time Protocol
StbM	(Global) Time Domain
Timesync	Time Synchronization
Sync	Time synchronization message (Sync)
TG	Time Gateway
TLV	Type, Length, Value field (acc. to IEEE 802.1AS)
TS	Time Slave
TSD	Time Sub-domain
VLAN	Virtual Local Area Network

5 Protocol specification

5.1 General

[PRS_TS_00002] ⌈ The Time Master and Time Slave shall use the default configuration values as defined by [1, IEEE 802.1 AS] (e.g. MAC destination address or Ethernet frame type), if not otherwise specified within this specification.⌋*(RS_TS_20048)*

[PRS_TS_00005] ⌈ The Time Master and Time Slave shall start their protocol state machines without Announce message recognition. ⌋*(RS_TS_20048)*

[PRS_TS_00206] ⌈ All messages belonging to the IEEE Rapid Spanning Tree Protocol (`PortAnnounceReceive`, `PortAnnounceInformation`, `PortRoleSelection`, `PortAnnounceTransmit`) shall be ignored on the receiver side. ⌋*(RS_TS_20048)*

Note: AUTOSAR implementations shall not send those messages.

5.2 VLAN Support

[PRS_TS_00163] ⌈ If FramePrio exists, a frame format with priority and VLAN tags shall be used. Otherwise a frame format without priority and VLAN tags shall be used. ⌋*(RS_TS_20048)*

5.3 Message format

Some message extensions to the [1, IEEE 802.1 AS] are required. This is accomplished by a new AUTOSAR specific *TLV*, which is using a new IEEE CID (0x1A75FB) belonging to AUTOSAR only. An IEEE 802.1AS *TLV* is only available for the message-type Announce (not considered by this specification) and Follow_Up (extended by this specification). The organizationId of the new *TLV* identifies the AUTOSAR *TLV*, which is succeeding the IEEE 802.1AS *TLV*.

The AUTOSAR *TLV* contains *Sub-TLVs* which always consist of a Type, a Length and a data area.

The usage of the *CRC* is optional. To ensure a great variability between several time observing units, the configuration decides of how to handle the *CRC* of a secured *Sub-TLV*. If the receiver does not support the *CRC* calculation, it might be possible, that a receiver just uses the given values, without evaluating the *CRC* itself.

If the *CRC* option is used, one side effect must be considered. Due to the fact, that Pdelay messages do not contain any *TLV*, a *CRC* protection of the related timestamps is not possible. If applications using a *CRC* for Follow_Up together with a non-static Pdelay, unprotected Pdelay time values have to be mixed with protected Follow_Up time values, while calculating the value of the corresponding Time Base.

[PRS_TS_00028] ⌈ The message format, etc. shall be derived from [1, IEEE 802.1 AS] chapter 10. Media-independent layer specification and chapter 11. Media-dependent layer specification for full-duplex, point-to-point links, if not otherwise specified. ⌋ *(RS_TS_20048, RS_TS_20061, RS_TS_20062)*

[PRS_TS_00181] ⌈ The byte order for multibyte values is Big Endian, which is equal to the byte order defined by [1, IEEE 802.1 AS]. ⌋*(RS_TS_20048, RS_TS_20061, RS_TS_20062)*

5.3.1 Header format

5.3.1.1 Sync and Follow_Up acc. to IEEE 802.1AS

[PRS_TS_00061] ⌈ If MessageCompliance is set to TRUE, Sync and Follow_Up format shall be supported acc. to [1, IEEE 802.1 AS]. ⌋*(RS_TS_20048)*

Note: This implies one Time Domain (0).

The table below gives an overview, how an [1, IEEE 802.1 AS] conformant Sync looks like.

Sync Message Header [IEEE 802.1AS]				
High Nibble	**Low Nibble**	**Octets**	**Offset**	**Value**
transportSpecific	**message-type**	1	0	0x10
reserved	**versionPTP**	1	1	2
length of the message		2	2	44
domainNumber		1	4	(UInteger8) domainNumber = 0
reserved		1	5	0
flags		2	6	Octet 0: 0x02, Octet 1: 0x08
correctionField		8	8	0..281474976710655ns [1ns = 2^16 = 0x0000 0000 0001 0000]
reserved		4	16	0
sourcePortIden-tity		10	20	(PortIdentity) portIdentity from origin Time Aware End Station
sequenceId		2	30	(UInteger16) SyncSequenceId =(UInteger16) (pre-vSyncSequenceId+1)
control		1	32	0
logMessageInter-val		1	33	(Integer8) current-LogSyncInterval
Sync Message Fields [IEEE 802.1AS]				
High Nibble	**Low Nibble**	**Octets**	**Offset**	**Value**
PTP Message Header		34	0	[refer Sync Message Header]
reserved		10	34	0

The table below gives an overview, how an [1, IEEE 802.1 AS] conformant Follow_Up looks like.

Follow_Up Message Header [IEEE 802.1AS]

Follow_Up Message Header [IEEE 802.1AS]				
High Nibble	**Low Nibble**	**Octets**	**Offset**	**Value**
transportSpecific	**message-type**	1	0	`0x18`
reserved	**versionPTP**	1	1	`0x02`
length of the message		2	2	`76`
domainNumber		1	4	`(UInteger8) domainNumber = 0`
reserved		1	5	`0`
flags		2	6	`Octet 0: 0x00, Octet 1: 0x08`
correctionField		8	8	`0..281474976710655ns [1ns = 2^16 = 0x0000 0000 0001 0000]`
reserved		4	16	`0`
sourcePortIdentity		10	20	`(PortIdentity) portIdentity` from origin Time Aware End Station
sequenceId		2	30	`UInteger16) SyncSequenceId`
control		1	32	`2`
logMessageInterval		1	33	`(Integer8) currentLogSyncInterval`
Follow_Up Message Fields [IEEE 802.1AS]				
High Nibble	**Low Nibble**	**Octets**	**Offset**	**Value**
PTP Message Header		34	0	[refer `Follow_Up` Message Header]
preciseOriginTimestamp		10	34	(Timestamp) preciseOriginTimestamp
Follow_Up information TLV		32	44	refer `Follow_Up` information TLV
Follow_Up information TLV [IEEE 802.1AS]				
High Nibble	**Low Nibble**	**Octets**	**Offset**	**Value**
tlvType		2	0	3
lengthField		2	2	28
organizationId		3	4	0x0080c2
organizationSubType		3	7	1
cumulativeScaledRateOffset		4	10	`(Integer32) ((RateRatio-1)*2^41)`

▽

△

gmTimeBaseIndicator	2	14	0
lastGmPhaseChange	12	16	0
scaledLastGmFreqChange	4	28	0

5.3.1.2 Sync and Follow_Up acc. to AUTOSAR

[PRS_TS_00062] ⌈ If MessageCompliance is set to FALSE, the Sync and Follow_Up format shall be supported acc. to: Follow_Up Message Header [AUTOSAR] and Sync Message Header [AUTOSAR] depending on configuration. ⌋ *(RS_TS_20048, RS_TS_20061, RS_TS_20062)*

[PRS_TS_00063] ⌈ If MessageCompliance is set to FALSE, the Follow_Up shall contain an AUTOSAR *TLV*, depending on configuration. ⌋ *(RS_TS_20048, RS_TS_20061, RS_TS_20062)*

Message Header [AUTOSAR]

Sync Message Header [AUTOSAR]				
High Nibble	**Low Nibble**	**Octets**	**Offset**	**Value**
transportSpecific	**messagetype**	1	0	0x10
reserved	**versionPTP**	1	1	2
length of the message		2	2	44
domainNumber		1	4	(UInteger8) domainNumber = 0..15
reserved		1	5	0
flags		2	6	Octet 0: 0x02, Octet 1: 0x08
correctionField		8	8	0..281474976710655ns [1ns = 2^16 = 0x0000 0000 0001 0000]
reserved		4	16	0
sourcePortIdentity		10	20	(PortIdentity) portIdentity from origin Time Aware End Station

▽

△

sequenceId		2	30	(UInteger16) SyncSequenceId = (UInteger16) (prevSyncSequenceId+1)
control		1	32	0
logMessageInterval		1	33	(Integer8) currentLogSyncInterval
Sync Message Fields [AUTOSAR]				
High Nibble	**Low Nibble**	**Octets**	**Offset**	**Value**
PTP Message Header		34	0	[refer Sync Message Header]
reserved		10	34	0

Follow_Up Message Header [AUTOSAR]				
High Nibble	**Low Nibble**	**Octets**	**Offset**	**Value**
transportSpecific	**messagetype**	1	0	0x18
reserved	**versionPTP**	1	1	0x02
length of the message		2	2	76+10+Sum(Sub-TLVs)
domainNumber		1	4	(UInteger8) domainNumber = 0..15
reserved		1	5	0
flags		2	6	Octet 0: 0x00, Octet 1: 0x08
correctionField		8	8	0..281474976710655ns [1ns = 2^16 = 0x0000 0000 0001 0000]
reserved		4	16	0
sourcePortIdentity		10	20	(PortIdentity) portIdentity from origin Time Aware End Station
sequenceId		2	30	(UInteger16) SyncSequenceId
control		1	32	2
logMessageInterval		1	33	(Integer8) currentLogSyncInterval
Follow_Up Message Fields [AUTOSAR]				
High Nibble	**Low Nibble**	**Octets**	**Offset**	**Value**

▽

△

PTP Message Header		34	0	[refer `Follow_Up` Message Header]
preciseOrigin-Timestamp		10	34	(Timestamp) preciseOriginTimestamp
Follow_Up information TLV		32 + 10 + sum(`Sub-TLVs`)	44	[refer `Follow_Up` information TLV]
Follow_Up information TLV [IEEE 802.1AS]				
High Nibble	**Low Nibble**	**Octets**	**Offset**	**Value**
tlvType		2	0	3
length of the field		2	2	28
organizationId		3	4	`0x0080C2` [`IEEE 802.1AS`]
organizationSub-Type		3	7	1
cumulativeScale-dRateOffset		4	10	(Integer32)((RateRatio−1)∗2^41)
gmTimeBa-seIndicator		2	14	0
lastGm-PhaseChange		12	16	0
scaledLastGm-FreqChange		4	28	0
Follow_Up information TLV [AUTOSAR]				
High Nibble	**Low Nibble**	**Octets**	**Offset**	**Value**
AUTOSAR TLV Header				
tlvType		2	0	3
length of the field		2	0	6 + Sum(*Sub-TLVs*)
organizationId		3	4	0x1A75FB [AUTOSAR]
organizationSub-Type		3	7	0x605676 [BCD coded GlobalTimeEthTSyn]
AUTOSAR TLV Sub-TLV: Time Secured				
High Nibble	**Low Nibble**	**Octets**	**Offset**	**Value**
Type		1	0	0x28 [Time secured]
Length		1	1	3

▽

△

CRC_Time_Flags	1	2	BitMask 0x01 [length of the message] BitMask 0x02 [domainNumber] BitMask 0x04 [correctionField] BitMask 0x08 [sourcePortIdentity] BitMask 0x10 [sequenceId] BitMask 0x20 [preciseOriginTimestamp] BitMask 0x40 [reserved] BitMask 0x80 [reserved]
CRC_Time_0	1	3	0..255
CRC_Time_1	1	4	0..255
AUTOSAR TLV Sub-TLV: Status Secured			
High Nibble **Low Nibble**	**Octets**	**Offset**	**Value**
Type	1	0	0x50[Status secured]
Length	1	1	2
Status	1	2	BitMask 0x01 [SGW with SyncToGTM = 0 SyncToSubDomain = 1] BitMask 0x02 [reserved] BitMask 0x04 [reserved] BitMask 0x08 [reserved] BitMask 0x10 [reserved] BitMask 0x20 [reserved] BitMask 0x40 [reserved] BitMask 0x80 [reserved]
CRC_Status	1	3	0..255
AUTOSAR TLV Sub-TLV: Status Not Secured			
High Nibble **Low Nibble**	**Octets**	**Offset**	**Value**
Type	1	0	0x51 [Status Not Secured]
Length	1	1	2
Status	1	2	BitMask 0x01 [SGW with SyncToGTM = 0 SyncToSubDomain = 1] BitMask 0x02 [reserved] BitMask 0x04 [reserved] BitMask 0x08 [reserved] ▽

▽

△

			△ BitMask 0x10 [reserved] BitMask 0x20 [reserved] BitMask 0x40 [reserved] BitMask 0x80 [reserved]
reserved	1	3	0
AUTOSAR TLV Sub-TLV: UserData Secured			
High Nibble Low Nibble	**Octets**	**Offset**	**Value**
Type	1	0	0x60[UserData secured]
Length	1	1	5
UserDataLength	1	2	1..3
UserByte_0	1	3	0..255 (default: 0)
UserByte_1	1	4	0..255 (default: 0)
UserByte_2	1	5	0..255 (default: 0)
CRC_UserData	1	6	0..255
AUTOSAR TLV Sub-TLV: UserData Not Secured			
High Nibble Low Nibble	**Octets**	**Offset**	**Value**
Type	1	0	0x61 [UserData not secured]
Length	1	1	5
UserDataLength	1	2	1..3
UserByte_0	1	3	0..255 (default: 0)
UserByte_1	1	4	0..255 (default: 0)
UserByte_2	1	5	0..255 (default: 0)
reserved	1	6	0
AUTOSAR TLV Sub-TLV: OFS Secured			
High Nibble Low Nibble	**Octets**	**Offset**	**Value**
Type	1	0	0x44 [OFS secured]
Length	1	1	17
OfsTimeDomain	1	2	16..31
OfsTimeSec	6	3	0..281474976710655s
OfsTimeNSec	4	9	0..999999999ns
Status	1	13	BitMask 0x01 [SGW with SyncToGTM = 0 SyncToSubDomain = 1] BitMask 0x02 [reserved] BitMask 0x04 [reserved] BitMask 0x08 [reserved] ▽

▽

△

			△ BitMask 0x10 [reserved] BitMask 0x20 [reserved] BitMask 0x40 [reserved] BitMask 0x80 [reserved]
UserDataLength	1	14	0..3 (default: 0)
UserByte_0	1	15	0..255 (default: 0)
UserByte_1	1	16	0..255 (default: 0)
UserByte_2	1	17	0..255 (default: 0)
CRC_OFS	1	18	0..255
AUTOSAR TLV Sub-TLV: OFS Not Secured			
High Nibble Low Nibble	**Octets**	**Offset**	**Value**
Type	1	0	0x34 [OFS not secured]
Length	1	1	17
OfsTimeDomain	1	2	16..31
OfsTimeSec	6	3	0..281474976710655s
OfsTimeNSec	4	9	0..999999999ns
Status	1	13	BitMask 0x01 [SGW with SyncToGTM = 0 SyncToSubDomain = 1] BitMask 0x02 [reserved] BitMask 0x04 [reserved] BitMask 0x08 [reserved] BitMask 0x10 [reserved] BitMask 0x20 [reserved] BitMask 0x40 [reserved] BitMask 0x80 [reserved]
UserDataLength	1	14	0..3 (default: 0)
UserByte_0	1	15	0..255 (default: 0)
UserByte_1	1	16	0..255 (default: 0)
UserByte_2	1	17	0..255 (default: 0)
reserved	1	18	0

5.3.1.3 `Follow_Up` Message Header [AUTOSAR]

[PRS_TS_00066] ⌈ The `messageLength` of the `Follow_Up` Message Header has to be adapted according to the length of all existing *TLVs*. ⌋*(RS_TS_20048, RS_TS_20061, RS_TS_20062)*

5.3.1.4 AUTOSAR *TLV* Header

[PRS_TS_00067] ⌈ The AUTOSAR *TLV* Header has a multiplicity of 1. ⌋*(RS_TS_20048, RS_TS_20061, RS_TS_20062)*

[PRS_TS_00068] ⌈ If an AUTOSAR *TLV* Header exists, at least one AUTOSAR *Sub-TLV* must exist as well. ⌋*(RS_TS_20048, RS_TS_20061, RS_TS_20062)*

[PRS_TS_00069] ⌈ If an AUTOSAR *TLV* Header exists, the `lengthField` shall be adapted according the number of existing AUTOSAR *Sub-TLVs*. ⌋*(RS_TS_20048, RS_TS_20061, RS_TS_20062)*

5.3.1.5 AUTOSAR TLV Sub-TLVs

[PRS_TS_00070] ⌈ If an AUTOSAR *Sub-TLV* exists, it shall be placed after the AUTOSAR *TLV* Header. ⌋*(RS_TS_20048, RS_TS_20061, RS_TS_20062)*

[PRS_TS_00071] ⌈ If more than one AUTOSAR *Sub-TLV* exists, each *Sub-TLV* shall be placed after the preceding *Sub-TLV* without gaps. ⌋*(RS_TS_20048, RS_TS_20061, RS_TS_20062)*

[PRS_TS_00072] ⌈ If more than one AUTOSAR *Sub-TLV* exists, the positon of each *Sub-TLV* is arbitrary. ⌋*(RS_TS_20048, RS_TS_20061, RS_TS_20062)*

5.3.1.6 AUTOSAR *TLV Sub-TLV*: Time Secured

[PRS_TS_00074] ⌈ The AUTOSAR *Sub-TLV*: Time Secured has a multiplicity of 1 and is only available, if *CRC* protection is required. ⌋*(RS_TS_20061, RS_TS_20062)*

[PRS_TS_00075] ⌈ If `MessageCompliance` is `FALSE` and `TLVFollowUpTimeSubTLV` is set to `TRUE`, the Time Master shall send a `Follow_Up`, which contains an AUTOSAR *Sub-TLV*: Time Secured. ⌋*(RS_TS_20048, RS_TS_20061, RS_TS_20062)*

5.3.1.7 AUTOSAR *TLV Sub-TLV*: Status Secured / Not Secured

[PRS_TS_00076] ⌈ The AUTOSAR *Sub-TLV*: Status has a multiplicity of 1 and can either be *CRC* protected (Status Secured) or not (Status Not Secured). ⌋*(RS_TS_20061, RS_TS_20062)*

[PRS_TS_00077] ⌈ If `MessageCompliance` is set to `FALSE` and `TLVFollowUpStatusSubTLV` is set to `TRUE`, the Time Master shall send a `Follow_Up`, which contains an AUTOSAR *Sub-TLV*: Status. ⌋*(RS_TS_20048, RS_TS_20061, RS_TS_20062)*

5.3.1.8 AUTOSAR *TLV Sub-TLV*: UserData Secured / Not Secured

[PRS_TS_00078] ⌈ The AUTOSAR *Sub-TLV*: UserData has a multiplicity of 1 and can either be *CRC* protected (UserData Secured) or not (UserData Not Secured). ⌋*(RS_TS_20061, RS_TS_20062)*

[PRS_TS_00079] ⌈ If MessageCompliance is set to FALSE and Follow_Up TLV User Data configuration is set to TRUE, the Time Master shall send a Follow_Up, which contains an AUTOSAR *Sub-TLV*: UserData. ⌋*(RS_TS_20048, RS_TS_20061, RS_TS_20062)*

[PRS_TS_00081] ⌈ The AUTOSAR *Sub-TLV*: UserData shall be read from the current incoming message consistently. ⌋*(RS_TS_20061, RS_TS_20062)*

[PRS_TS_00082] ⌈ The AUTOSAR *Sub-TLV*: UserData shall be written to the next outgoing message consistently. ⌋*(RS_TS_20061, RS_TS_20062)*

5.3.1.9 AUTOSAR *TLV Sub-TLV*: OFS Secured / Not Secured

[PRS_TS_00084] ⌈ The AUTOSAR *Sub-TLV*: OFS has a multiplicity of 16 and can either be *CRC* protected (OFS Secured) or not (OFS Not Secured). ⌋*(RS_TS_20061, RS_TS_20062)*

[PRS_TS_00085] ⌈ The element OfsTimeDomain of the AUTOSAR *Sub-TLV*: OFS shall contain the Offset Time Domain identifier, which is in a range between 16 and 31. ⌋*(RS_TS_20061, RS_TS_20062)*

Note: Compared to CAN and FlexRay, Ethernet does need any optimization on payload bytes on bit-level.

[PRS_TS_00086] ⌈ If MessageCompliance is set to FALSE and TSynTLVFollowUpOFSSubTLV is set to TRUE, the Time Master shall send a Follow_Up, which contains at least one AUTOSAR *Sub-TLV*: OFS. ⌋*(RS_TS_20048, RS_TS_20061, RS_TS_20062)*

[PRS_TS_00088] ⌈ The User Data of the AUTOSAR *Sub-TLV*: OFS shall be read from an incoming message consistently. ⌋*(RS_TS_20061, RS_TS_20062)*

[PRS_TS_00089] ⌈ The User Data of the AUTOSAR *Sub-TLV*: OFS shall be written to an outgoing message consistently. ⌋*(RS_TS_20061, RS_TS_20062)*

5.3.2 Body/Payload format

Placeholder for upcoming Autosar releases.

5.3.3 Data Types

Refer to [1, IEEE 802.1 AS].

5.4 Message types

Refer to [1, IEEE 802.1 AS].

5.4.1 Data Messages

Refer to [1, IEEE 802.1 AS].

5.4.2 Control Messages

Refer to [1, IEEE 802.1 AS].

5.5 Services / Commands

Placeholder for upcoming Autosar releases.

5.6 Sequences (lower layer)

5.6.1 Pdelay Protocol for Latency Calculation

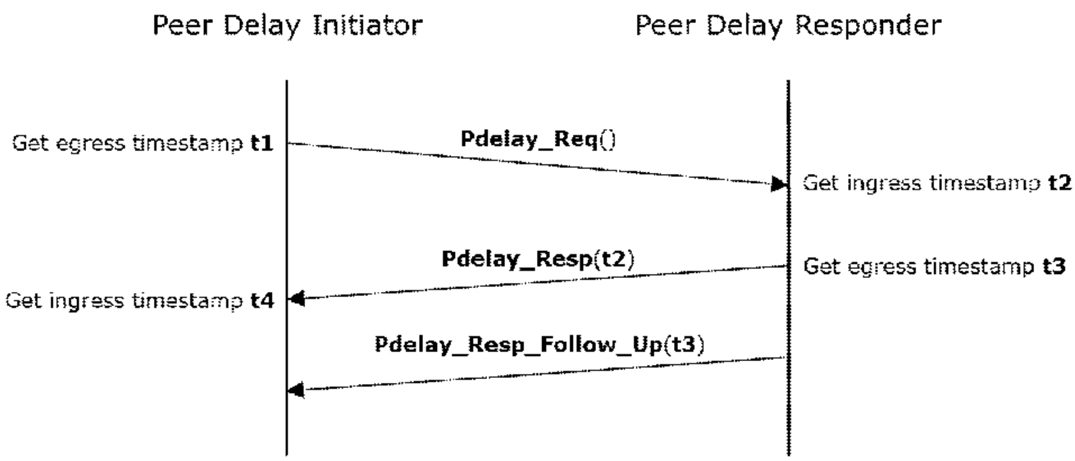

Figure 5.1: Propagation Delay Measurement (Pdelay)

[PRS_TS_00003] ⌈ The Time Sync module shall use for latency calculation

- either static Pdelay values (GlobalTimePropagationDelay)
- or runtime-based values calculated by Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up according to Figure 5.1: Propagation Delay Measurement (Pdelay)

depending on configuration of GlobalTimeTxPdelayReqPeriod ⌋*(RS_TS_20048, RS_TS_20066)*

[PRS_TS_00154] ⌈ If GlobalTimeTxPdelayReqPeriod is not equal to 0 and if the Pdelay latency calculation result exceeds PdelayLatencyThreshold, the measured value shall be discarded and the previous value shall be kept.⌋*(RS_TS_20048)*

[PRS_TS_00004] ⌈ A Pdelay_Resp timeout or incomplete Pdelay protocol shall stop the latency calculation algorithm. In such cases, the device shall use the latest successful calculated latency value.⌋*(RS_TS_20048, RS_TS_20051)*

Note: A timeout is detected, when sending the next subsequent Pdelay_Req before receiving the Pdelay_Resp resp. Pdelay_Resp_Follow_Up belonging to the Pdelay_Req before.

[PRS_TS_00164] ⌈ Time Master and Time Slave shall observe the Pdelay timeout as given by PdelayRespAndRespFollowUpTimeout , if a Pdelay_Req has been transmitted (waiting for Pdelay_Resp) or if a Pdelay_Resp has been received (waiting for Pdelay_Resp_Follow_Up). A value of 0 deactivates this timeout observation. ⌋(RS_TS_20048, RS_TS_20051)

[PRS_TS_00210] ⌈ If a reception timeout occurs (refer to [PRS_TS_00164]), any received Pdelay_Resp resp Pdelay_Resp_Follow_Up shall be ignored, until a new Pdelay_Req has been sent. ⌋(RS_TS_20048, RS_TS_20051)

[PRS_TS_00140] ⌈ If GlobalTimeTxPdelayReqPeriod equals 0, Time Master and Time Slave shall not measure the propagation delay. The Time Slave shall use a static value GlobalTimePropagationDelay as propagation delay instead. ⌋ (RS_TS_20066)

Note: Since GlobalTimeTxPdelayReqPeriod is ECU specific, neither a Time Master nor all Time Slaves have to measure the propagation delay. Global Time Synchronization in AUTOSAR does yet not define dynamic reconfiguration or backup strategies that will reassign the role as Time Master, therefore propagation delay measurements make currently no sense for a Time Master (although a Time Master shall be able to handle Pdelay_Req initiated by a Time Slave).

[PRS_TS_00141] ⌈ If GlobalTimeTxPdelayReqPeriod is greater than 0, Time Master and Time Slave shall cyclically measure the propagation delay using Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up as defined in [1, IEEE802.1 AS] chapter 11.1.2 "Propagation delay measurement". ⌋(RS_TS_20048, RS_TS_20066)

[PRS_TS_00149] ⌈ If GlobalTimeTxPdelayReqPeriod is greater than 0, Time Master and Time Slave shall cyclically measure the propagation delay only on that Time Domain with the lowest Time Domain ID and shall use this value to adjust all corresponding Time Bases. ⌋(RS_TS_20048, RS_TS_20066)

Note: There is no need to measure the propagation delay for all Time Domains, because the same value is expected. This requirement ensures also the usage of Time Domain 0 for Pdelay, to be compatible to [1, IEEE 802.1 AS].

[PRS_TS_00142] ⌈ If GlobalTimeTxPdelayReqPeriod is greater than 0, GlobalTimePropagationDelay shall be used as default value for the propagation delay, until first valid propagation delay has been measured. ⌋(RS_TS_20048, RS_TS_20066)

[PRS_TS_00011] ⌈ If GlobalTimeTxPdelayReqPeriod is greater than 0, Time Master and Time Slave shall periodically transmit Pdelay_Req for latency calculation with the cycle GlobalTimeTxPdelayReqPeriod as defined in [1, IEEE 802.1 AS] chapter 11.1.2 "Propagation delay measurement". ⌋(RS_TS_20048, RS_TS_20066)

Note: GlobalTimePdelayRespEnable allows disabling of Pdelay_Resp and Pdelay_Resp_Follow_Up, if no Pdelay_Req is expected to be received, i.e. for the Time Master, if all Time Slaves have set GlobalTimeTxPdelayReqPeriod to 0 or for any Time Slave if the Time Master has set GlobalTimeTxPdelayReqPeriod to 0.

[PRS_TS_00012] ⌈ If GlobalTimePdelayRespEnable is set to TRUE, Time Master and Time Slave shall react to Pdelay_Req by transmitting Pdelay_Resp for latency calculation as defined in [1, IEEE 802.1 AS] chapter 11.1.2 "Propagation delay measurement". ⌋*(RS_TS_20048, RS_TS_20066)*

[PRS_TS_00143] ⌈ If `GlobalTimePdelayRespEnable` is set to `FALSE`, `Pdelay_Resp` and `Pdelay_Resp_Follow_Up` shall be omitted. ⌋*(RS_TS_20066)*

5.6.2 Acting as Time Master

A Time Master is an entity which is the master for a certain Time Base and which propagates this Time Base to a set of Time Slaves within a certain segment of a communication network, being a source for this Time Base.

If a Time Master is also the owner of the Time Base then he is the Global Time master. A Time Gateway typically consists of one Time Slave and one or more Time Masters. When mapping time entities to real ECUs, an ECU could be Time Master (or even Global Time Master) for one Time Base and Time Slave for another Time Base.

5.6.2.1 Message Processing

[PRS_TS_00050] ⌈ The Time Master shall support the transmission of `Sync` and `Follow_Up` according as well as the transmission and reception of `Pdelay_Req`, `Pdelay_Resp` and `Pdelay_Resp_Follow_Up`. ⌋*(RS_TS_20047, RS_TS_20048)*

[PRS_TS_00016] ⌈ The Time Master shall periodically transmit `Sync` with the cycle `GlobalTimeTxPeriod` as defined in [1, IEEE 802.1 AS] chapter 11.1.3 "Transport of time-synchronization information", if the `GLOBAL_TIME_BASE` bit within the `timeBaseStatus`, which is read from the corresponding Time Base, is set and `GlobalTimeTxPeriod` is not 0. ⌋*(RS_TS_20047, RS_TS_20048)*

[PRS_TS_00018] ⌈ The `preciseOriginTimestamp` as calculated above, shall be used in the transmission of the `Follow_Up` as defined in [1, IEEE 802.1 AS] chapter 11.1.3 "Transport of time-synchronization information". ⌋*(RS_TS_20048)*

5.6.2.1.1 Frame Debouncing

[PRS_TS_00186] ⌈ If multiple frames are triggered at the same time, the frames shall be sent in the following order:

1. `Sync`
2. `Follow_Up`
3. `Pdelay_Req`
4. `Pdelay_Resp, Pdelay_Resp_Follow_Up`

⌋*(RS_TS_20047)*

5.6.2.2 Message Field Calculation and Assembling

[PRS_TS_00092] ⌈ If MessageCompliance is set to FALSE, a Time Master shall add an *AUTOSAR TLV* to the Follow_Up frame. ⌋*(RS_TS_20061, RS_TS_20062, RS_TS_20063)*

[PRS_TS_00091] ⌈ If MessageCompliance is set to FALSE, CRC_SUPPORT shall be considered. ⌋*(RS_TS_20061)*

[PRS_TS_00093] ⌈ Depending on CRC_SUPPORT the Follow_Up.TLV[AUTOSAR].Sub-TLV.Type shall be: ⌋*(RS_TS_20061)*

Follow_Up Message Header [IEEE 802.1AS]

	Sub-TLV.Type	
GlobalTimeTxCrcSecured	CRC_SUPPORTED	CRC_NOT_SUPPORTED
	0x28 *Sub-TLV*: Time Secured is *CRC* secured	n.a.
	0x50 *Sub-TLV*: Status is *CRC* secured	0x51 *Sub-TLV*: Status is not *CRC* secured
	0x60 *Sub-TLV*: UserData is *CRC* secured	0x61 *Sub-TLV*: UserData is not *CRC* secured
	0x44 *Sub-TLV*: OFS is *CRC* secured	0x34 *Sub-TLV*: OFS is not *CRC* secured

5.6.2.2.1 SGW Calculation

[PRS_TS_00094] ⌈ The *SGW* value (Time Gateway synchronization status) shall be mapped to the Status element of the *AUTOSAR Sub-TLV*: Status resp. the *AUTOSAR Sub-TLV*: OFS. If the SYNC_TO_GATEWAY is set, the *SGW* value shall be *SyncToSubDomain*. Otherwise, it shall be *SyncToGTM*. ⌋*(RS_TS_20052, RS_TS_20054)*

5.6.2.2.2 OFS Calculation

[PRS_TS_00095] ⌈ The Time Master of an Offset Time Base shall send the "second" part of the Offset Time Base value via the OfsTimeSec element of the corresponding AUTOSAR *Sub-TLV*: OFS and the "nanosecond" part of the Offset Time Base value via the OfsTimeNSec element of the corresponding AUTOSAR *Sub-TLV*: OFS ⌋*(RS_TS_20063)*

5.6.2.2.3 CRC Calculation

[PRS_TS_00097] ⌈ The DataID shall be calculated as: DataID = DataIDList[Follow_Up.sequenceId mod 16], where DataIDList is given by configuration for the Follow_Up. ⌋*(RS_TS_20061)*

Note: A specific DataID out of a predefined DataIDList ensures the identification of data elements of Time Synchronization messages.

[PRS_TS_00182] ⌈ If applying the *CRC* calculation on multibyte values, the byte order shall be such, that the byte containing the most significant bit of the value shall be used first. ⌋*(RS_TS_20061)*

[PRS_TS_00184] ⌈ If applying the *CRC* calculation on multibyte message data, the byte order shall be in ascending order of the octets, i.e., the octet with the lowest offset shall be used first. ⌋*(RS_TS_20061)*

5.6.2.2.3.1 AUTOSAR TLV *Sub-TLV*: Time Secured

[PRS_TS_00098] ⌈ If GlobalTimeTxCrcSecured is CRC_SUPPORTED, the Time Master shall write the contents of CrcTimeFlagsTxSecured to CRC_Time_Flags acc. to the following rule. ⌋*(RS_TS_20061)*

	CrcTimeFlagsTxSecured contents:	
CRC_Time_Flags	Follow_Up Message Header	Follow_Up Message Field
BitMask 0x01	CRCMessageLength	n.a.
BitMask 0x02	CRCDomainNumber	n.a.
BitMask 0x04	CrcCorrectionField	n.a.
BitMask 0x08	CRCSourcePortIdentity	n.a.
BitMask 0x10	CRCSequenceIdentity	n.a.
BitMask 0x20	n.a.	CRCPrecise-OriginTimestamp
BitMask 0x40	n.a.	n.a.
BitMask 0x80	n.a.	n.a.

[PRS_TS_00099] ⌈ If CrcTimeFlagsTxSecured is supported, the Time Master shall calculate the *CRC* for CRC_Time_0 by considering the contents of CRC_Time_Flags itself, the contents of the dependent fields as defined in CrcTimeFlagsTxSecured acc. to the rule in the table below and the DataID. The data elements used for the calculation of the CRC shall apply the following order:

1. the value of CRC_Time_Flags
2. the domainNumber inside the Follow_Up Message Header, if CRC_Time_Flags contains BitMask *0x02*

3. the sourcePortIdentity inside the Follow_Up Message Header, if *CRC_Time_Flags*contains BitMask *0x08*
4. the preciseOriginTimestamp inside the Follow_Up Message Field, if *CRC_Time_Flags* contains BitMask *0x20*
5. the DataID

⌋(*RS_TS_20061*)

	For CRC_Time_0 ***calculation considered contents:***	
If CRC_Time_Flags **is set to 1**	Follow_Up **Message Header**	Follow_Up **Message Field**
BitMask 0x01	n.a.	n.a.
BitMask 0x02	CRCDomainNumber	n.a.
BitMask 0x04	n.a.	n.a.
BitMask 0x08	CRCSourcePortIdentity	n.a.
BitMask 0x10	n.a.	n.a.
BitMask 0x20	n.a.	CRCPrecise-OriginTimestamp
BitMask 0x40	n.a.	n.a.
BitMask 0x80	n.a.	n.a.

Note: CRC_Time_Flags is having the same value like the configuration item CrcTimeFlagsTxSecured, whereas the resulting *CRC* of the dependent items remains network wide unchanged.

[PRS_TS_00100] ⌈ If GlobalTimeTxCrcSecured is set to CRC_SUPPORTED, the Time Master shall calculate the *CRC* for CRC_Time_1 by considering the contents of CRC_Time_Flags itself, the contents of the dependent fields as defined in CrcTimeFlagsTxSecured acc. to the rule in the table below and the DataID. The data elements used for the calculation of the *CRC* shall apply the following order:

1. the value of CRC_Time_Flags
2. the *messageLength* inside the Follow_Up Message Header, if CRC_Time_Flags contains BitMask 0x01
3. the CrcCorrectionField inside the Follow_Up Message Header, if CRC_Time_Flags contains BitMask 0x04
4. the sequenceId inside the Follow_Up Message Header, if CRC_Time_Flags contains BitMask 0x10
5. the DataID

⌋(*RS_TS_20061*)

	For CRC_Time_1 calculation considered contents:	
If CRC_Time_Flags is set to 1	Follow_Up Message Header	Follow_Up Message Field
BitMask 0x01	messageLength	n.a.
BitMask 0x02	n.a.	n.a.
BitMask 0x04	CrcCorrectionField.	n.a.
BitMask 0x08	n.a.	n.a.
BitMask 0x10	sequenceId	n.a.
BitMask 0x20	n.a.	n.a.
BitMask 0x40	n.a.	n.a.
BitMask 0x80	n.a.	n.a.

Note: CRC_Time_Flags has the same value as the configuration item CrcTimeFlagsTxSecured.

5.6.2.2.3.2 AUTOSAR TLV Sub-TLV: Status secured

[PRS_TS_00101] ⌈ If GlobalTimeTxCrcSecured is set to CRC_SUPPORTED, the Time Master shall calculate the *CRC* for CRC_STATUS by considering the contents of Status and DataID (in this order). ⌋*(RS_TS_20061)*

5.6.2.2.3.3 AUTOSAR TLV Sub-TLV: UserData secured

[PRS_TS_00102] ⌈ If GlobalTimeTxCrcSecured is set to CRC_SUPPORTED, the Time Master shall calculate the *CRC* for CRC_UserData by considering the contents of UserDataLength, UserByte_0, UserByte_1, UserByte_2 and DataID (in this order). ⌋*(RS_TS_20061)*

5.6.2.2.3.4 AUTOSAR TLV Sub-TLV: OFS secured

[PRS_TS_00103] ⌈ If GlobalTimeTxCrcSecured is set to CRC_SUPPORTED, the Time Master shall calculate the *CRC* for CRC_OFS by considering the contents of OfsTimeDomain, OfsTimeSec, OfsTimeNSec, Status, UserDataLength, UserByte_0, UserByte_1, UserByte_2 and DataID (in this order). ⌋*(RS_TS_20061, RS_TS_20062, RS_TS_20063)*

5.6.2.2.4 Message Assembling

[PRS_TS_000104] ⌈ For each transmission of a Time Synchronization message, the Time Synchronization module shall assemble the message as follows:

1. If Sync: Calculate Message Header
2. If Follow_Up: Calculate Follow_Up.preciseOriginTimestamp and Message Header inclusive CrcCorrectionField
3. If Follow_Up: Calculate IEEE *TLV*
4. If Follow_Up: Calculate AUTOSAR *TLV* (configuration dependent)

For 4: Calculate *CRC* (configuration dependent) and copy all data to the appropriate position within the related message ⌋*(RS_TS_20061, RS_TS_20062, RS_TS_20063)*

5.6.3 Acting as Time Slave

A Time Slave is an entity, which is the recipient for a certain Time Base within a certain segment of a communication network, being a consumer for this Time Base .

5.6.3.1 Message processing

[PRS_TS_00023] ⌈ The Time Slave shall support the reception of Sync and Follow_Up according [1, IEEE 802.1 AS] as well as the transmission and reception of Pdelay_Req, Pdelay_Resp and Pdelay_Resp_Follow_Up, [PRS_TS_00140], [PRS_TS_00141],[PRS_TS_00004]. ⌋*(RS_TS_20048)*

[PRS_TS_00025] ⌈ For each configured Time Slave the Ethernet module shall observe the reception timeout GlobalTimeFollowUpTimeout between the Sync and its Follow_Up. If the reception timeout occurs, the sequence shall be reset (i.e. waiting for a new Sync). A value of 0 deactivates this timeout observation. ⌋*(RS_TS_20048, RS_TS_20051)*

Note: A timeout is detected when receiving the next subsequent Sync before receiving the Follow_Up belonging to the Sync before. The general timeout monitoring for the Time Base update is located in the Implementation of Time Synchronization and not in the provider modules.

5.6.3.2 Message Field Validation and Disassembling

[PRS_TS_00105] ⌈ If MessageCompliance is set to FALSE, RxCrcValidated shall be considered. ⌋*(RS_TS_20061, RS_TS_20062, RS_TS_20063)*

[PRS_TS_00106] ⌈ If MessageCompliance is set to FALSE, a Time Slave shall check if an AUTOSAR *TLV* in the Follow_Up frame exists. ⌋*(RS_TS_20061, RS_TS_20062, RS_TS_20063)*

[PRS_TS_00107] ⌈ The *CRC* of the Follow_Up *TLV* shall be validated, depending on RxCrcValidated and the Follow_Up.TLV[AUTOSAR].Sub-TLV.Type acc. to: ⌋ *(RS_TS_20061)*

		Sub-TLV.Type
RxCrcValidated	CRC_VALIDATED	CRC_NOT_VALIDATED
	0x28 Sub-TLV: Time Secured is CRC secured	n.a.
	0x50 Sub-TLV: Status is CRC secured	0x51 Sub-TLV: Status is not CRC secured
	0x60 Sub-TLV: UserData is CRC secured	0x61 Sub-TLV: UserData is not CRC secured
	0x44 Sub-TLV: OFS is CRC secured	0x34 Sub-TLV: OFS is not CRC secured

[PRS_TS_00108] ⌈ The *CRC* of the Follow_Up *TLV* shall be ignored, if RxCrcValidated is set to CRC_IGNORED and the Follow_Up.TLV[AUTOSAR].Sub-TLV.Type contains any of the following defined values: ⌋ *(RS_TS_20061)*

	Sub-TLV.Type	
RxCrcValidated	CRC_IGNORED	
	0x28 Sub-TLV: Time Secured is CRC secured	n.a.
	0x50 Sub-TLV: Status is CRC secured	0x51 Sub-TLV: Status is not CRC secured
	0x60 Sub-TLV: UserData is CRC secured	0x61 Sub-TLV: UserData is not CRC secured
	0x44 Sub-TLV: OFS is CRC secured	0x34 Sub-TLV: OFS is not CRC secured

[PRS_TS_00109] ⌈ The *CRC* of the Follow_Up *TLV* shall be either validated or not validated, if RxCrcValidated is set to CRC_OPTIONAL and the Follow_Up.TLV[AUTOSAR].Sub-TLV.Type contains any of the following defined values: ⌋ *(RS_TS_20061)*

	Sub-TLV.Type	
RxCrcValidated	CRC_OPTIONAL	
	CRC shall be validated	CRC shall not be validated
	0x28 Sub-TLV: Time Secured is CRC secured	n.a.
	0x50 Sub-TLV: Status is CRC secured	0x51 Sub-TLV: Status is not CRC secured

▽

△

	0x60 Sub-TLV: UserData is CRC secured	0x61 Sub-TLV: UserData is not CRC secured
	0x44 Sub-TLV: OFS is CRC secured	0x34 Sub-TLV: OFS is not CRC secured

5.6.3.2.1 SGW Calculation

[PRS_TS_00207] ⌈ If MessageCompliance is set to TRUE the SYNC_TO_GATEWAY bit within timeBaseStatus shall be set to zero. ⌋*(RS_TS_20052)*

[PRS_TS_00156] ⌈ The *SGW* value (Time Gateway synchronization status) shall be retrieved from the Status element of the AUTOSAR *Sub-TLV*: Status resp. the AUTOSAR *Sub-TLV*: OFS. If the *SGW* value is set to SyncToSubDomain, the SYNC_TO_GATEWAY bit within timeBaseStatus shall be set. Otherwise, it shall be zero. ⌋*(RS_TS_20053, RS_TS_20054)*

[PRS_TS_00208] ⌈ If MessageCompliance is set to FALSE and if an *AUTOSAR Sub-TLV:* Status in the Follow_Up message does not exist, the SYNC_TO_GATEWAY bit within timeBaseStatus shall be set to zero. ⌋*(RS_TS_20052)*

[PRS_TS_00209] ⌈ If MessageCompliance is set to FALSE and if an *AUTOSAR TLV* in the Follow_Up message exists the *SGW* value (Time Gateway synchronization status) shall be retrieved from the Status element of each *AUTOSAR Sub-TLV*: OFS that is part of the *AUTOSAR TLV*. If the *SGW* value is set to SyncToSubDomain, the SYNC_TO_GATEWAY bit within timeBaseStatus shall be set to one. Otherwise, it shall be set to zero. ⌋*(RS_TS_20052)*

5.6.3.2.2 OFS Calculation

[PRS_TS_00110] ⌈ The Time Slave of an Offset Time Base shall calculate the Offset Time Base from the OfsTimeSec element of the corresponding AUTOSAR *Sub-TLV*: OFS and the OfsTimeNSec element of the corresponding AUTOSAR *Sub-TLV*: OFS. ⌋*(RS_TS_20063)*

5.6.3.2.3 CRC Validation

[PRS_TS_00112] ⌈ The DataID shall be calculated as: DataID = DataIDList[Follow_Up.sequenceId mod 16], where DataIDList is given by configuration for the Follow_Up. ⌋*(RS_TS_20061)*

Note: A specific DataID out of a predefined DataIDList ensures the identification of data elements of Time Synchronization messages.

[PRS_TS_00183] ⌈ If applying the *CRC* calculation on multibyte values, the byte order shall be such that the byte containing the most significant bit of the value shall be used first. ⌋*(RS_TS_20061)*

[PRS_TS_00185] ⌈ If applying the *CRC* calculation on multibyte message data, the byte order shall be in ascending order of the octets, i.e., the octet with the lowest offset shall be used first. ⌋*(RS_TS_20061)*

5.6.3.2.3.1 AUTOSAR TLV Sub-TLV: Time Secured

[PRS_TS_00157] ⌈ If RxCrcValidated is set to CRC_VALIDATED or CRC_OPTIONAL, the Time Slave shall validate the *CRC* as defined in CrcFlagsRxValidated acc. to the following rule. ⌋*(RS_TS_20061)*

	Validate if CrcFlagsRxValidated element is set to TRUE:	
Element	Follow_Up Message Header	Follow_Up Message Field
CrcMessageLength	messageLength	n.a.
CrcDomainNumber	domainNumber	n.a.
CrcCorrectionField	CrcCorrectionField	n.a.
CrcSourcePortIdentity	sourcePortIdentity	n.a.
CrcSequenceId	sequenceId	n.a.
CrcPrecise-OriginTimestamp	n.a.	preciseOriginTimestamp

[PRS_TS_00113] ⌈ If RxCrcValidated is set to CRC_VALIDATED or CRC_OPTIONAL, , the Time Slave shall validate the *CRC* for CRC_Time_0 by considering the contents of CRC_Time_Flags itself, the contents of the dependent fields as defined in CrcFlagsRxValidated acc. to the rule in the table below and the DataID. The data elements used for the calculation and thus validation of the *CRC* shall apply the following order:

1. the value of CRC_Time_Flags
2. the domainNumber inside the Follow_Up Message Header, if CrcDomainNumber is set to TRUE
3. the preciseOriginTimestamp inside the Follow_Up Message Field, if CrcPreciseOriginTimestampis set to TRUE
4. the DataID (refer to [PRS_TS_00112])
5. the sourcePortIdentity inside the Follow_Up Message Header, if CrcSourcePortIdentity is set to TRUE

⌋*(RS_TS_20061)*

	For CRC_Time_0 verification required contents:	
If CrcFlagsRxValidated element is set to TRUE:	Follow_Up Message Header	Follow_Up Message Field
CrcMessageLength	n.a.	n.a.
CrcDomainNumber	domainNumber	n.a.
CrcCorrectionField	n.a.	n.a.
CrcSourcePortIdentity	sourcePortIdentity	n.a.
CrcSequenceId	n.a.	n.a.
CrcPrecise-OriginTimestamp	n.a.	preciseOriginTimestamp

[PRS_TS_00114] ⌈ If RxCrcValidated is set to CRC_VALIDATED or CRC_OPTIONAL, the Time Slave shall validate the *CRC* for CRC_Time_1 by considering the contents of CRC_Time_Flags itself, the contents of the dependent fields as defined in CrcFlagsRxValidated acc. to the rule in the table below and the DataID. The data elements used for the calculation and thus validation of the *CRC* shall apply the following order:

1. the value of CRC_Time_Flags
2. the messageLength inside the Follow_Up Message Header, if CrcMessageLength is set to TRUE
3. the CrcCorrectionField inside the Follow_Up Message Header, if CrcCorrectionField is set to TRUE
4. the sequenceId inside the Follow_Up Message Field, if CrcSequenceId is set to TRUE
5. the DataID (refer to [PRS_TS_00112])

⌋*(RS_TS_20061)*

	For CRC_Time_1 verification required contents:	
If CrcFlagsRxValidated element is set to TRUE:	Follow_Up Message Header	Follow_Up Message Field
CrcMessageLength	messageLength	n.a.
CrcDomainNumber	n.a.	n.a.
CrcCorrectionField	CrcCorrectionField	n.a.
CrcSourcePortIdentity	n.a.	n.a.
CrcSequenceId	sequenceId	n.a.
CrcPrecise-OriginTimestamp	n.a.	n.a.

5.6.3.2.3.2 AUTOSAR TLV Sub-TLV: Status secured

[PRS_TS_00115] ⌈ If RxCrcValidated is set to CRC_VALIDATED or CRC_OPTIONAL, the Time Slave shall validate the *CRC* for CRC_Status by considering the contents of Status and DataID (in this order). ⌋*(RS_TS_20061)*

5.6.3.2.3.3 AUTOSAR TLV Sub-TLV: UserData secured

[PRS_TS_00116] ⌈ If RxCrcValidated is set to CRC_VALIDATED or CRC_OPTIONAL, the Time Slave shall validate the *CRC* for CRC_UserData by considering the contents of UserDataLength, UserByte_0, UserByte_1, UserByte_2 and DataID (in this order). ⌋*(RS_TS_20061)*

5.6.3.2.3.4 AUTOSAR TLV Sub-TLV: OFS secured

[PRS_TS_00117] ⌈ If RxCrcValidated is set to CRC_VALIDATED or CRC_OPTIONAL, the Time Slave shall validate the *CRC* for CRC_OFS by considering the contents of OfsTimeDomain, OfsTimeSec, OfsTimeNSec, Status, UserDataLength, UserByte_0, UserByte_1, UserByte_2 and DataID (in this order). ⌋*(RS_TS_20061, RS_TS_20063)*

5.6.3.2.4 Message Disassembling

[PRS_TS_00118] ⌈ If the Type of a *Sub-TLV* cannot be recognized at the receiver side, it shall be ignored and the next subsequent *Sub-TLV* shall be evaluated. ⌋*(RS_TS_20061, RS_TS_20062, RS_TS_20063)*

Note: The Length field of each *Sub-TLV* is always at the same position within each *Sub-TLV*. It will be used to jump over the unknown *Sub-TLV* to the next Type field.

[PRS_TS_00119] ⌈ For each received Time Synchronization message, the time synchronization protocol shall validate the message as follows (all conditions must match):

1. If Follow_Up: The sequenceId of the Follow_Up matches the sequenceId of the corresponding Sync.
2. If Follow_Up: Follow_Up.TLV[AUTOSAR].Sub-TLV.Type matches depending on configuration of RxCrcValidated
3. The Time Domain matches to the defined Time Domain range for each domainNumber resp. to the element *OfsTimeDomain* of the AUTOSAR *Sub-TLV*: OFS (configuration dependent).
4. The Time Domain matches to one of the configured Time Domains

5. If Follow_Up: The range of the element OfsTimeNSec of the AUTOSAR *Sub-TLV*: OFS matches the defined range of 0..999999999.

6. If *Follow_Up*: All *CRCs* (including DataID) matching depending on the configuration of RxCrcValidated and CrcFlagsRxValidated.

⌋*(RS_TS_20061, RS_TS_20062, RS_TS_20063)*

[PRS_TS_00120] ⌈ For each received Time Synchronization message, the Time synchronization protocol shall disassemble the message after successful validation. ⌋ *(RS_TS_20061, RS_TS_20062, RS_TS_20063)*

5.7 Time measurement with Switches

In a time aware Ethernet network, two basic HW types of control units exists:

1. Endpoints directly working on a local Ethernet-Controller

2. Time Gateways, resp. Time Aware Bridges, where the local Ethernet-Controller connects to an external Switch device.

The extension "Time measurement with Switches" focusses on 2. A Switch device leads to additional delays, which have to be considered for the calculation of the corresponding Time Base. Additionally, the support of time stamping in HW is a Switch-Port specific feature, which leads to an extension of the used function APIs. These APIs enabling a Switch port specific detection of ingress and egress messages together with a given timestamp, if enabled.

If the Switch Management and Global Time support is implemented as a part of the program running on the Switch HW, this will not be considered by 2. For this case, the behavior can be seen as described in 1.

[PRS_TS_00053] ⌈ Time measurement with Switches supports the use case "Time Aware Bridge with GTM as Management CPU" like shown in Figure Figure 5.2. ⌋(RS_TS_20048, RS_TS_20059)

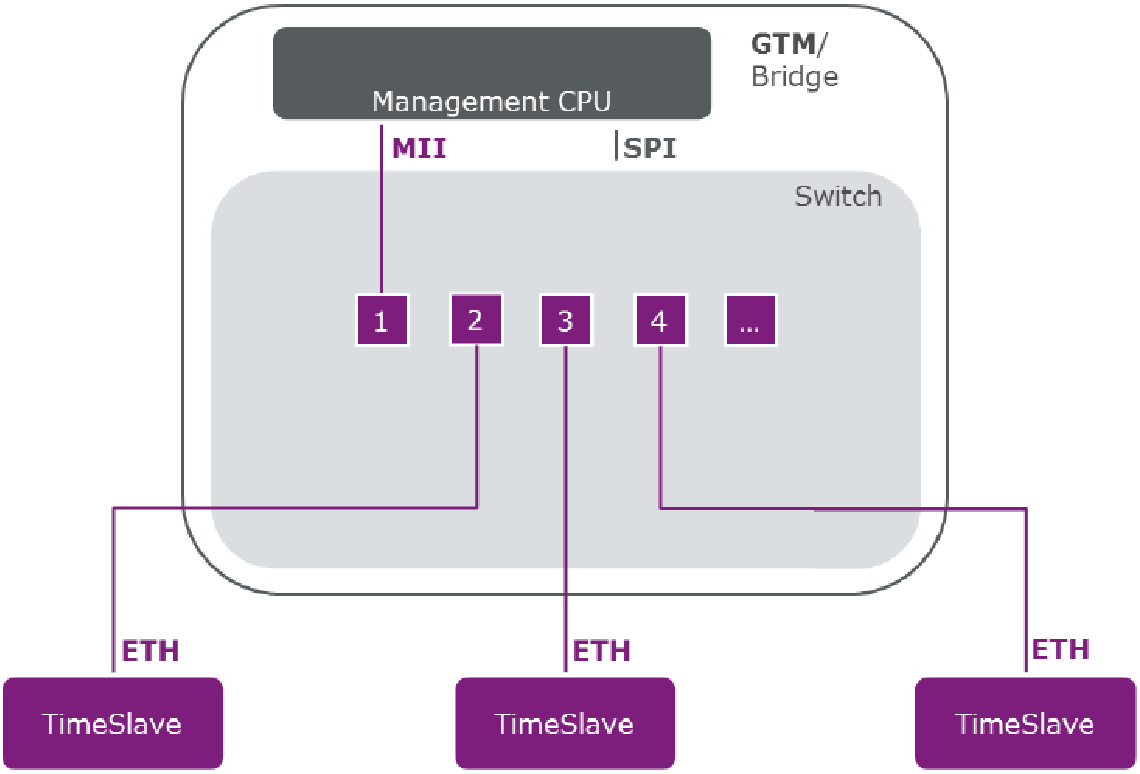

Figure 5.2: Time Aware Bridge with GTM as Management CPU

[PRS_TS_00054] ⌈ Time measurement with Switches supports the use case "Time Aware Bridge with GTM not as Management

CPU" like shown in Figure 5.3. ⌋(RS_TS_20048, RS_TS_20059)

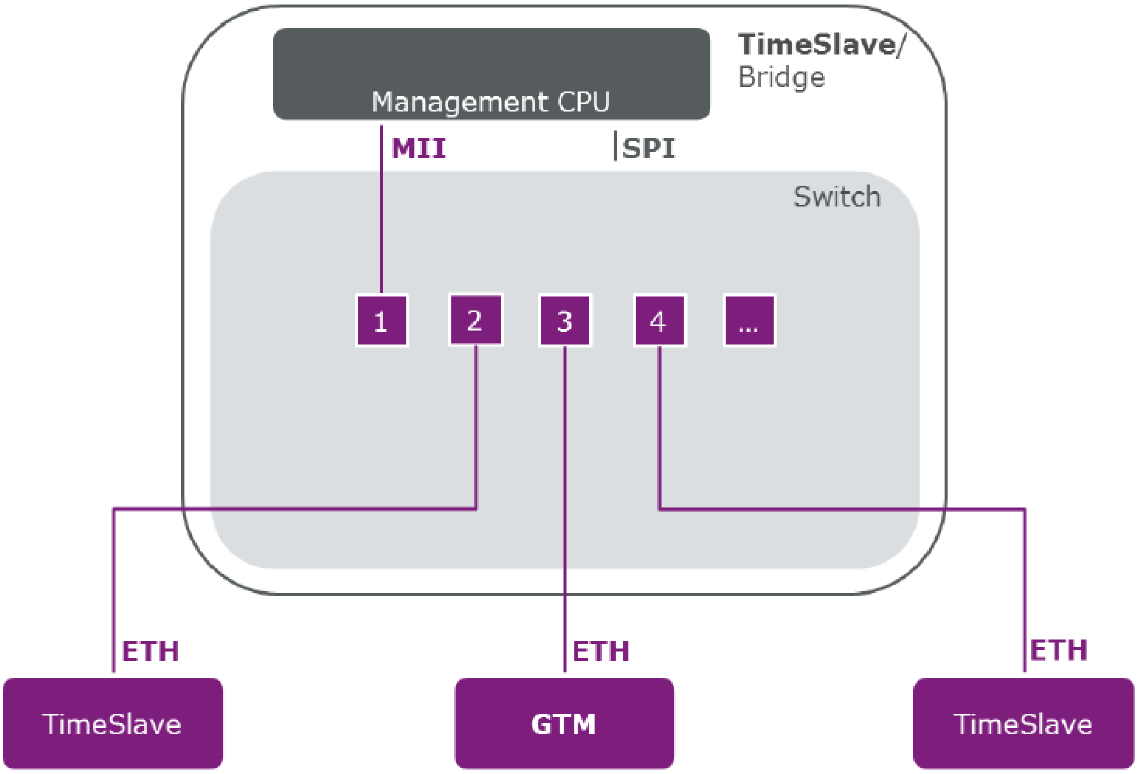

Figure 5.3: Time Aware Bridge with GTM not as Management CPU

5.8 Pdelay and Time Synchronization measurement point

[PRS_TS_00055] ⌈ The path delay measurement will be done always as Port-to-Port measurement like specified in in [1, IEEE 802.1 AS] chapter 11.1.2 Propagation delay measurement for the device external Ethernet path. ⌋(RS_TS_20048, RS_TS_20059)

[PRS_TS_00056] ⌈ The inner delay of the Ethernet path (Residence Time) is determined at the time where Sync is received and transmitted, by using the message specific ingress and egress timestamps. ⌋(RS_TS_20048, RS_TS_20059)

Note: This belongs to the fact, that the Residence Time might be discontinuous, depending on the current busload, while Sync messages are transmitted / received, the Switch HW architecture and the message forwarding method. A static delay measurement method for this part of the communication path might lead to an unprecise time measurement. Nevertheless, static Residence Time parameters are considered by this specification, to increase the performance while calculating the Global Time resp. the CrcCorrectionField and the flexibility to support different Switch devices, such as Switches, which do not support time stamping on each ingress or egress port.

5.9 Time Aware Bridge with GTM as Management CPU

[PRS_TS_00057] ⌈ Time measurement with Switches supporting the use case "Time Aware Bridge with GTM as Management CPU" following the given timestamping points like shown in Figure 5.4 and Figure 5.5 ⌋ *(RS_TS_20048, RS_TS_20059)*

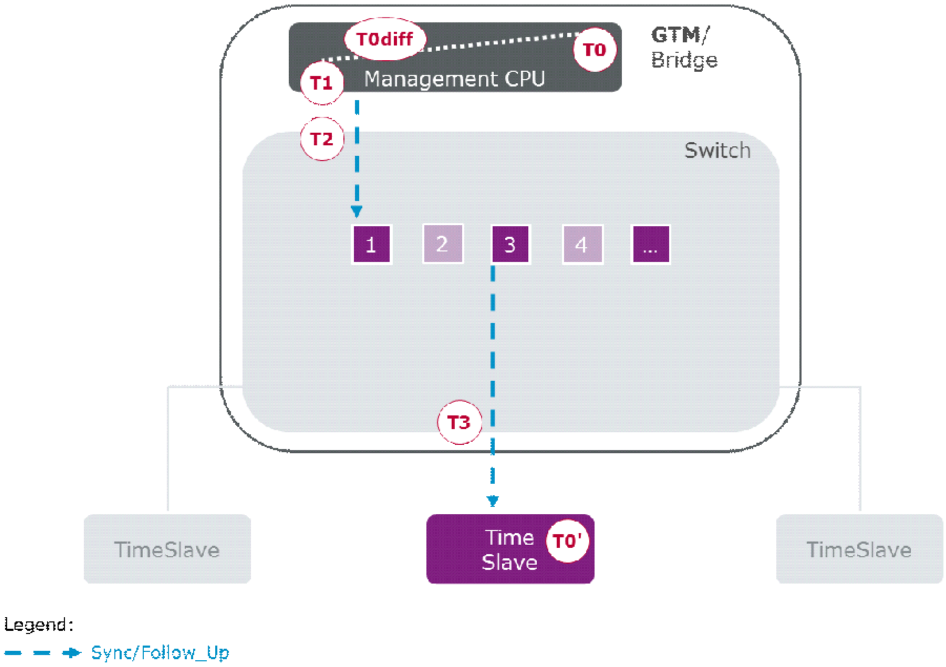

Figure 5.4: Sync/Follow_Up message flow with Timestamping points for Sync for Time Aware Bridge with GTM as Management CPU

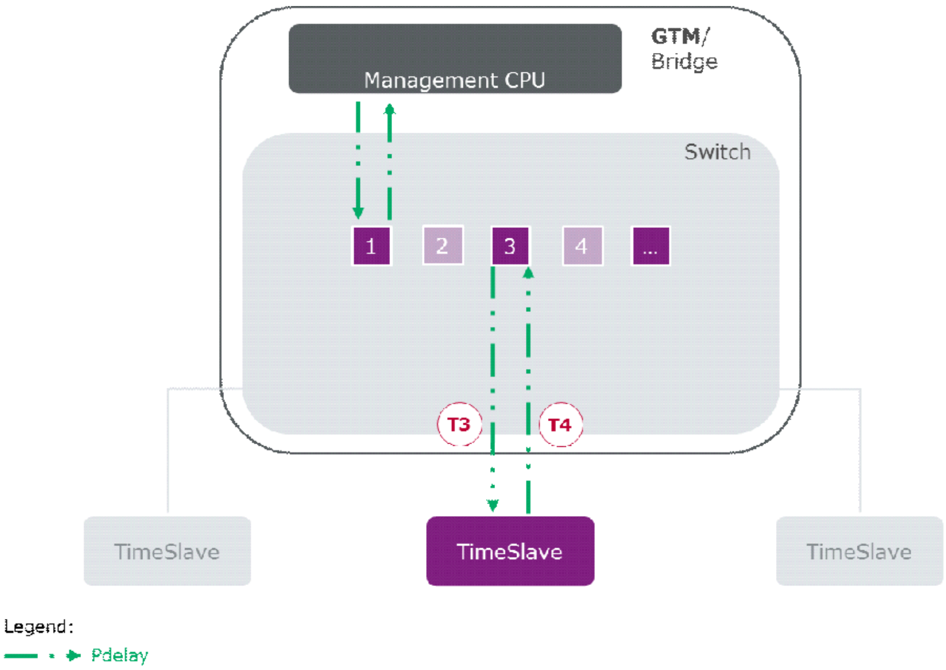

Figure 5.5: Pdelay message flow with Timestamping points for Time Aware Bridge with GTM as Management CPU

Note: The picture Figure 5.4 and Figure Figure 5.5 shows an example Port selection as simplification.

[PRS_TS_00058] ⌈ Time measurement with Switches supporting the use case "Time Aware Bridge with GTM as Management CPU" considers the inner Switch delay by a modification of the CrcCorrectionField as well as Pdelay timestamping for requestReceiptTimestamp and responseOriginTimestamp like shown in Figure 5.6. ⌋*(RS_TS_20048, RS_TS_20059)*

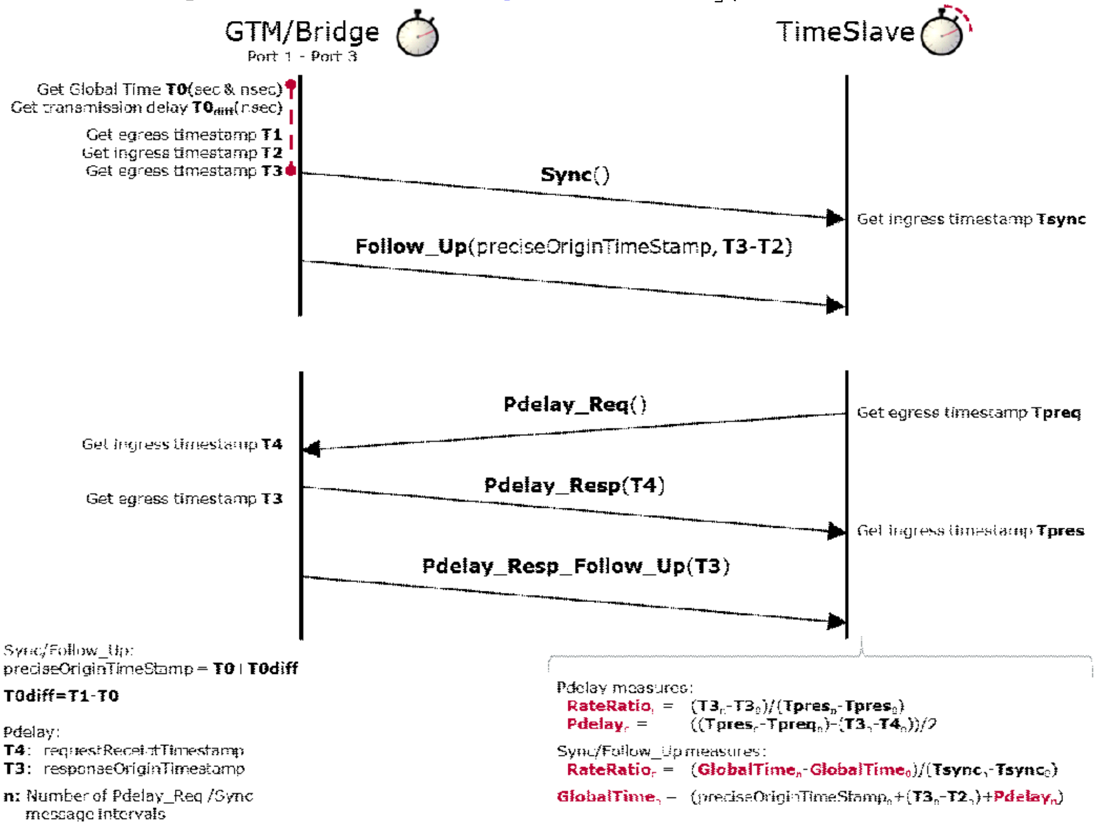

Figure 5.6: Timestamping sequence for Time Aware Bridge with GTM as Management CPU

Note: The calculation in Figure 5.6 shows an example Port selection as simplification.

[PRS_TS_00166] ⌈ If GlobalTimeUplinkToTxSwitchResidenceTime is set to 0, the Ethernet module shall ignore this parameter and measure the inner delay of the Switch egress Ethernet path (Uplink to Tx Residence Time (**T3 - T2**)) by using always the ingress (**T2**) and egress (**T3**) timestamp as given in Figure 5.6. ⌋*(RS_TS_20048, RS_TS_20059)*

[PRS_TS_00167] ⌈ If GlobalTimeUplinkToTxSwitchResidenceTime is greater than 0, the Ethernet module shall use this parameter as value for the inner delay of the Switch egress Ethernet path (Uplink to Tx Residence Time (**T3 - T2**)) instead of using the measurement method described in [PRS_TS_00166]. ⌋*(RS_TS_20048, RS_TS_20059)*

5.10 Time Aware Bridge with GTM not as Management CPU

[PRS_TS_00059] ⌈ Time measurement with Switches supporting the use case Time Aware Bridge with GTM not as Management CPU following the given timestamping points like shown in Figure 5.7 and Figure 5.8. ⌋(*RS_TS_20048, RS_TS_20059*)

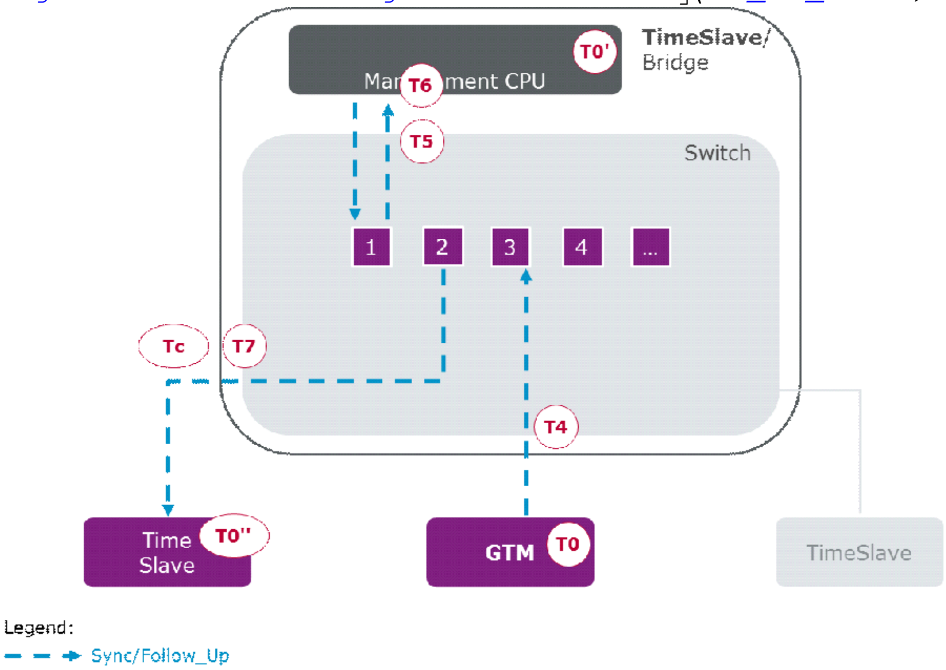

Figure 5.7: Sync/Follow_Up message flow with Timestamping points for Sync for Time Aware Bridge with GTM not as Management CPU

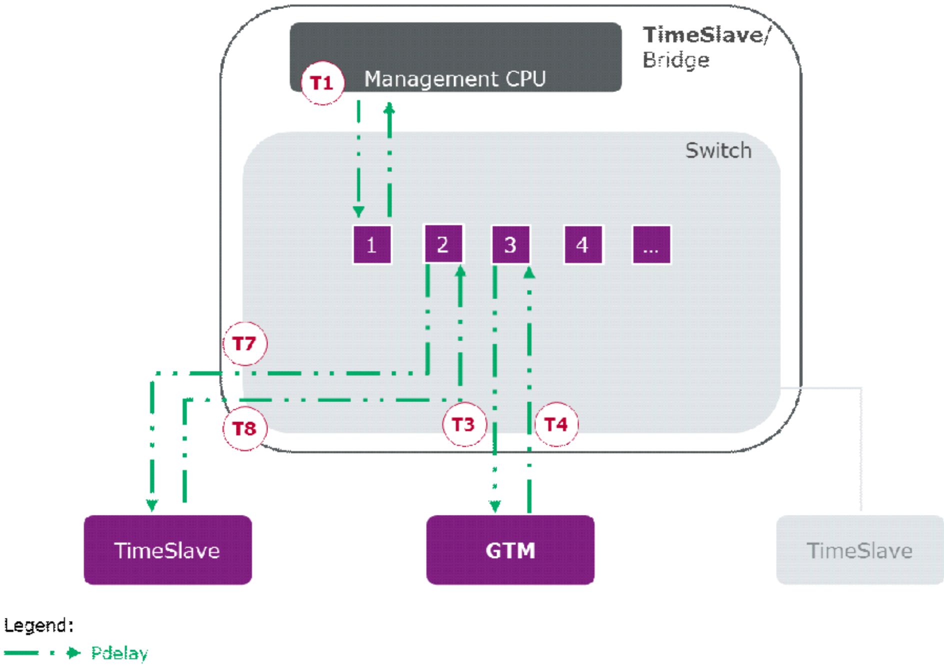

Figure 5.8: Pdelay message flow with Timestamping points for Time Aware Bridge with GTM not as Management CPU

[PRS_TS_00060] ⌈ Time measurement with Switches supporting the use case Time Aware Bridge with GTM not as Management CPU considers the inner Switch delay by a modification of the `CrcCorrectionField` as well as `Pdelay timestamping` for `requestReceiptTimestamp` and `responseOriginTimestamp`. ⌋ *(RS_TS_20048, RS_TS_20059)*

[PRS_TS_00207] ⌈ If the `Follow_Up` message contains an AUTOSAR *TLV*, which contains a *Sub-TLV*: Time Secured it shall be checked, if the element `CRC_Time_Flags` contains BitMask `0x04` (i.e., the content of `CrcCorrectionField` is *CRC* protected). If this bit is set then the validation of the `CRC_Time_1` element shall be done as follows: The *CRC* Validation shall be done as specified in section 5.6.3.2.3. The data elements used for the calculation and thus validation of the CRC shall be applied with the following order:

1. the value of `CRC_Time_Flags`
2. the `length of the message` inside the `Follow_Up` Message Header, if the element `CRC_Time_Flags` contains BitMask `0x01`
3. the `CrcCorrectionField` inside the `Follow_Up` Message Header
4. the `CRCsequenceId` inside the `Follow_Up` Message Header, if the element `CRC_Time_Flags` contains BitMask `0x10`
5. the *DataID*

⌋ *(RS_TS_20048, RS_TS_20059)*

[PRS_TS_00208] ⌈ If the validation CRC validation of an *AUTOSAR TLV* fails, the Follow_Up message shall be dropped instead of being forwarded. ⌋*(RS_TS_20048, RS_TS_20059)*

[PRS_TS_00209] ⌈ If the validation CRC validation of an *AUTOSAR TLV* is successful, the CrcCorrectionField shall be modified and the element CRC_Time inside the *Sub-TLV*: Time Secured shall be calculated according to the content of the CRC_Time_Flags element. ⌋*(RS_TS_20048, RS_TS_20059)*

[PRS_TS_00168] ⌈ If rx_residence_time is set to 0, the Time Synchronization over Ethernet shall ignore this parameter and measure the inner delay of the Switch ingress Ethernet path (Rx to Uplink Residence Time (T5 - T4)) by using always the ingress (T4) and egress (T5) timestamp. ⌋*(RS_TS_20048, RS_TS_20059)*

[PRS_TS_00171] ⌈ If rx_residence_time is greater than 0, the Time Synchronization over Ethernet shall use this parameter as value for the inner delay of the Switch ingress Ethernet path (Rx to Uplink Residence Time (T5 - T4)) instead of using the measurement method. ⌋*(RS_TS_20048, RS_TS_20059)*

[PRS_TS_00169] ⌈ If rx_residence_time and tx_residence_time are set to 0, the Ethernet module shall ignore both parameter and measure the inner delay of the Switch ingress and egress Ethernet path (Rx to Uplink and Uplink to Tx Residence Time (T7 to T4)) by using always the ingress (T4) and egress (T7) timestamp. ⌋ *(RS_TS_20048, RS_TS_20059)*

[PRS_TS_00170] ⌈ If rx_residence_time and tx_residence_time are greater than 0, the Ethernet module shall use the sum of both parameter for the value of the inner delay of the Switch ingress and egress Ethernet path (Rx to Uplink and Uplink to Tx Residence Time (T7 to T4)) instead of using the measurement method ⌋ *(RS_TS_20048, RS_TS_20059)*

Note: A separate Uplink to Tx Residence Time (T7 to $T_{UplinkMmCpu}$) replacement by using tx_residence_time might be also possible, but is not considered by the scenario.

5.11 Error messages

Error handling is specified in the corresponding classic and adaptive platform documents.

6 Configuration parameters

The Following chapter summarizes all the configuration parameters that are used.

Name	Description
CRCSupport	represents whether the configuration is supported or not

rx_residence_time	This parameter is specifying the default value used for the residence time
tx_residence_time	This parameter is specifying the default value used for the residence time
FramePrio	This optional parameter, if present, indicates the priority of outgoing messages, if sent via VLAN (used for the 3-bit PCP field of the VLAN tag). If this optional parameter is not present, frames are sent without a priority and VLAN field.
GlobalTimeTxPdelayReqPeriod	This parameter represents configuration of the TX period for Pdelay_Req messages. A value of 0 disables the cyclic Pdelay measurement.
PdelayLatencyThreshold	Threshold for calculated Pdelay. If a measured Pdelay exceeds PdelayLatencyThreshold, this value is discarded.
PdelayRespAndResp-FollowUpTimeout	Timeout value for Pdelay_Resp and Pdelay_Resp_Follow_Up after a Pdelay_Req has been transmitted resp. a Pdelay_Resp has been received. A value of 0 deactivates this timeout observation.
GlobalTimePropagationDelay	If cyclic propagation delay measurement is enabled, this parameter represents the default value of the propagation delay until the first actually measured propagation delay is available. If cyclic propagation delay measurement is disabled, this parameter replaces a measured propagation delay by a fixed value.
GlobalTimePdelayRespEnable	This parameter allows disabling Pdelay_Resp, Pdelay_Resp_Follow_Up transmission, if no Pdelay_Req messages are expected. FALSE: No Pdelay requests expected. Pdelay_Resp / Pdelay_Resp_Follow_Up transmission is disabled. TRUE: Pdelay requests expected. Pdelay_Resp, Pdelay_Resp_Follow_Up transmission is enabled.
GlobalTimeTxPeriod	This parameter represents configuration of the TX period.
GlobalTimeFollowUpTimeout	Timeout value of the Follow_Up message (of the subsequent Sync message). A value of 0 deactivates this timeout observation.
MasterSlaveConflictDetection	Enables master / slave conflict detection and notification. true: detection and notification is enabled. false: detection and notification is disabled.
MessageCompliance	true: IEEE 802.1AS compliant message format will be used. false: IEEE 802.1AS message format with AUTOSAR extension will be used.

RxCrcValidated	• CRC_IGNORED (ignores any CRC inside the Sub-TLVs) • CRC_NOT_VALIDATED (If MessageCompliance is set to FALSE: Ethernet discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60) • CRC_OPTIONAL (If MessageCompliance is set to FALSE: Ethernet discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an incorrect CRC value.) • CRC_VALIDATED (If MessageCompliance is set to FALSE: Ethernet discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an incorrect CRC value. Ethernet rejects Follow_Up messages with Sub-TLVs of Type 0x34, 0x51 or 0x61)
CrcFlagsRxValidated	This container collects definitions which parts of the Follow_Up message elements shall be included in the CRC validation.
CrcMessageLength	messageLength from the Follow_Up Message Header shall be included in CRC calculation.
CrcDomainNumber	domainNumber from the Follow_Up Message Header shall be included in CRC calculation.
CrcCorrectionField	correctionField from the Follow_Up Message Header shall be included in CRC calculation.
CrcSourcePortIdentity	sourcePortIdentity from the Follow_Up Message Header shall be included in CRC calculation.
CrcSequenceId	sequenceId from the Follow_Up Message Header shall be included in CRC calculation.
CrcPreciseOriginTimestamp	preciseOriginTimestamp from the Follow_Up Message Field shall be included in CRC calculation.
GlobalTimeUplinkToTxSwitchResidenceTime	This parameter is specifying the default value used for the residence time of the Ethernet Switch [Uplink to Egress]. This value is used by the Ethernet module if the calculation of the residence time failed.
TLVFollowUpTimeSubTLV	This represents the configuration of whether an AUTOSAR Follow_Up TLV Time Sub-TLV is used or not.
TLVFollowUpStatusSubTLV	This represents the configuration of whether an AUTOSAR Follow_Up TLV Status Sub-TLV is used or not.
TLVFollowUpUserDataSubTLV	This represents the configuration of whether an AUTOSAR Follow_Up TLV UserData Sub-TLV is used or not.
TSynTLVFollowUpOFSSubTLV	This represents the configuration of whether an AUTOSAR Follow_Up TLV OFS Sub-TLV is used or not.

CrcTimeFlagsTxSecured	This item collects definitions which parts of the Follow_Up message elements shall be used for CRC calculation.
CRCcorrectionField	correctionField from the Follow_Up Message Header shall be included in CRC calculation.
GlobalTimeTxCrcSecured	This represents the configuration of whether or not CRC is supported.

Table 6.1: Configuration Parameters

7 Protocol usage and guidelines

Please note that chapter 5 provides several requirements on usage.

8 References

References

[1] IEEE Standard 802.1AS-30
http://standards.ieee.org/getieee802/download/802.1AS-2011.pdf

[2] IEEE 802.1Q-2011 - IEEE Standard for Local and metropolitan area networks - Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks