Document Title	Specification of Time Synchronization over Ethernet
Document Category	Software specification (SWS)
Document UID	676
Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R26-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2026-11-26	R26-11	AUTOSAR Release Management	• Virtual Free Running Clock added • Rate deviation granularity improved • APIs reentrancy attribute udpated • Chapter 8 restructured • GetVersionInfo APIs and Multicore Distribution requirements removed
2025-11-27	R25-11	AUTOSAR Release Management	• Cleanup and Update EthTSyn Configuration • Improve mandatory/optional interfaces • Improve specification of behavior of Timesync message debouncing • Enable PostBuild configuration for GlobalTimeDomainId • Update Uptraces
2024-11-27	R24-11	AUTOSAR Release Management	• Changed lower layer to LSduR • Offset Time Bases removed • Retry strategy for transmission added • Clarification for configuration of VLAN
2023-11-23	R23-11	AUTOSAR Release Management	• Integrated Support of PTP physical clock adjustment • Rx Time Tuple calculation improvements • Single shot behaviour for EthTSynCyclicMsgResumeTime

△

2022-11-24	R22-11	AUTOSAR Release Management	• Support for "Secured Time Synchronization" added • AUTOSAR TLV processing enhanced • Several minor clarifications and corrections
2021-11-25	R21-11	AUTOSAR Release Management	• Origin Time Stamp calculation corrected • Sync reception delay corrected • Sequence Counter specified • Removed De-Init and re-Init requirments • New parameter for handling of Sequence Counter jumps introduced • Migration to Latex Based document
2020-11-30	R20-11	AUTOSAR Release Management	• Clarification of Follow_Up information TLV message • Clarification of Safety validation service interface • Sequence Counter specified • Improvement the structure of the Error classification • Clarification of EthTSynPortConfig
2019-11-28	R19-11	AUTOSAR Release Management	• Time Validation (draft) • Clarification regarding cyclic operation entry after timebase startup • Clarification regarding transmission and reception of User Bytes • Clarified SGW value handling for missing Sub-TLVs • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• Modifications to enhance precision of Global Time Synchronization • Split into FO Protocol Spec and CP SWS
2017-12-08	4.3.1	AUTOSAR Release Management	• Clarification of handling of unexpected Sub-TLVs • Clarification for configuration parameter • Clarification of handling FUP messages

▽

△

2016-11-30	4.3.0	AUTOSAR Release Management	• Resident time compensation for switches added • AUTOSAR specific TLV added • Interface to StbM and EthIf reworked (incl. support for immediate Timesync message transmission) • Various enhancements and corrections (e.g. postbuild configuration)
2015-07-31	4.2.2	AUTOSAR Release Management	• <Bus>TSyn_SetTransmissionMode changed to return "void" • Call of StbM_UpEthSetGlobalTime() added - sequence diagrams corrected • 'const' added to input arguments passed by pointer
2014-10-31	4.2.1	AUTOSAR Release Management	• Initial release

Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Introduction and functional overview 9
2 Acronyms, Abbreviations and Definitions 11
3 Related documentation 13
3.1 Input documents . . . 13
3.2 Related specification . . . 14
4 Constraints and assumptions 15
4.1 Limitations . . . 15
4.2 Accuracy . . . 16
4.3 Applicability to car domains . . . 16
5 Dependencies to other modules 17
5.1 File structure . . . 19
5.1.1 Code file structure . . . 19
6 Requirements Tracing 20
7 Functional specification 23
7.1 Overview . . . 23
7.1.1 General . . . 23
7.1.2 VLAN Support . . . 23
7.1.3 Message Transmission . . . 23
7.2 Initialization . . . 24
7.3 Handling of different Virtual Local Time sources . . . 24
7.4 Virtual Free-Running Clock . . . 25
7.5 Debounce Time . . . 28
7.6 Pdelay Protocol . . . 29
7.6.1 Pdelay Message Transmission . . . 29
7.6.2 Pdelay Message Reception . . . 31
7.6.3 Pdelay measurements with PHC correction . . . 33
7.7 Message Format . . . 34
7.7.1 Sync and Follow_Up acc. to IEEE 802.1AS . . . 34
7.7.2 Sync and Follow_Up acc. to AUTOSAR . . . 34
7.7.2.1 Follow_Up Message Header [AUTOSAR] . . . 35
7.7.2.2 AUTOSAR and OEM Sub-TLV's . . . 35
7.7.2.3 Pdelay Message Header . . . 36
7.8 Acting as Time Master . . . 36
7.8.1 Message processing . . . 38
7.8.1.1 Runtime Error detection . . . 41
7.8.1.2 Frame Debouncing . . . 41
7.8.1.3 Immediate Time Synchronization . . . 43
7.8.1.4 Secure Time Synchronization . . . 45

7.8.2 Link State and Transmission Mode . . . 48
7.8.3 Message Field Calculation and Assembling . . . 49
7.8.3.1 SGW Calculation . . . 49
7.8.3.2 CRC Calculation . . . 49
7.8.3.3 Sequence Counter (sequenceId) Calculation . . . 50
7.8.3.4 ICV Generation . . . 50
7.8.3.5 Message Assembling . . . 50
7.8.3.6 Dynamic port configuration for Time Master or Time Slave . . . 51
7.8.4 System Wide Global Time Master . . . 51
7.9 Acting as Time Slave . . . 52
7.9.1 Message processing . . . 52
7.9.1.1 Runtime Error detection . . . 57
7.9.1.2 Frame Debouncing . . . 57
7.9.1.3 Secure Time Synchronization . . . 58
7.9.2 Message Field Validation and Disassembling . . . 62
7.9.2.1 SGW Calculation . . . 62
7.9.2.2 CRC Validation . . . 62
7.9.2.3 Sequence Counter (sequenceId) Validation . . . 63
7.9.2.4 ICV Verification . . . 63
7.9.2.5 Message Disassembling . . . 63
7.10 Time Gateway . . . 63
7.10.1 Differences to a Time Aware Bridge . . . 63
7.11 Time Recording . . . 64
7.11.1 Time Validation . . . 64
7.11.1.1 Recording of Pdelay Measurement . . . 66
7.12 Security Events . . . 68
7.13 Error Classification . . . 70
7.13.1 Development Errors . . . 70
7.13.2 Runtime Errors . . . 71
7.13.3 Production Errors . . . 71
8 API specification 72
8.1 Imported types . . . 72
8.2 Type definitions . . . 73
8.2.1 EthTSyn_ConfigType . . . 73
8.2.2 EthTSyn_TransmissionModeType . . . 73
8.3 Function definitions . . . 74
8.3.1 EthTSyn_Init . . . 74
8.3.2 EthTSyn_SetTransmissionMode . . . 74
8.3.3 EthTSyn_SetProtocolParam . . . 75
8.3.4 EthTSyn_GetProtocolParam . . . 76
8.3.5 EthTSyn_GetCurrentTimeTuple . . . 77
8.3.6 EthTSyn_SetAndGetPhcCorrection . . . 77
8.3.7 EthTSyn_RxIndication . . . 78

8.3.8 EthTSyn_TxConfirmation . . . 79
8.3.9 EthTSyn_TrcvLinkStateChg . . . 80
8.3.10 EthTSyn_IcvGenerationIndication . . . 80
8.3.11 EthTSyn_IcvVerificationIndication . . . 81
8.4 Scheduled functions . . . 82
8.4.1 EthTSyn_MainFunction . . . 82
8.5 Expected Interfaces . . . 82
8.5.1 Mandatory Interfaces . . . 82
8.5.2 Optional Interfaces . . . 83
8.5.3 Callout Interfaces . . . 84
8.5.4 Configurable Interfaces . . . 84
8.6 Service Interfaces . . . 84
9 Sequence diagrams 85
9.1 EthIf_EnableEgressTimeStamp . . . 85
9.2 Time Synchronization Sequence . . . 86
9.3 Pdelay Measurement Sequence . . . 87
9.4 EthTSyn Egress Timestamping . . . 88
9.5 EthTSyn Ingress Timestamping . . . 89
9.6 Time measurement with Switches . . . 90
9.6.1 Time Aware Bridge with GTM as Management CPU - Tx . . . 90
9.6.2 Time Aware Bridge without GTM as Management CPU - Tx . . . 91
9.6.3 Time Aware Bridge without GTM as Management CPU - Rx . . . 92
9.7 Secure Time Synchronization Sequence . . . 93
10 Configuration specification 94
10.1 How to read this chapter . . . 94
10.2 Containers and configuration parameters . . . 94
10.2.1 EthTSyn . . . 95
10.2.2 EthTSynGeneral . . . 97
10.2.3 EthTSynSecurityEventRefs . . . 100
10.2.4 EthTSynGlobalTimeDomain . . . 103
10.2.5 EthTSynGlobalTimeFollowUpDataIDList . . . 105
10.2.6 EthTSynGlobalTimeFollowUpDataIDListElement . . . 106
10.2.7 EthTSynPTPPort . . . 107
10.2.8 EthTSynPTPPortDetails . . . 111
10.2.9 EthTSynCMLDS . . . 116
10.2.10 EthTSynPhysicalPortConfiguration . . . 118
10.2.11 EthTSynVirtualPortConfiguration . . . 121
10.2.12 EthTSynGlobalTimeTxIcvGeneration . . . 123
10.2.13 EthTSynGlobalTimeRxIcvVerification . . . 126
10.2.14 EthTSynTxPdu . . . 129
10.2.15 EthTSynRxPdu . . . 131
10.2.16 EthTSynPdelayConfig . . . 132

10.2.17 EthTSynGlobalTimeMasterConfig . . . 135
10.2.18 EthTSynCrcTimeFlagsTxSecured . . . 140
10.2.19 EthTSynGlobalTimeSlaveConfig . . . 141
10.2.20 EthTSynCrcFlagsRxValidated . . . 146
10.3 Constraints . . . 149
10.4 Configuration Constraints . . . 150
10.5 Published Information . . . 152
A Not applicable requirements 153
B Change history of AUTOSAR traceable items 154
B.1 Traceable item history of this document according to AUTOSAR Release R26-11 . . . 154
B.1.1 Added Specification Items in R26-11 . . . 154
B.1.2 Changed Specification Items in R26-11 . . . 154
B.1.3 Deleted Specification Items in R26-11 . . . 155
B.1.4 Added Constraints in R26-11 . . . 155
B.1.5 Changed Constraints in R26-11 . . . 155
B.1.6 Deleted Constraints in R26-11 . . . 155
B.2 Traceable item history of this document according to AUTOSAR Release R25-11 . . . 155
B.2.1 Added Specification Items in R25-11 . . . 155
B.2.2 Changed Specification Items in R25-11 . . . 156
B.2.3 Deleted Specification Items in R25-11 . . . 157

1 Introduction and functional overview

The EthTSyn module handles the Time Synchronization Protocol on Ethernet as specified in [1, PRS Time Synchronization Over Ethernet Protocol].

In addition to what is specified in [1, PRS Time Synchronization Over Ethernet Protocol] the EthTSyn module supports the following features:

- Debouncing of Timesync PDUs to avoid that a PDU with higher priority blocks those with lower priority
- "Immediate" transmission of Time Synchronization messages for fast (re-) synchronization of a Time Master and a Time Slave

The EthTSyn is tightly coupled to the Synchronized Time-Base Manager (StbM; refer to [2, SWS-SynchronizedTimeBaseManager]), which is responsible for interpolating (a local instance of) a Synchronized Time Base between the reception of 2 consecutive Sync messages for that Time Base. The StbM also provides the service interface for Time Synchronization to the application. Figure 1 shows the Time Synchronization related modules in the AUTOSAR Layered Architecture.

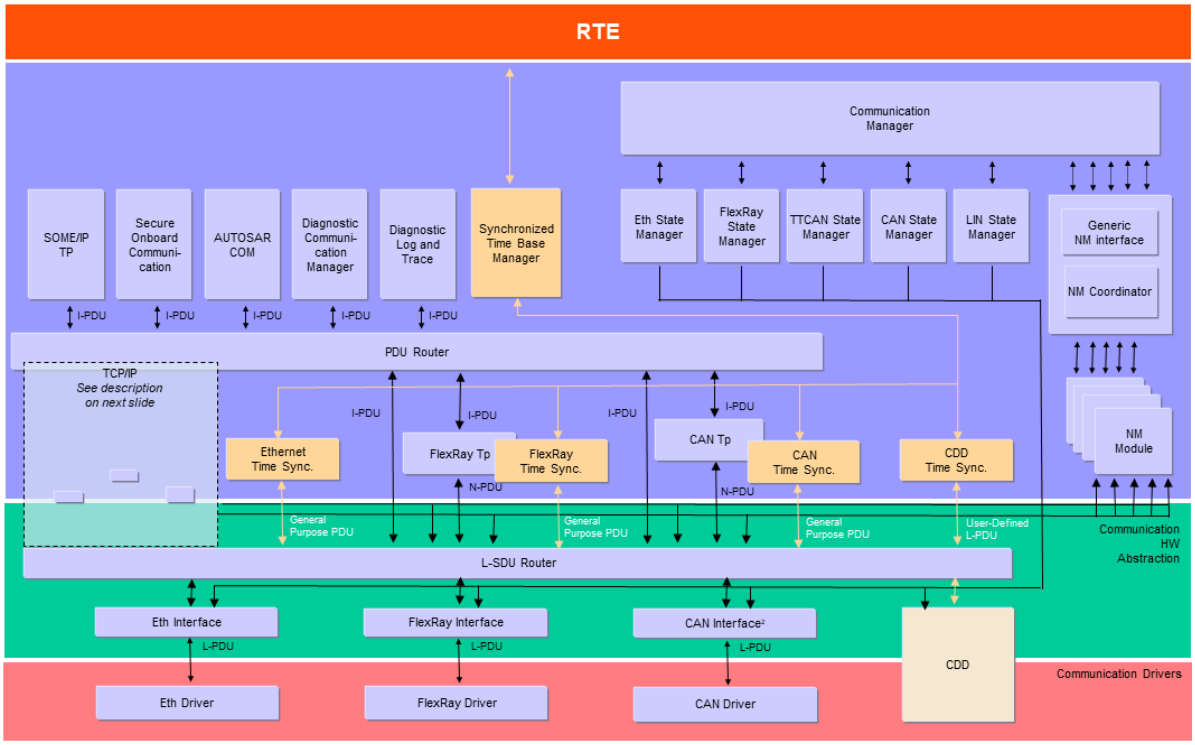


Figure 1.1: Timesync modules in the AUTOSAR Layered Architecture

The EthTSyn supports securing the global time messages on the Ethernet communication bus. The figure below shows the time provider mod-

ules interface with the security modules in the AUTOSAR Layered Architecture.

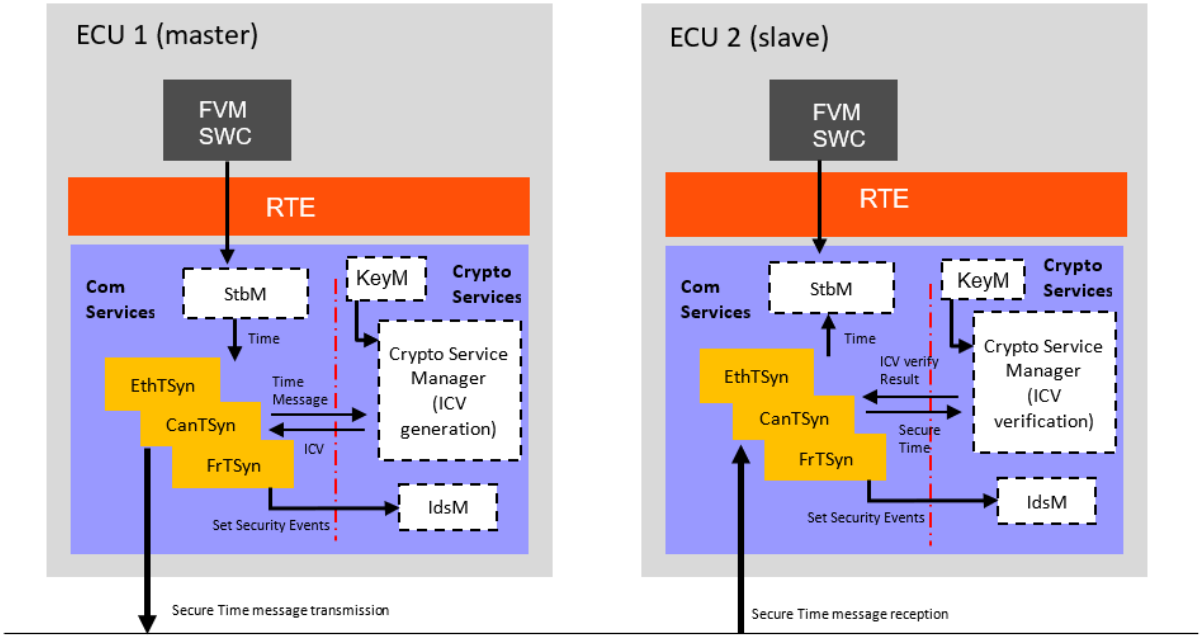


Figure 1.2: Timesync modules interface with security modules in the AUTOSAR Layered Architecture

2 Acronyms, Abbreviations and Definitions

This section lists module local Abbreviations and Definitions. For a complete set of Synchronized Time Base related terms refer to the corresponding chapter in [3, CP-SWS-BSWGeneral].

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
CSM	Crypto Service Manager
Debounce Time	Minimum gap between sending (Event) messages.
DEM	Diagnostic Event Manager
DET	Default Error Tracer
ETH	Ethernet
EthTSyn	Time Synchronization Provider module for Ethernet
Follow_Up	Time transport message (Follow-Up)
FV	Freshness Value
FVM	Freshness Value Manager
GM(C)	Grand Master (Clock)
ICV	Integrity Check Value
LSduR	L-SDU Router module
MAC [context - Ethernet protocol]	Media Access Control
MAC [context - security]	Message Authentication Code
meanPropagationDelay	meanPropagationDelay as defined by [4, IEEE Std 802.1AS-2011]
neighborRateRatio	neighborRateRatio as defined by [4, IEEE Std 802.1AS-2011]
Pdelay	Propagation / path delay as given in [4, IEEE Std 802.1AS-2011]
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
PHC	PTP Hardware Clock. A hardware clock that can be adjusted in offset and rate (also referred to as Disciplined Hardware Clock)
PTP	Precision Time Protocol
rateRatio	rateRatio as defined by [4, IEEE Std 802.1AS-2011]
StbM	Synchronized Time-Base Manager
Timesync	Time Synchronization
Sync	Time synchronization message (Sync)

△

Abbreviation / Acronym:	Description
TG	Time Gateway
TLV	Type, Length, Value field (acc. to [4, IEEE Std 802.1AS-2011])
TS	Time Slave
TSD	Time Sub-domain
VFRC	Virtual Free-Running Clock. A clock value maintained by Eth TSyn that represents the time of the PHC as if no PHC correction had been applied, see Chapter 7.4
VLAN	Virtual Local Area Network

Table 2.1: Acronyms and Abbreviations

3 Related documentation

3.1 Input documents

[1] Time Synchronization over Ethernet Protocol Specification
AUTOSAR_FO_PRS_TimeSyncOverEthernetProtocol

[2] Specification of Synchronized Time-Base Manager
AUTOSAR_CP_SWS_SynchronizedTimeBaseManager

[3] General Specification of Basic Software Modules
AUTOSAR_CP_SWS_BSWGeneral

[4] IEEE 802.1AS-2011 - IEEE Standard for Local and metropolitan area networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks, Rev. 2011

[5] IEEE 802.1AS-2020 - IEEE Standard for Local and metropolitan area networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks, Rev. 2020

[6] IEEE P 802.1ASds - Amendment: Support for the IEEE Std 802.3 Clause 4 Media Access Control (MAC) operating in half-duplex

[7] IEEE 802.1ASdm-2024 - IEEE Standard for Local and metropolitan area networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks, Amendment 3: Hot Standby and Clock Drift Error Reduction

[8] Explanation of Time Sensitive Network features
AUTOSAR_FO_EXP_TimeSensitiveNetworkFeatures

[9] Requirements on Time Synchronization
AUTOSAR_FO_RS_TimeSync

[10] General Requirements on Basic Software Modules
AUTOSAR_CP_RS_BSWGeneral

[11] Specification of Crypto Service Manager
AUTOSAR_CP_SWS_CryptoServiceManager

[12] Specification of CRC Library
AUTOSAR_CP_SWS_CRCLibrary

[13] System Template
AUTOSAR_CP_TPS_SystemTemplate

[14] Specification of Intrusion Detection System Manager
AUTOSAR_CP_SWS_IntrusionDetectionSystemManager

3.2 Related specification

AUTOSAR provides

- a General Specification on Basic Software [3, SWS BSW General] which is also valid for EthTSyn and
- a Time Synchronization Protocol Specification [1, PRS Time Synchronization Over Ethernet Protocol] which is also valid for EthTSyn.

Thus, the specification [3, SWS BSW General] and [1, PRS Time Synchronization Over Ethernet Protocol] shall be considered as additional and required specification for Eth TSyn.

4 Constraints and assumptions

This section gives limitations with respect to the protocol implemented, specifically compared to [4, IEEE Std 802.1AS-2011]. Constraints of the configuration can be found in 10.4.

4.1 Limitations

- No support of BMCA protocol, like specified in [4, IEEE 802.1 AS].
- No support of Announce and Signaling messages, like specified in [4, IEEE 802.1 AS].
- The reception of a Pdelay_Req is not taken as a pre-condition to start with the transmission of Sync messages.
- The Rate Correction will be performed by the StbM, (refer to [2]) based on Sync messages, which does not require the Pdelay mechanism, though the IEEE Standard mandates to calculate the rate correction based on Pdelay messages. This is considered to be a deviation from the IEEE-Standard, but it is considered to be interoperable. For some applications, e.g. for Audio/Video, it might be necessary to use Pdelay based Rate Correction performed by EthTSyn itself, which is optional and not considered by this specification.
- The Time Validation use case (Time Validation enabled) requires that the Pdelay measurement appears for a higher layer Validation application as if it was performed with timestamps from that Global Time Base that needs to be validated. The relevant timestamps are therefore mapped to the local instance of that Global Time. This is not considered to be a deviation from the IEEE-Standard, as no restrictions on the on-wire timestamps arise, i.e. one can still put Virtual Local Time into the PTP messages for each and every Pdelay measurement; only the corresponding instances of Global Time must be made available.
- EthTSyn will not maintain the Ethernet HW clock, but may use it as a source for the Virtual Local Time.
- While [4, IEEE Std 802.1AS-2011] states, that IEEE 802.1AS messages shall not have a VLAN tag nor a priority tag and are to be sent to a specific reserved multicast address, `EthTSynPTPPortDetails` allows to configure a `EthTSynGlobalTimeEthIfRef` using a VLAN while configuring a `EthTSynFramePrio` and `EthTSynDestPhyAddr`.
- "CRC secured" in the context of this document refers to CRC integrity protection mechanism and does not imply that CRC is used as a cybersecurity solution.
- No support of securing the messages of PDelay protocol.
- The Common Mean Link Delay Service (CMLDS) is adopted as a configuration option from [5, IEEE Std 802.1AS-2020].

- The configuration options for executing one-way Pdelay are adopted from [6, IEEE P802.1ASds] to prepare for operation on shared media.
- The configuration options for the `DriftTrackingTLV` are adopted from [7, IEEE Std 802.1ASdm-2024] to allow accuracy improvements in large diameter networks.
- A configuration option to use synchronized hardware clocks (PHC) for time-stamping is introduced without technical detail. The implementation details are vendor specific.

4.2 Accuracy

The accuracy of Time Synchronization depends on various factors (e.g., oscillator accuracy, number of bridges in the network path, configuration, ...). Refer to [8] Chapter 4.7 *"Ensure Accuracy of Time Synchronization"*, for recommendations on how to properly configure the overall system for highest possible accuracy.

4.3 Applicability to car domains

Automotive systems requiring a common Time Base for ECUs regardless of which bus system the ECUs are connected to.

5 Dependencies to other modules

The Global Time Synchronization over Ethernet (EthTSyn) has interfaces towards the Synchronized Time-Base Manager (StbM), the Ethernet Interface (EthIf), the L-SDU Router (LSduR), the Basic Software Mode Manager (BswM), the Crypto Service Manager (CSM), the Intrusion Detection System Manager (IdsM) and the Default Error Tracer (DET).

- StbM -
  - Get and set the current time value
  - Get FV from FVM
- EthIf - Receiving and transmitting messages
- LSduR - Routing of L-SDUs between EthIf and upper layer modules (e.g. Eth TSyn),
- BswM - Coordination of network access
- DET - Reporting of development errors
- CSM -
  - Generation of ICV for Time Master
  - Verification of ICV for Time Slave
- IdsM - Reporting of security events

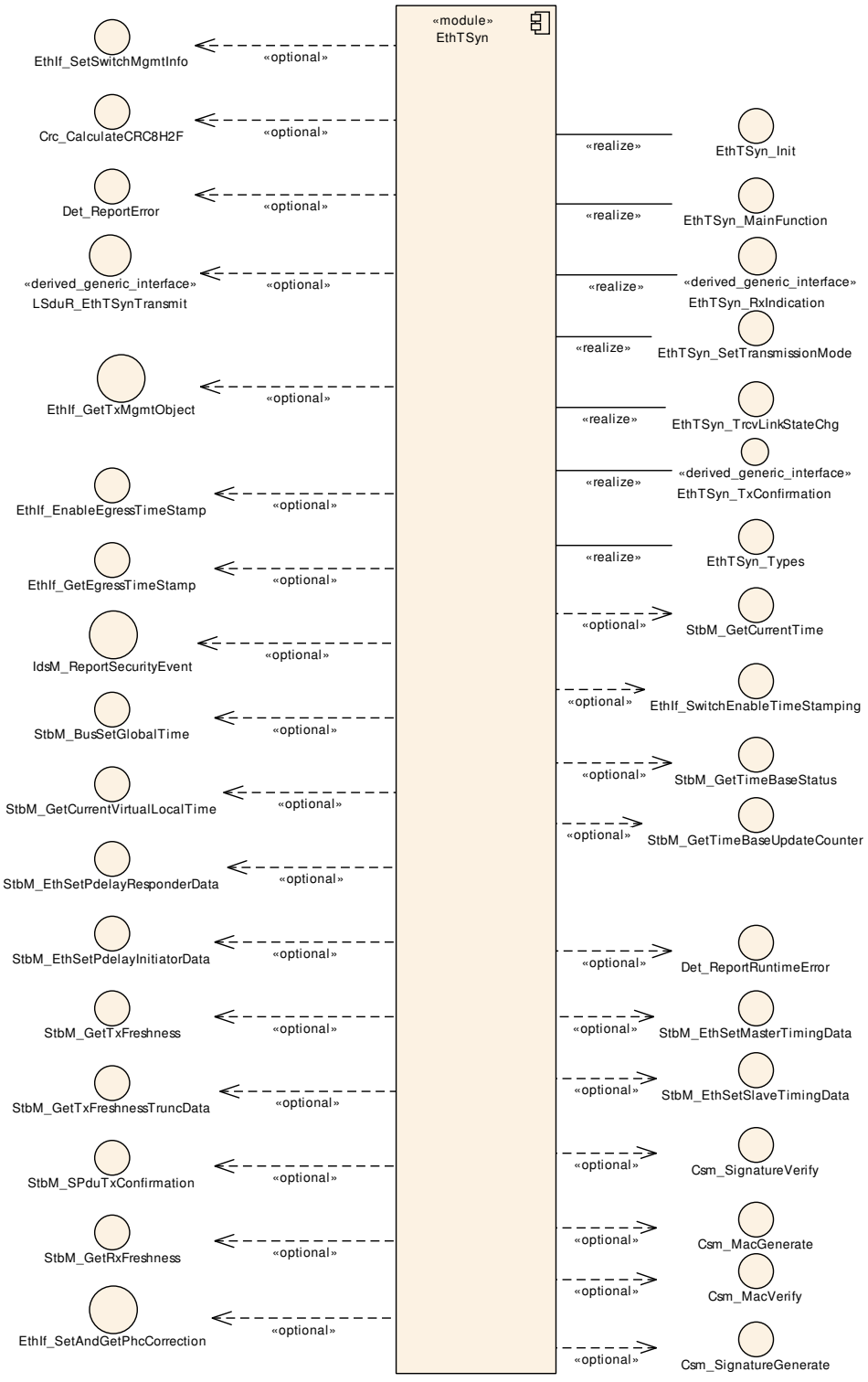


Figure 5.1: Module dependencies of the EthTSyn module

5.1 File structure

5.1.1 Code file structure

For details, refer to the section 5.1.6 "Code file structure" of the SWS BSW General [3].

6 Requirements Tracing

The following tables reference the requirements specified in [9, RS TimeSync] and [10, SRS BSW General] and links to the fulfillment of these. Please note that if column "Satisfied by" is empty for a specific requirement this means that this requirement is not fulfilled by this document.

Requirement	Description	Satisfied by
[RS_TS_00002]	Independent Maintenance of Time Base	[SWS_EthTSyn_00210]
[RS_TS_00034]	Provide Measurement Data to Applications	[SWS_EthTSyn_00212] [SWS_EthTSyn_00213] [SWS_EthTSyn_00216] [SWS_EthTSyn_00217] [SWS_EthTSyn_00218] [SWS_EthTSyn_00219] [SWS_EthTSyn_00220] [SWS_EthTSyn_00221] [SWS_EthTSyn_00222] [SWS_EthTSyn_00223]
[RS_TS_20047]	Trigger Time Base Synchronization Transmission	[SWS_EthTSyn_00130] [SWS_EthTSyn_00131] [SWS_EthTSyn_00132] [SWS_EthTSyn_00133] [SWS_EthTSyn_00134] [SWS_EthTSyn_00135] [SWS_EthTSyn_00137] [SWS_EthTSyn_00139] [SWS_EthTSyn_00187] [SWS_EthTSyn_00202] [SWS_EthTSyn_00211] [SWS_EthTSyn_00265] [SWS_EthTSyn_00400] [SWS_EthTSyn_00401] [SWS_EthTSyn_00421] [SWS_EthTSyn_00422] [SWS_EthTSyn_00423] [SWS_EthTSyn_00424]
[RS_TS_20048]	Support IEEE 802.1AS and AUTOSAR Extensions	[SWS_EthTSyn_00013] [SWS_EthTSyn_00014] [SWS_EthTSyn_00017] [SWS_EthTSyn_00019] [SWS_EthTSyn_00020] [SWS_EthTSyn_00021] [SWS_EthTSyn_00022] [SWS_EthTSyn_00031] [SWS_EthTSyn_00032] [SWS_EthTSyn_00033] [SWS_EthTSyn_00035] [SWS_EthTSyn_00039] [SWS_EthTSyn_00040] [SWS_EthTSyn_00042] [SWS_EthTSyn_00043] [SWS_EthTSyn_00044] [SWS_EthTSyn_00045] [SWS_EthTSyn_00047] [SWS_EthTSyn_00049] [SWS_EthTSyn_00052] [SWS_EthTSyn_00104] [SWS_EthTSyn_00122] [SWS_EthTSyn_00123] [SWS_EthTSyn_00124] [SWS_EthTSyn_00127] [SWS_EthTSyn_00128] [SWS_EthTSyn_00138] [SWS_EthTSyn_00159] [SWS_EthTSyn_00160] [SWS_EthTSyn_00161] [SWS_EthTSyn_00162] [SWS_EthTSyn_00180] [SWS_EthTSyn_00188] [SWS_EthTSyn_00189] [SWS_EthTSyn_00190] [SWS_EthTSyn_00200] [SWS_EthTSyn_00201] [SWS_EthTSyn_00202] [SWS_EthTSyn_00203] [SWS_EthTSyn_00204] [SWS_EthTSyn_00214] [SWS_EthTSyn_00215] [SWS_EthTSyn_00263] [SWS_EthTSyn_00264] [SWS_EthTSyn_00266] [SWS_EthTSyn_00267] [SWS_EthTSyn_00268] [SWS_EthTSyn_00412] [SWS_EthTSyn_00413]
[RS_TS_20051]	Synchronization Error Handling over Ethernet	[SWS_EthTSyn_00019] [SWS_EthTSyn_00020] [SWS_EthTSyn_00021] [SWS_EthTSyn_00022] [SWS_EthTSyn_00029] [SWS_EthTSyn_00129] [SWS_EthTSyn_00145] [SWS_EthTSyn_00146] [SWS_EthTSyn_00417] [SWS_EthTSyn_00418] [SWS_EthTSyn_00419] [SWS_EthTSyn_00420]
[RS_TS_20052]	Time Master Role Configuration over Ethernet	[SWS_EthTSyn_00051] [SWS_EthTSyn_00414] [SWS_EthTSyn_00415] [SWS_EthTSyn_00416]
[RS_TS_20053]	Time Slave Role Configuration over Ethernet	[SWS_EthTSyn_00051] [SWS_EthTSyn_00414] [SWS_EthTSyn_00415] [SWS_EthTSyn_00416]

△

Requirement	Description	Satisfied by
[RS_TS_20054]	Time Gateway Information Propagation	[SWS_EthTSyn_00051]
[RS_TS_20058]	Provide Time Synchronization Precision Data	[SWS_EthTSyn_00150]
[RS_TS_20059]	Access All Communication Ports for Time Sync	[SWS_EthTSyn_00031] [SWS_EthTSyn_00047]
[RS_TS_20061]	Protect Time Synchronization Protocol over Ethernet	[SWS_EthTSyn_00080] [SWS_EthTSyn_00096] [SWS_EthTSyn_00111]
[RS_TS_20062]	Support User-Specific Data in Time Protocol over Ethernet	[SWS_EthTSyn_00080] [SWS_EthTSyn_00230]
[RS_TS_20066]	Peer-to-Peer Delay Support over Ethernet	[SWS_EthTSyn_00200] [SWS_EthTSyn_00201] [SWS_EthTSyn_00224] [SWS_EthTSyn_00225]
[RS_TS_20069]	Provide Read/Write Access to Bus-Specific Parameters	[SWS_EthTSyn_00226] [SWS_EthTSyn_00227] [SWS_EthTSyn_00330] [SWS_EthTSyn_00331]
[RS_TS_20072]	Secure Time Synchronization Protocol over Ethernet	[SWS_EthTSyn_00104] [SWS_EthTSyn_00232] [SWS_EthTSyn_00233] [SWS_EthTSyn_00234] [SWS_EthTSyn_00236] [SWS_EthTSyn_00237] [SWS_EthTSyn_00238] [SWS_EthTSyn_00239] [SWS_EthTSyn_00240] [SWS_EthTSyn_00241] [SWS_EthTSyn_00242] [SWS_EthTSyn_00243] [SWS_EthTSyn_00244] [SWS_EthTSyn_00245] [SWS_EthTSyn_00246] [SWS_EthTSyn_00247] [SWS_EthTSyn_00248] [SWS_EthTSyn_00249] [SWS_EthTSyn_00250] [SWS_EthTSyn_00251] [SWS_EthTSyn_00252] [SWS_EthTSyn_00253] [SWS_EthTSyn_00254] [SWS_EthTSyn_00255] [SWS_EthTSyn_00256] [SWS_EthTSyn_00257] [SWS_EthTSyn_00258] [SWS_EthTSyn_00402] [SWS_EthTSyn_00403] [SWS_EthTSyn_00404] [SWS_EthTSyn_00405] [SWS_EthTSyn_00406] [SWS_EthTSyn_00407] [SWS_EthTSyn_00408] [SWS_EthTSyn_00409] [SWS_EthTSyn_00410] [SWS_EthTSyn_00411] [SWS_EthTSyn_91001] [SWS_EthTSyn_91002]
[SRS_BSW_00101]	The Basic Software Module shall be able to initialize variables and hardware in a separate initialization function	[SWS_EthTSyn_00006]
[SRS_BSW_00323]	All AUTOSAR Basic Software Modules shall check passed API parameters for validity	[SWS_EthTSyn_00029] [SWS_EthTSyn_00030] [SWS_EthTSyn_00041] [SWS_EthTSyn_00172] [SWS_EthTSyn_00174] [SWS_EthTSyn_00175] [SWS_EthTSyn_00176] [SWS_EthTSyn_00228] [SWS_EthTSyn_00229] [SWS_EthTSyn_00259] [SWS_EthTSyn_00260]
[SRS_BSW_00337]	Classification of development errors	[SWS_EthTSyn_00030] [SWS_EthTSyn_00041] [SWS_EthTSyn_00172] [SWS_EthTSyn_00174] [SWS_EthTSyn_00175] [SWS_EthTSyn_00176] [SWS_EthTSyn_00228] [SWS_EthTSyn_00229] [SWS_EthTSyn_00259] [SWS_EthTSyn_00260]
[SRS_BSW_00385]	List possible error notifications	[SWS_EthTSyn_00030] [SWS_EthTSyn_00144]
[SRS_BSW_00386]	The BSW shall specify the configuration and conditions for detecting an error	[SWS_EthTSyn_00172] [SWS_EthTSyn_00228] [SWS_EthTSyn_00229]
[SRS_BSW_00406]	API handling in uninitialized state	[SWS_EthTSyn_00030]
[SRS_BSW_00417]	Software which is not part of the SW-C shall report error events only after the Dem is fully operational.	[SWS_EthTSyn_00029]

▽

△

Requirement	Description	Satisfied by
[SRS_BSW_00452]	Classification of runtime errors	[SWS_EthTSyn_00172] [SWS_EthTSyn_00228] [SWS_EthTSyn_00229]
[SRS_BSW_00488]	Classification of security events	[SWS_EthTSyn_00261] [SWS_EthTSyn_92000] [SWS_EthTSyn_92001] [SWS_EthTSyn_92002] [SWS_EthTSyn_92003]
[SRS_BSW_00489]	Reporting of security events	[SWS_EthTSyn_00231] [SWS_EthTSyn_00261] [SWS_EthTSyn_92000] [SWS_EthTSyn_92001] [SWS_EthTSyn_92002] [SWS_EthTSyn_92003]
[SRS_BSW_00490]	List possible security events	[SWS_EthTSyn_00261]
[SRS_BSW_00491]	Specification of trigger conditions and context data	[SWS_EthTSyn_00231]
[SRS_BSW_00496]	BSW modules shall be able to handle PDU meta data	[SWS_EthTSyn_00127]

Table 6.1: Requirements Tracing

7 Functional specification

This chapter defines the behavior of the module EthTSyn, responsible for the Time Synchronization over Ethernet. The API of the module is defined in Chapter 8, while the configuration is defined in Chapter 10.

7.1 Overview

The module EthTSyn is responsible to ensure the collection and distribution of synchronized time information across the Ethernet network. It interacts with the StbM and provides all Ethernet specific functions to the StbM.

7.1.1 General

Refer to chapter General in [1, PRS Time Synchronization Over Ethernet Protocol].

7.1.2 VLAN Support

[SWS_EthTSyn_00162]

Upstream requirements: RS_TS_20048

⌈When calling `LSduR_EthTSynTransmit`, EthTSyn shall use a meta data item of type `PRIORITY_8` (referenced by the parameter `PduInfoPtr`) to set the frame priority to the value of the configuration parameter `EthTSynFramePrio`.

Use a metadata item of type `ETHERNET_MAC_64` (referenced by parameter `PduInfoPtr`) to set the destination MAC address based on the configuration parameter `EthTSynDestPhyAddr`.⌋

Refer to chapter VLAN Support in [1, PRS Time Synchronization Over Ethernet Protocol] for additional requirements.

7.1.3 Message Transmission

[SWS_EthTSyn_00417] Retry Counter for Transmission

Upstream requirements: RS_TS_20051

⌈For every transmission of messages, EthTSyn shall maintain a retry counter (refer to `EthTSynMaxNumberOfTransmitRetries`).⌋

[SWS_EthTSyn_00418] Initialization of Transmission Retry Counter

Upstream requirements: RS_TS_20051

⌈Upon the initial processing of messages, the retry counter shall be set to 0.⌋

[SWS_EthTSyn_00419] Incrementing the Transmission Retry Counter

Upstream requirements: RS_TS_20051

⌈If `LSduR_EthTSynTransmit` returns a value different than E_OK, the retry counter shall be incremented, and the message transmission shall be retried in the next call of `EthTSyn_MainFunction`.⌋

[SWS_EthTSyn_00420] Stop of retrying transmissions

Upstream requirements: RS_TS_20051

⌈If the retry counter has reached the configuration value `EthTSynMaxNumberOfTransmitRetries`, the EthTSyn shall stop retrying and resume cyclic transmission.⌋

7.2 Initialization

The EthTSyn is initialized via `EthTSyn_Init`. Except for `EthTSyn_Init`, the API functions of the EthTSyn module may only be called when the module has been properly initialized.

[SWS_EthTSyn_00006]

Upstream requirements: SRS_BSW_00101

⌈A call to `EthTSyn_Init` initializes all internal variables and sets the EthTSyn module to the initialized state.⌋

Note: Unless specified otherwise EthTSyn uses default values as given in [4, IEEE Std 802.1AS-2011].

7.3 Handling of different Virtual Local Time sources

If HW Timestamping is enabled, the StbM could also use the ETH free running counter for interpolation of the local instance of the Global Time. There are however use cases when the StbM is configured to use the GPT instead, e.g.

- A Global Time Master or a Time Gateway is connected to different CAN/ETH busses and HW timestamping of each CAN/ETH communication controller is unsynchronized with each other.

In such a case conversions are required between the timestamps of different Virtual Local Time sources:

- The StbM uses (i.e., captures, stores and returns) only timestamps in the scope of its Virtual Local Time source.
- <Bus>TSyn modules thus need to convert timestamps from their Virtual Local Time source to the scope of the StbM's Virtual Local Time source in case different scopes are used when either passing a global time to the StbM or when obtaining it from the StbM (refer to alternative label "Time Source of StbM" in sequence diagrams Figure 9.4 and Figure 9.5).
- The conversion can happen linearly, i.e., no rate correction terms need to be determined and applied.

[SWS_EthTSyn_00210]

Upstream requirements: RS_TS_00002

⌈EthTSyn shall discard a timestamp derived from the Ethernet Controller HW (e.g., via `EthIf_GetCurrentTimeTuple` or `EthIf_GetEgressTimeStamp`), if the quality of the timestamp (refer to `TimeStampQualType`) is indicated as `INVALID` or `UNCERTAIN`.⌋

7.4 Virtual Free-Running Clock

If the PHC of an Ethernet controller is used as timestamping clock and the PHC is corrected in offset and rate (see `EthTSyn_SetAndGetPhcCorrection`), the timestamps taken by the hardware are no longer based on a free-running clock. A PHC correction that is performed while a measurement (e.g. a Pdelay measurement) is ongoing would render calculations based on such timestamps invalid, because the involved timestamps would refer to different states of the clock.

To avoid such inconsistencies, EthTSyn maintains a virtual free-running clock (VFRC). The VFRC represents the time of the PHC as if no PHC correction had ever been applied, i.e. it is a bookkeeping of all performed PHC corrections. All calculations that require an elapsed time between two timestamps are performed based on the VFRC.

[SWS_EthTSyn_00425] Maintenance of a virtual free-running clock

Status: DRAFT

⌈If a PHC is used as timestamping clock of a Time Domain, EthTSyn shall maintain a virtual free-running clock (VFRC) for this timestamping clock. The VFRC shall be based on the initial value of the PHC and shall increase monotonically, unaffected by any PHC correction.⌋

[SWS_EthTSyn_00426] PHC correction trace

Status: DRAFT

⌈EthTSyn shall maintain a PHC correction trace containing the three most recent PHC corrections. Each trace entry shall contain the following information:

- the time of the PHC directly before the correction was applied (refer to `PreCorrectionTime`)
- the time of the VFRC belonging to the time of the PHC directly before the correction was applied
- the applied offset correction (jump correction) of the PHC
- the resulting total rate of the PHC after the correction (refer to `ResultingPhcRate`)

⌋

[SWS_EthTSyn_00427] Update of the PHC correction trace

Status: DRAFT

⌈Upon each successfully performed PHC correction via `EthTSyn_SetAndGetPhcCorrection`, EthTSyn shall update the PHC correction trace as follows:

1. Discard the oldest trace entry if the trace is full.
2. Calculate the VFRC time of the new trace entry as $VFRC_{new} = VFRC_{prev} + (T_{preCorr} - T_{postCorr,prev}) \cdot R_{prev}$, where $VFRC_{prev}$ is the VFRC time of the most recent trace entry, $T_{preCorr}$ is the PHC time directly before the new correction, $T_{postCorr,prev}$ is the PHC time directly after the most recent correction, and R_{prev} is the total rate of the PHC after the most recent correction.
3. Add the new PHC correction as most recent trace entry.

⌋

[SWS_EthTSyn_00428] Conversion of a PHC timestamp into the VFRC time

Status: DRAFT

⌈To convert a PHC timestamp into the corresponding VFRC time, EthTSyn shall perform the following steps:

1. Determine the most recent PHC correction trace entry that was recorded before the provided PHC timestamp was taken.
2. Calculate the time passed between the determined PHC correction trace entry and the provided PHC timestamp, considering the offset correction of the determined trace entry.
3. Apply the rate of the determined trace entry to the calculated delta. This results in the corresponding VFRC delta.
4. Add the VFRC delta to the VFRC time of the determined trace entry.

⌋

[SWS_EthTSyn_00429] Invalidity of PHC timestamps older than the PHC correction trace

Status: DRAFT

⌈EthTSyn shall consider a PHC timestamp as invalid and discard it, if the timestamp is older than the PHC time of the oldest available PHC correction trace entry.⌋

Rationale: for timestamps older than the oldest trace entry it is unknown which corrections were applied to the PHC in the past, hence a conversion into the VFRC is not possible.

[SWS_EthTSyn_00430] Invalidity of ambiguous PHC timestamps

Status: DRAFT

⌈EthTSyn shall consider a PHC timestamp as invalid and discard it, if the timestamp cannot be unambiguously assigned to the time before or after a backward jump correction of the PHC.⌋

Note: with a backward jump correction, PHC times within the corrected offset occur twice. A timestamp within such a period can only be assigned unambiguously, if additional information (e.g. the current PHC time or the maximal processing delay of a timestamp) allows to exclude one of the two possible assignments.

Note: three PHC correction trace entries are sufficient to convert any valid PHC timestamp into the VFRC under the following assumptions:

- PHC corrections, especially jump corrections, are performed with a low frequency compared to the processing of timestamps. On a Time Slave, PHC corrections are triggered upon reception of new time information (refer to `StbM_BusSetGlobalTime`) or upon expiry of the adaption interval for offset correction by rate adaption. On a Time Master, PHC corrections are triggered by the application, typically with a low frequency.
- Timestamps are processed shortly after they are captured, i.e. at most one jump correction can occur between capturing and processing of a timestamp.
- No old or outdated timestamps are provided to EthTSyn.

[SWS_EthTSyn_00431] Provision of the VFRC time via EthTSyn_GetCurrentTimeTuple

Status: DRAFT

⌈If a PHC is used as timestamping clock of the Time Domain for which `EthTSyn_GetCurrentTimeTuple` is called, EthTSyn shall set the element `timestampClockValue` of the returned time tuple to the current PHC time converted into the VFRC time (refer to [SWS_EthTSyn_00428]). Otherwise, EthTSyn shall set `timestampClockValue` to the current value of the timestamping clock. In all cases, EthTSyn shall set the element `disciplinedClockValue` to the current value of the disciplined clock of the clock unit.⌋

[SWS_EthTSyn_00432] Rejection of EthTSyn_GetCurrentTimeTuple upon failed VFRC conversion

Status: DRAFT

⌈If a PHC is used as timestamping clock of the Time Domain for which `EthTSyn_GetCurrentTimeTuple` is called and the current PHC time cannot be converted into the VFRC time (refer to [SWS_EthTSyn_00429] and [SWS_EthTSyn_00430]), EthTSyn shall return `E_NOT_OK`.⌋

7.5 Debounce Time

[SWS_EthTSyn_00130]

Upstream requirements: RS_TS_20047

⌈If `EthTSynGlobalTimeDebounceTime` is set to 0, EthTSyn shall ignore any debouncing.⌋

[SWS_EthTSyn_00131]

Upstream requirements: RS_TS_20047

⌈If `EthTSynGlobalTimeDebounceTime` is greater than 0, EthTSyn shall always consider debouncing for all Timesync PDUs (Sync, Follow_Up, Pdelay_Req, Pdelay_Resp and Pdelay_Resp_Follow_Up) as described below.
⌋

Note: The Debouncing avoids misassignment of time stamps to false event message.

[SWS_EthTSyn_00132]

Upstream requirements: RS_TS_20047

⌈`EthTSynGlobalTimeDebounceTime` represents the reload value of a `debounceCounter` that shall be reloaded at that point in time, where a Timesync PDU has been sent and that shall be decremented on each `EthTSyn_MainFunction` call if no Timesync PDU is transmitted.
⌋

[SWS_EthTSyn_00133]

Upstream requirements: RS_TS_20047

⌈A new Timesync PDU shall only be sent, if the corresponding `debounceCounter` has reached 0.⌋

[SWS_EthTSyn_00187]

Upstream requirements: RS_TS_20047

⌈Each `EthTSynPTPPort` of a `EthTSynGlobalTimeDomain` shall have its own `debounceCounter`.⌋

7.6 Pdelay Protocol

This chapter defines EthTSyn specific requirements in addition to the generic requirements in chapter "Pdelay Protocol for Latency Calculation" in [1, PRS Time Synchronization Over Ethernet Protocol].

The Pdelay protocol is executed:

- For every Domain for which `EthTSynNeighbourRateRatioConfig` is set to `TransportSpecific` or `EthTSynLinkDelayConfig` is set to `TransportSpecific`
- Only once for multiple Domains, if `EthTSynNeighbourRateRatioConfig` is set to `CMLDS` or `EthTSynLinkDelayConfig` is set to `CMLDS`

The Pdelay protocol is not executed, if neither of the above conditions are met.

The configuration restrictions of Chapter 10.4 “Configuration Constraints” apply.

The overall sequence of actions for the Pdelay measurement are given in Figure 9.3.

7.6.1 Pdelay Message Transmission

The detailed sequences of actions for the transmission of

- the Pdelay_Req message
- the Pdelay_Resp message and
- the Pdelay_Resp_Follow_Up message

are given in Figure 9.4.

[SWS_EthTSyn_00200]

Upstream requirements: RS_TS_20048, RS_TS_20066

⌈If the Pdelay Initiator transmits a Pdelay_Req for latency calculation with the cycle (refer to PRS_TS_00011 in [1, PRS Time Synchronization Over Ethernet Protocol]), the following sequence shall be applied:

1. Activate the time stamping via `EthIf_EnableEgressTimeStamp` if `EthTSynHardwareTimestampSupport` is set to TRUE
2. Transmit the Pdelay_Req message by calling `LSduR_EthTSynTransmit` with
   - parameter `TxPduId` derived from the configuration parameter `EthTSynTxPduRef` of the corresponding port
   - and parameter `SduDataPtr` set to the actual Pdelay_Req message data
   - and parameter `SduLength` set to length of the Pdelay_Req message data

⌋

[SWS_EthTSyn_00201]

Upstream requirements: RS_TS_20048, RS_TS_20066

⌈If the Pdelay Responder transmits a Pdelay_Resp for latency calculation (refer to PRS_TS_00012) in [1, PRS Time Synchronization Over Ethernet Protocol]) the following sequence shall be applied:

1. Activate the time stamping via `EthIf_EnableEgressTimeStamp` if `EthTSynHardwareTimestampSupport` is set to TRUE
2. Transmit the Pdelay_Resp message by calling `LSduR_EthTSynTransmit` with
   - parameter `TxPduId` derived from the configuration parameter `EthTSynTxPduRef` of the corresponding port
   - and parameter `SduDataPtr` set to the actual Pdelay_Resp message data
   - and parameter `SduLength` set to length of the Pdelay_Resp message data

⌋

[SWS_EthTSyn_00013]

Upstream requirements: RS_TS_20048

⌈On invocation of `EthTSyn_TxConfirmation` with parameter `Result` equal to `E_OK` the egress time stamp shall be retrieved for t1 from the EthIf via `EthIf_GetEgressTimeStamp` on egress of the Pdelay_Req message, if `EthTSynHardwareTimestampSupport` is set to TRUE.

If the StbM does not use the Ethernet controller as source for the Virtual Local Time (refer to parameter `StbMLocalTimeHardware`, in [2]), the EthTSyn shall convert the egress time stamp to the Virtual Local Time as used in the StbM.⌋

[SWS_EthTSyn_00123]

Upstream requirements: RS_TS_20048

⌈On invocation of `EthTSyn_TxConfirmation` with parameter `Result` equal to `E_OK` the egress time stamp shall be retrieved for t1 from the StbM via `StbM_GetCurrentVirtualLocalTime` on egress of the Pdelay_Req message, if `EthTSynHardwareTimestampSupport` is set to FALSE.⌋

[SWS_EthTSyn_00159]

Upstream requirements: RS_TS_20048

⌈On invocation of `EthTSyn_TxConfirmation` with parameter `Result` equal to `E_OK` the egress timestamp shall be retrieved for t3 from the EthIf via `EthIf_EnableEgressTimeStamp` on egress of the Pdelay_Resp message, if `EthTSynHardwareTimestampSupport` is set to TRUE.

If the StbM does not use the Ethernet controller as source for the Virtual Local Time (refer to parameter `StbMLocalTimeHardware`, in [2]), the EthTSyn shall convert the egress time stamp to the Virtual Local Time as used in the StbM.⌋

[SWS_EthTSyn_00122]

Upstream requirements: RS_TS_20048

⌈On invocation of EthTSyn_TxConfirmation with parameter Result equal to E_OK the egress timestamp shall be retrieved for t3 from the StbM via StbM_GetCurrentVirtualLocalTime on egress of Pdelay_Resp message, if EthTSynHardwareTimestampSupport is set to FALSE.⌋

[SWS_EthTSyn_00225]

Upstream requirements: RS_TS_20066

⌈The Time Master shall set responseOriginTimestamp (for the Pdelay_Resp_Follow_Up message) to t3.⌋

[SWS_EthTSyn_00014]

Upstream requirements: RS_TS_20048

⌈If EthTSynGlobalTimePdelayRespEnable is set to TRUE, a Pdelay Responder shall transmit a Pdelay_Resp_Follow_Up with the transmission timestamp of that message as defined in [SWS_EthTSyn_00159] as well as defined in [1, PRS Time Synchronization Over Ethernet Protocol] [1, PRS Time Synchronization Over Ethernet Protocol] Chapter 4.6.1 *"Pdelay Protocol for Latency Calculation"* considering debounceCounter which represents a time offset between Pdelay_Resp and Pdelay_Resp_Follow_Up.

For that, the following sequence shall be applied:

1. Transmit the Pdelay_Resp_Follow_Up message by calling LSduR_EthTSynTransmit with
   - parameter TxPduId derived from the configuration parameter EthTSynRxPduRef of the corresponding port
   - and parameter SduDataPtr set to the actual Pdelay_Resp_Follow_Up message data including the transmission timestamp of [SWS_EthTSyn_00159]
   - and parameter SduLength set to length of the Pdelay_Resp_Follow_Up message data

⌋

7.6.2 Pdelay Message Reception

The detailed sequences of actions for the reception of

- the Pdelay_Req message
- the Pdelay_Resp message and
- the Pdelay_Resp_Follow_Up message

are given in sequence diagram Figure 9.5.

[SWS_EthTSyn_00160]

Upstream requirements: RS_TS_20048

⌈If `EthTSynHardwareTimestampSupport` is set to TRUE, when `EthTSyn_RxIndication` is called on ingress of the Pdelay_Req message, then the EthTSyn shall retrieve the ingress timestamp t2 from the meta data item of type `TIMETUPLE_TYPE_PTR` of the PDU identified by parameter `RxPduId`.

If the StbM does not use the Ethernet controller as source for the Virtual Local Time (refer to parameter `StbMLocalTimeHardware` in [2]), the EthTSyn shall convert the ingress time stamp to the Virtual Local Time as used in the StbM.⌋

[SWS_EthTSyn_00124]

Upstream requirements: RS_TS_20048

⌈If `EthTSynHardwareTimestampSupport` is set to FALSE, when `EthTSyn_RxIndication` is called on ingress of Pdelay_Req message, then the EthTSyn shall retrieve the ingress timestamp t2 from the StbM via `StbM_GetCurrentVirtualLocalTime`.⌋

[SWS_EthTSyn_00224]

Upstream requirements: RS_TS_20066

⌈The Time Master shall set `requestReceiptTimestamp` (to be used in the Pdelay_Resp message) to t2.⌋

[SWS_EthTSyn_00049]

Upstream requirements: RS_TS_20048

⌈If `EthTSynHardwareTimestampSupport` is set to TRUE, when `EthTSyn_RxIndication` is called on ingress of the Pdelay_Resp message, then the EthTSyn shall retrieve the ingress time stamp t4 from the meta data item of type `TIMETUPLE_TYPE_PTR` of the PDU identified by parameter `RxPduId`.

If the StbM does not use the Ethernet controller as source for the Virtual Local Time (refer to parameter `StbMLocalTimeHardware` in [2]), the EthTSyn shall convert the ingress time stamp to the Virtual Local Time as used in the StbM.⌋

[SWS_EthTSyn_00161]

Upstream requirements: RS_TS_20048

⌈If `EthTSynHardwareTimestampSupport` is set to FALSE, when `EthTSyn_RxIndication` is called on ingress of the Pdelay_Resp message, then the EthTSyn shall retrieve the ingress time stamp t4 from the StbM via `StbM_GetCurrentVirtualLocalTime`.⌋

[SWS_EthTSyn_00263]
Status: DRAFT
Upstream requirements: RS_TS_20048

⌈If

- configuration parameter `EthTSynNeighbourRateRatioConfig` is set to `CMLDS` or `TransportSpecific`
- and `EthTSynRateRatioMeasurementCount` consecutive Pdelay measurements have been completed successfully,

then EthTSyn shall calculate in the next main function call the `neighborRateRatio` as given in PRS_TS_00259 in [1, PRS Time Synchronization Over Ethernet Protocol].⌋

[SWS_EthTSyn_00264]
Status: DRAFT
Upstream requirements: RS_TS_20048

⌈If

- configuration parameter `EthTSynGlobalTimeTxPdelayReqPeriod` is not 0
- and a valid `Pdelay_Resp_Follow_Up` message has been received,

then EthTSyn shall calculate the value `linkDelay` in the next main function call as given in `PRS_TS_00003` in [1, PRS Time Synchronization Over Ethernet Protocol].

⌋

7.6.3 Pdelay measurements with PHC correction

If a PHC is used as timestamping clock and the PHC is corrected while a Pdelay measurement is ongoing, the timestamps of the measurement refer to different states of the clock. To keep the calculated values valid, all involved timestamps are converted into the VFRC (see Chapter 7.4).

[SWS_EthTSyn_00433] Pdelay calculation based on the VFRC
Status: DRAFT

⌈If a PHC is used as timestamping clock of the Time Domain, EthTSyn shall convert the timestamps t1, t2, t3, and t4 into the corresponding VFRC time (refer to [SWS_EthTSyn_00428]) before calculating `linkDelay` or `neighborRateRatio`.⌋

[SWS_EthTSyn_00434] Pdelay responder timestamps derived from the VFRC
Status: DRAFT

⌈If a PHC is used as timestamping clock of the Time Domain, the Pdelay responder shall derive `requestReceiptTimestamp` and `responseOriginTimestamp` from the VFRC time of the corresponding ingress and egress timestamps (refer to [SWS_

EthTSyn_00428]), so that the Pdelay initiator calculates a remote residence time that is not influenced by any PHC correction.⌋

[SWS_EthTSyn_00435] Discard of Pdelay measurements upon failed VFRC conversion

Status: DRAFT

⌈If a timestamp belonging to an ongoing Pdelay measurement cannot be converted into the VFRC time (refer to [SWS_EthTSyn_00429] and [SWS_EthTSyn_00430]), Eth TSyn shall discard the affected Pdelay measurement.⌋

[SWS_EthTSyn_00436] Rate conversion between VFRC and PHC

Status: DRAFT

⌈If a PHC is used as timestamping clock of the Time Domain, EthTSyn shall convert a rate calculated on basis of the VFRC into the relative rate of the PHC by considering the current total rate of the PHC (refer to `ResultingPhcRate`), whenever the rate is applied as PHC correction.⌋

Note: rates that are transmitted (e.g. the `cumulativeScaledRateOffset`) always represent the absolute rate, which is calculated on basis of the VFRC and needs no conversion. The conversion into the relative rate of the PHC is only required when a rate is applied as PHC correction, because the requested rate deviation relates to the current, already corrected rate of the PHC.

7.7 Message Format

Refer to chapter Message format in [1, PRS Time Synchronization Over Ethernet Protocol] for additional requirements.

7.7.1 Sync and Follow_Up acc. to IEEE 802.1AS

Refer to chapter Sync and Follow_Up acc. to IEEE 802.1AS in [1, PRS Time Synchronization Over Ethernet Protocol].

7.7.2 Sync and Follow_Up acc. to AUTOSAR

Refer to chapter Sync and Follow_Up acc. to AUTOSAR in [1, PRS Time Synchronization Over Ethernet Protocol].

7.7.2.1 Follow_Up Message Header [AUTOSAR]

Refer to chapter Follow_Up Message Header [AUTOSAR] in [1, PRS Time Synchronization Over Ethernet Protocol].

7.7.2.2 AUTOSAR and OEM Sub-TLV's

Refer to chapter AUTOSAR and OEM Sub-TLVs in [1, PRS Time Synchronization Over Ethernet Protocol].

7.7.2.2.1 AUTOSAR Sub-TLV: Time Secured

Refer to chapter AUTOSAR Sub-TLV: Time Secured in [1, PRS Time Synchronization Over Ethernet Protocol].

7.7.2.2.2 AUTOSAR Sub-TLV: Status Secured / Not Secured

Refer to chapter AUTOSAR Sub-TLV: Status Secured in [1, PRS Time Synchronization Over Ethernet Protocol].

7.7.2.2.3 AUTOSAR Sub-TLV: UserData Secured / Not Secured

[SWS_EthTSyn_00080]

Upstream requirements: RS_TS_20061, RS_TS_20062

⌈The AUTOSAR Sub-TLV: UserData shall be mapped to the `StbM_UserDataType`, whereas the User Byte number given in the message and by the `StbM_UserDataType` shall match (`UserByte_0` mapped to `StbM_UserDataType.userByte0` etc.).

The `UserDataLength` shall be mapped to `StbM_UserDataType.userDataLength` and vice versa.⌋

Refer to chapter AUTOSAR Sub-TLV: UserData Secured / Not Secured in [1, PRS Time Synchronization Over Ethernet Protocol] for additional requirements.

7.7.2.2.4 AUTOSAR Sub-TLV: Time Authenticated

Refer to chapter AUTOSAR Sub-TLV: Time Authenticated in [1, PRS Time Synchronization Over Ethernet Protocol].

7.7.2.3 Pdelay Message Header

Refer to [1, PRS Time Synchronization over Ethernet Protocol] Chapter 4.3.1.1 *"Sync and Follow_Up acc. to IEEE 802.1AS"* for the general header structure.

If `EthTSynNeighbourRateRatioConfig` or `EthTSynLinkDelayConfig` are configured to `TransportSpecific`:

- Either compliant with [5, IEEE Std 802.1AS-2020] using `domainNumber 0x00`,
- or Autosar specific using the lowest `domainNumber`,
- or with all configured `domainNumber`s for shared media like 10BASE-T1S.

If `EthTSynNeighbourRateRatioConfig` or `EthTSynLinkDelayConfig` is configured to `CMLDS`, the `majorSdoId` shall be transmitted as `0x2` and the `minorSdoId` shall be transmitted as `0x00`. As a result, the aggregation of the the previous transportSpecific, message-type, and reserved fields (now `SdoId`) for CMLDS is `0x200`.

The `domainNumber` shall be transmitted as `0x00` for all CMLDS Pdelay messages.

7.8 Acting as Time Master

Refer to chapter Acting as Time Master in [1, PRS Time Synchronization Over Ethernet Protocol] for additional requirements.

If the `EthTSyn` has at least one `EthTSynPTPPort` configured in the `EthTSynPortRole TIME_MASTER`, the `EthTSyn` module checks on each `EthTSyn_MainFunction` call the necessity for a Timesync message transmission for the respective Time Domains.

Figure 7.1 illustrates the flow for the Time Master to trigger a (immediate and cyclic) message transmission of a Timesync message.

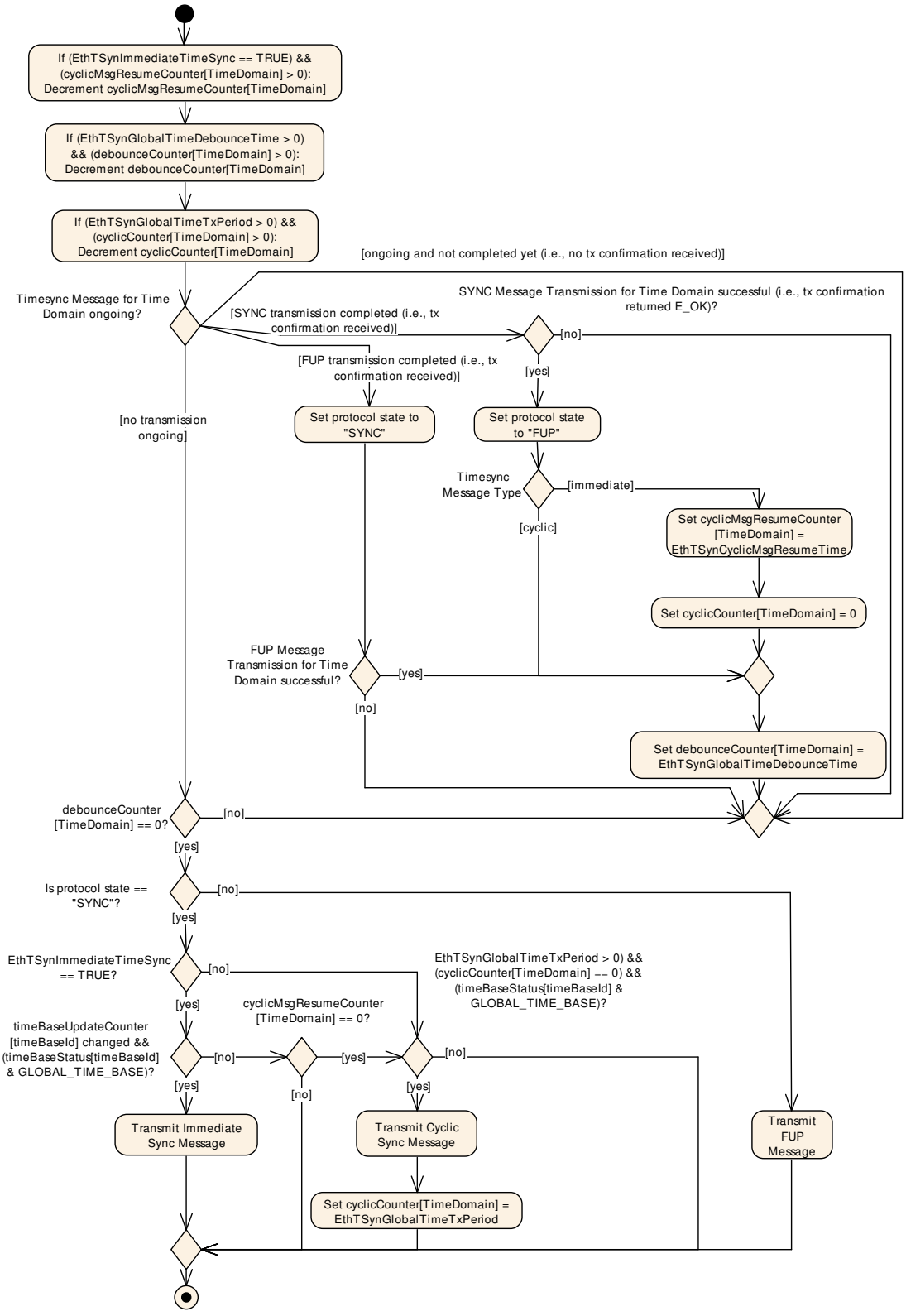


Figure 7.1: Timesync Message Transmission

7.8.1 Message processing

[SWS_EthTSyn_00265]
Status: DRAFT
Upstream requirements: RS_TS_20047

⌈When `EthTSynPortRole` is set to `DYNAMIC` or `TIME_MASTER`, ports (for eth controller only 1 port and for switch individual port) shall transmit Sync and `Follow_Up` message according to the configuration parameter `EthTSynGlobalTimeTxPeriod`.⌋

Note: For [SWS_EthTSyn_00265] Immediate Synchronization is not further affected by the port-specific `EthTSynGlobalTimeTxPeriod`.

Refer to chapter Message Processing in [1, PRS Time Synchronization Over Ethernet Protocol] for additional requirements.

[SWS_EthTSyn_00202]
Upstream requirements: RS_TS_20047, RS_TS_20048

⌈If the Time Master transmits a Sync message (refer to [PRS_TS_00016] in [1, PRS Time Synchronization Over Ethernet Protocol]), the following sequence shall be applied:

- Retrieve the Global Time Tuple `[T0;T0`$_{VLT}$`]` from the StbM via `StbM_GetCurrentTime` according to Figure 9.2.
- Activate the time stamping via `EthIf_EnableEgressTimeStamp`, if `EthTSynHardwareTimestampSupport` is set to `TRUE`
- Transmit the Pdelay_Resp_Follow_Up message by calling `LSduR_EthTSynTransmit` with
  - parameter `TxPduId` derived from the configuration parameter `EthTSynTxPduRef` of the corresponding port
  - and parameter `SduDataPtr` set to the actual Sync message data
  - and parameter `SduLength` set to the length of the Sync message data

⌋

Note: The timeBaseStatus can be read from StbM by `StbM_GetTimeBaseStatus` or `StbM_GetCurrentTime`.

Note: For further details refer to sequence diagram Figure 9.4.

[SWS_EthTSyn_00211]
Upstream requirements: RS_TS_20047

⌈If

- the protocol requirement [PRS_TS_00016] is fulfilled,

- and the associated cyclicMsgResumeCounter is equal to or less than 0

the Time Master shall start cyclic transmission of Sync messages in the earliest possible EthTSyn_MainFunction call.⌋

Note: "earliest possible" means:

- In the next EthTSyn_MainFunction, because GLOBAL_TIME_BASE is set outside the EthTSyn_MainFunction.
- In the current EthTSyn_MainFunction, when switching from immediate to cyclic transmission (because this decision is made inside the EthTSyn_MainFunction).

[SWS_EthTSyn_00127]

Upstream requirements: RS_TS_20048, SRS_BSW_00496

⌈On invocation of EthTSyn_TxConfirmation with parameter 'Result' equal to E_OK the egress time stamp of the Sync message shall be retrieved via EthIf_GetEgressTimeStamp from the EthIf and converted to the Virtual Local Time $T2_{VLT}$ according to EthTSyn_Egress_Time_Stamping, if EthTSynHardwareTimestampSupport is set to TRUE.

⌋

Note: EthTSyn_Egress_Time_Stamping is shown in Figure 9.4.

[SWS_EthTSyn_00017]

Upstream requirements: RS_TS_20048

⌈If EthTSynHardwareTimestampSupport is set to TRUE and if the StbM does not use the Ethernet hardware counter as Virtual Local Time Source for the Time Base, the following sequence shall be applied on invocation of EthTSyn_TxConfirmation with parameter 'Result' equal to E_OK or in the following EthTSyn_MainFunction call:

1. Protect the following two steps against interruptions:
2. the current time of the Ethernet hardware counter shall be retrieved via parameter currentTimeTuplePtr->timestampClockValue of EthIf_GetCurrentTimeTuple from theEthIf and converted to the Virtual Local Time $T3_{VLT}$.
3. the current value of the Virtual Local Time of the Time Base shall be retrieved as $T4_{VLT}$ via StbM_GetCurrentVirtualLocalTime
4. the preciseOriginTimestamp shall be calculated as $T0 - (T3_{VLT} - T2_{VLT}) + (T4_{VLT} - T0_{VLT})$

⌋

Note: When using interrupt mode with interrupt nesting disabled, the EthTSyn does not need to explicitly establish a protection against interruptions in EthTSyn_TxConfirmation, because this is implicitly done by the controller.

[SWS_EthTSyn_00188]

Upstream requirements: RS_TS_20048

⌈If EthTSynHardwareTimestampSupport is set to TRUE and if the StbM does use the Ethernet hardware counter as Virtual Local Time Source for the Time Base, the preciseOriginTimestamp shall be calculated as $T0 + (T2_{VLT} - T0_{VLT})$.⌋

[SWS_EthTSyn_00189]

Upstream requirements: RS_TS_20048

⌈If EthTSynHardwareTimestampSupport is set to FALSE the preciseOriginTimestamp shall be calculated as $T0 + (T4_{VLT} - T0_{VLT})$.⌋

[SWS_EthTSyn_00204]

Upstream requirements: RS_TS_20048

⌈The Time Master shall consider the debounceCounter, which represents a time offset between Sync and Follow_Up message, before transmitting the Follow_Up message.⌋

[SWS_EthTSyn_00226]

Upstream requirements: RS_TS_20069

⌈The following parameters provided by the invocation of EthTSyn_SetProtocolParam in argument protocolParam, shall be used by EthTSyn for the next Follow_Up information TLV message:

- cumulativeScaledRateOffset
- gmTimeBaseIndicator
- lastGmPhaseChange
- scaledLastGmFreqChange

⌋

[SWS_EthTSyn_00203]

Upstream requirements: RS_TS_20048

⌈The Time Master shall transmit a Follow_Up message (refer to [PRS_TS_00018] in [1, PRS Time Synchronization Over Ethernet Protocol]), by calling LSduR_EthTSynTransmit with

- parameter TxPduId derived from the configuration parameter EthTSynTxPdu of the corresponding port

- and parameter SduDataPtr set to the Follow_Up message data including the calculated preciseOriginTimestamp
- and parameter SduLength set to length of the Follow_Up message data

⌋

7.8.1.1 Runtime Error detection

[SWS_EthTSyn_00145]

Upstream requirements: RS_TS_20051

⌈If EthTSynMasterSlaveConflictDetection is set to TRUE and if the Time Master receives a Sync message from another Time Master, it shall report a runtime error by calling Det_ReportRuntimeError with error code ETHTSYN_E_TMCONFLICT and discard the received Sync message.⌋

7.8.1.2 Frame Debouncing

The EthTSyn debounces Tx PDUs of a Time Master to avoid bursts of Timesync messages on the network (e.g. if immediate transmission is enabled).

For each Tx PDU the EthTSyn maintains a debounceCounter. On each transmission of a Timesync message the debounce counter is (re-)loaded by the configured debounce time EthTSynGlobalTimeDebounceTime. The counter is decremented in each EthTSyn main cycle. Transmission of the same PDU can only be triggered, if the debounce counter has reached the value 0. Refer also to the overall sequence for the Timesync message transmission in Figure 7.1.

The EthTSyn does not support sharing of PDUs across domains and busses, i.e. same PDU ID shall not be used for different time domains.

[SWS_EthTSyn_00421] Setting the Debounce Counter After Successful Timesync PDU Transmission

Upstream requirements: RS_TS_20047

⌈If for a Time Domain

- EthTSynGlobalTimeDebounceTime is greater than 0
- and the corresponding Timesync PDU has been successfully sent,

then the EthTSyn shall set the PDU specific debounceCounter to EthTSynGlobalTimeDebounceTime.⌋

Note: A Timesync PDU is considered to be successfully sent, if a transmission confirmation for the PDU has been received.

[SWS_EthTSyn_00422] Periodic Decrement of Debounce Counter in EthTSyn Main Function

Upstream requirements: RS_TS_20047

⌈If for a Time Domain

- EthTSynGlobalTimeDebounceTime is greater than 0
- and the debounceCounter for the corresponding Timesync PDU is greater than 0,

then the EthTSyn shall decrement the debounceCounter value by EthTSynMainFunctionPeriod on each invocation of EthTSyn_MainFunction.⌋

[SWS_EthTSyn_00423] Deferring Timesync Transmission While Debounce Counter Is Active

Upstream requirements: RS_TS_20047

⌈If for a Time Domain

- EthTSynGlobalTimeDebounceTime is greater than 0
- and the debounceCounter for the corresponding Timesync PDU is greater than 0
- and a transmission of a TimeSync message is requested,

then EthTSyn shall defer the actual transmission of the Timesync message until debounceCounter is equal or less than 0.⌋

Rationale: While debouncing, a new transmission request should not get lost.

[SWS_EthTSyn_00424] Replacing Pending Timesync Transmission Request with New Request During Debounce

Upstream requirements: RS_TS_20047

⌈If for a Time Domain

- EthTSynGlobalTimeDebounceTime is greater than 0
- a SYNC transmission request is pending
- and a new immediate or cyclic transmission of a SYNC message is requested for the same PDU,

then the EthTSyn shall discard the pending request for that Time Domain.⌋

Rationale: While debouncing, there is no queuing of multiple transmission requests. The latest request is the best one.

Refer to chapter Frame Debouncing in [1, PRS Time Synchronization Over Ethernet Protocol].

7.8.1.3 Immediate Time Synchronization

In addition to the standard cyclic message transmission an immediate message transmission might be required. Depending on configuration, the EthTSyn module checks on each EthTSyn_MainFunction call the necessity for a Timesync message transmission for each Time Base, where a Master Port belongs to.

Figure 7.1 illustrates how immediate and cyclic message transmission align.

[SWS_EthTSyn_00134]

Upstream requirements: RS_TS_20047

⌈If EthTSynImmediateTimeSync is set to TRUE, EthTSyn shall check within each EthTSyn_MainFunction call by calling StbM_GetTimeBaseUpdateCounter if the returned timeBaseUpdateCounter has been changed.⌋

[SWS_EthTSyn_00135]

Upstream requirements: RS_TS_20047

⌈If

- EthTSynImmediateTimeSync is set to TRUE
- and the timeBaseUpdateCounter[timeBaseId] for the updated Time Base resp. timeBaseId has been changed
- and the GLOBAL_TIME_BASE bit within the timeBaseStatus, which is read from StbM, is set,

EthTSyn shall trigger an immediate transmission of Time Synchronization messages belonging to this Time Base.⌋

Note: The timeBaseStatus can be read from StbM by StbM_GetTimeBaseStatus or StbM_GetCurrentTime.

The debounceCounter as described in Chapter 7.5 has always to be considered.

In addition to the actual trigger condition for an immediate transmission (refer to [SWS_EthTSyn_00135] above) the parameter EthTSynCyclicMsgResumeTime needs to be considered for immediate transmission. Refer also to the trigger condition for cyclic Timesync message transmissions (refer to [SWS_EthTSyn_00211]).

Two main scenarios are relevant for configuration of EthTSynCyclicMsgResumeTime:

- With EthTSynCyclicMsgResumeTime and EthTSynGlobalTimeTxPeriod both being configured as zero, a single shot mode is achieved that is solely triggered by the change of the timeBaseUpdateCounter.
- With EthTSynCyclicMsgResumeTime greater than EthTSynGlobalTimeTxPeriod a hold-over scenario in a Time Gateway can be configured:

- While Timesync messages are received from the Time Master side, the Timesync messages on the sub-busses are only triggered by immediate transmission (cyclic transmission is suspended while `cyclicMsgResumeCounter` is running)
- If no Timesync messages from the Time Master side are received anymore and a timeout is detected, cyclic transmission takes over (cyclic transmission no longer suspended because `cyclicMsgResumeCounter` has elapsed)
- reception of Timesync messages from the Time Master side resumes, the Timesync messages on the sub-busses are again triggered by immediate transmission (cyclic transmission is again suspended by running `cyclicMsgResumeCounter`)

[SWS_EthTSyn_00137]

Upstream requirements: RS_TS_20047

⌈If for a Time Domain:

- `EthTSynImmediateTimeSync` is set to `TRUE`,
- and `EthTSynCyclicMsgResumeTime` is greater than 0,
- and an immediate SYNC message is successfully sent

EthSyn shall set the counter `cyclicMsgResumeCounter` to `EthTSynCyclicMsgResumeTime` for the corresponding Time Domain.⌋

[SWS_EthTSyn_00400]

Status: DRAFT
Upstream requirements: RS_TS_20047

⌈While for a Time Domain:

- `cyclicMsgResumeCounter` is greater than 0

EthTSyn shall discard cyclic Timesync message transmission requests for that Time Domain.⌋

[SWS_EthTSyn_00401]

Upstream requirements: RS_TS_20047

⌈While for a Time Domain the `cyclicMsgResumeCounter` is greater than 0, EthTSyn shall decrement the `cyclicMsgResumeCounter` of the corresponding Time Domain by EthTSynMainFunctionPeriod on each invocation of EthTSyn_MainFunction.⌋

[SWS_EthTSyn_00139]

Upstream requirements: RS_TS_20047

⌈If the `cyclicMsgResumeCounter` is decremented to 0 or below, `EthTSyn` shall resume within the same `EthTSyn_MainFunction` call cyclic Timesync message transmission by requesting either a SYNC message transmission.⌋

Note: [SWS_EthTSyn_00139] is to ensure, that the first cyclic transmission is requested in the same main function call in which also `cyclicMsgResumeCounter` reaches 0 (refer to term "earliest possible" main function call in [SWS_EthTSyn_00211]. Whether the message is actually transmitted depends also on the `debounceCounter`.

7.8.1.4 Secure Time Synchronization

Refer to the chapter in StbM [2] for the configuration details of FV referenced in each Time Domain.

[SWS_EthTSyn_00246]

Upstream requirements: RS_TS_20072

⌈When the FV is referenced (refer `EthTSynIcvGenerationFvIdRef`, see link in note below) and the configured truncated FV length (`StbMFreshnessValueTruncLength`) is equal to FV length (`StbMFreshnessValueLength`) in StbM, the Time Master shall call the `StbM_GetTxFreshness` Api in order to obtain the full FV by using the `StbMFreshnessValueId`.⌋

[SWS_EthTSyn_00247]

Upstream requirements: RS_TS_20072

⌈When the FV is referenced (refer `EthTSynIcvGenerationFvIdRef`) and the configured truncated FV length (`StbMFreshnessValueTruncLength`) is less than FV length (`StbMFreshnessValueLength`) in StbM, the Time Master shall call the `StbM_GetTxFreshnessTruncData` Api in order to obtain the full FV and the truncated FV by using the `StbMFreshnessValueId`.⌋

[SWS_EthTSyn_00248]

Upstream requirements: RS_TS_20072

⌈If `StbM_GetTxFreshness` returns `E_OK`, the Time Master shall construct of the `AUTOSAR Sub-TLV: Time Authenticated` with FV and use the full FV in ICV generation.⌋

[SWS_EthTSyn_00249]

Upstream requirements: RS_TS_20072

⌈If `StbM_GetTxFreshnessTruncData` returns `E_OK`, the Time Master shall construct of the `AUTOSAR Sub-TLV: Time Authenticated` with truncated FV and use the full FV in ICV generation.⌋

[SWS_EthTSyn_00250]

Upstream requirements: RS_TS_20072

⌈If `StbM_GetTxFreshness` or `StbM_GetTxFreshnessTruncData` returns non-recoverable error code i.e, `E_NOT_OK`, the Time Master shall:

- stop the ICV generation (refer to chapter "ICV Generation", see link in note below) and accordingly set the `ICV_Flags` in `AUTOSAR Sub-TLV: Time Authenticated` of `Follow_Up` message,
- call `Det_ReportRuntimeError` with the parameter ErrorId := `ETHTSYN_E_FRESHNESSFAILURE` (refer [SWS_EthTSyn_00144]),
- report the security event `SEV_TSYN_ETH_FRESHNESS_NOT_AVAILABLE` to the IdsM (refer [SWS_EthTSyn_00261]).

⌋

Note: Refer to chapter ICV Generation 7.8.3.4

Note: Refer to the chapter in [11] for the configuration details of CSM job used for ICV generation.

[SWS_EthTSyn_00251]

Upstream requirements: RS_TS_20072

⌈If `EthTSynIcvGenerationBase` for the Time Domain is configured to `ICV_MAC`, the Time Master shall call `Csm_MacGenerate` to generate the `ICV` value.⌋

[SWS_EthTSyn_00252]

Upstream requirements: RS_TS_20072

⌈If `EthTSynIcvGenerationBase` for the Time Domain is configured to `ICV_SIGNATURE`, the Time Master shall call `Csm_SignatureGenerate` to generate the `ICV` value.⌋

Note: The `mode` parameter is intentionally left open for the implementer to choose (i.e. `CRYPTO_OPERATIONMODE_SINGLECALL` would possibly be the best option since it does not require further calls to `Csm`).

The CSM job used to generate the ICV can be configured to synchronous or asynchronous behaviour.

[SWS_EthTSyn_00253]

Upstream requirements: RS_TS_20072

⌈If the CSM job used to generate ICV is configured in synchronous behaviour, the Time Master shall disable ICV generation timeout monitoring.⌋

[SWS_EthTSyn_00254]

Upstream requirements: RS_TS_20072

⌈If `Csm_MacGenerate` or `Csm_SignatureGenerate` returns `E_OK`, the Time Master shall start the `EthTSynIcvGenerationTimeout`.⌋

[SWS_EthTSyn_00255]

Upstream requirements: RS_TS_20072

⌈When the `EthTSyn_IcvGenerationIndication` callback is called, the Time Master shall stop the running ICV generation timeout timer (`EthTSynIcvGenerationTimeout`).⌋

[SWS_EthTSyn_00256]

Upstream requirements: RS_TS_20072

⌈If one of the following conditions is true:

- authentication build counter has reached the configuration value `EthTSynTxAuthenticationBuildAttempts`,
- the verification of the ICV has returned a non-recoverable error such as returning E_NOT_OK or KEY_FAILURE,
- `EthTSynIcvGenerationTimeout` expires before the notification of the `EthTSyn_IcvGenerationIndication` callback,

the time master shall:

- stop the ICV generation and accordingly set the `ICV_Flags` in `AUTOSAR Sub-TLV: Time Authenticated` of `Follow_Up` message,
- report the security event `SEV_TSYN_ETH_ICV_GENERATION_FAILED` to the IdsM (refer to [SWS_EthTSyn_00261]).

⌋

Note: If ICV generation failed, there is no need to include the FV in the AUTOSAR Sub-TLV: Time Authenticated.

[SWS_EthTSyn_00257]

Upstream requirements: RS_TS_20072

⌈With the notification of the `EthTSyn_IcvGenerationIndication` callback, the Time Master shall add the generated ICV to `AUTOSAR Sub-TLV: Time Authenticated` and transmit the `Follow_Up` message.⌋

[SWS_EthTSyn_00258]

Upstream requirements: RS_TS_20072

⌈When a FV is referenced (refer `EthTSynIcvGenerationFvIdRef`), the Time Master shall notify the successful transmission of the Follow_Up message to FVM by calling StbM_SPduTxConfirmation.⌋

[SWS_EthTSyn_00402]
Status: DRAFT
Upstream requirements: RS_TS_20072

⌈For every transmission of messages that contain the AUTOSAR Sub-TLV: Time Authenticated, EthTSyn shall maintain an authentication build counter (refer `EthTSyn-TxAuthenticationBuildAttempts`).⌋

[SWS_EthTSyn_00403]
Status: DRAFT
Upstream requirements: RS_TS_20072

⌈Upon the initial processing of messages that contain the AUTOSAR Sub-TLV: Time Authenticated, the authentication build counter shall be set to 0.⌋

[SWS_EthTSyn_00404]
Status: DRAFT
Upstream requirements: RS_TS_20072

⌈If `StbM_GetTxFreshness` or `StbM_GetTxFreshnessTruncData` return recoverable error code (e.g., STBM_E_BUSY), the authentication build counter shall be incremented.⌋

[SWS_EthTSyn_00405]
Status: DRAFT
Upstream requirements: RS_TS_20072

⌈If `Csm_MacGenerate` or `Csm_SignatureGenerate` return recoverable error code (e.g., E_BUSY, QUEUE_FULL), the authentication build counter shall be incremented.⌋

[SWS_EthTSyn_00406]
Status: DRAFT
Upstream requirements: RS_TS_20072

⌈If building the authenticated message generation has failed and the authentication build counter has not yet reached the configuration value `EthTSynTx-AuthenticationBuildAttempts`, the freshness attempt and ICV calculation shall be retried in the next call of the `EthTSyn_MainFunction`.⌋

7.8.2 Link State and Transmission Mode

[SWS_EthTSyn_00019]
Upstream requirements: RS_TS_20048, RS_TS_20051

⌈A transceiver link state change (notification call of `EthTSyn_TrcvLinkStateChg`) from ETHTRCV_LINK_STATE_ACTIVE to ETHTRCV_LINK_STATE_DOWN resets the state machines for transmission and reception of Time Synchronization messages.⌋

[SWS_EthTSyn_00020]

Upstream requirements: RS_TS_20048, RS_TS_20051

⌈A transceiver link state change (notification call of `EthTSyn_TrcvLinkStateChg`) from ETHTRCV_LINK_STATE_DOWN to ETHTRCV_LINK_STATE_ACTIVE (re-)starts the transmission and reception of Time Synchronization messages.⌋

[SWS_EthTSyn_00021]

Upstream requirements: RS_TS_20048, RS_TS_20051

⌈If `EthTSyn_SetTransmissionMode` is called and the parameter Mode equals ETHTSYN_TX_OFF, all transmit request from `EthTSyn` shall be omitted on this Ethernet controller.⌋

[SWS_EthTSyn_00022]

Upstream requirements: RS_TS_20048, RS_TS_20051

⌈If `EthTSyn_SetTransmissionMode` is called and the parameter Mode equals ETHTSYN_TX_ON, all transmit request from `EthTSyn` on this Ethernet controller shall be able to be transmitted.⌋

7.8.3 Message Field Calculation and Assembling

Refer to chapter Message Field Calculation and Assembling in [1, PRS Time Synchronization Over Ethernet Protocol] for additional requirements.

7.8.3.1 SGW Calculation

Refer to chapter SGW Calculation in [1, PRS Time Synchronization Over Ethernet Protocol].

7.8.3.2 CRC Calculation

Refer to chapter CRC Calculation in [1, PRS Time Synchronization Over Ethernet Protocol] for additional requirements.

[SWS_EthTSyn_00096]

Upstream requirements: RS_TS_20061

⌈The function `Crc_CalculateCRC8H2F` as defined in [12] shall be used to calculate the CRC if configured.⌋

7.8.3.2.1 AUTOSAR Sub-TLV: Time Secured

Refer to chapter AUTOSAR Sub-TLV: Time Secured in [1, PRS Time Synchronization Over Ethernet Protocol].

7.8.3.2.2 AUTOSAR Sub-TLV: Status secured

Refer to chapter AUTOSAR Sub-TLV: Status secured in [1, PRS Time Synchronization Over Ethernet Protocol].

7.8.3.2.3 AUTOSAR Sub-TLV: UserData secured

Refer to chapter AUTOSAR Sub-TLV: UserData secured in [1, PRS Time Synchronization Over Ethernet Protocol].

7.8.3.3 Sequence Counter (sequenceId) Calculation

Refer to chapter Sequence Counter (sequenceId) Calculation in [1, PRS Time Synchronization Over Ethernet Protocol] for additional requirements.

7.8.3.4 ICV Generation

Refer to chapter ICV Generation in [1, PRS Time Synchronization Over Ethernet Protocol].

7.8.3.5 Message Assembling

[SWS_EthTSyn_00104]

Upstream requirements: RS_TS_20048, RS_TS_20072

⌈Refer to chapter Message Assembling in [1, PRS Time Synchronization Over Ethernet Protocol].⌋

7.8.3.6 Dynamic port configuration for Time Master or Time Slave

[SWS_EthTSyn_00414]
Status: DRAFT
Upstream requirements: RS_TS_20052, RS_TS_20053

⌈If the parameter `EthTSynPortRole` is set to `DYNAMIC` on any port, receiving a Sync message shall turn the respective reception port automatically into a Slave port. All remaining ports which are set as `DYNAMIC` shall turn as Master port.⌋

Note: Receiving Sync messages on different ports can cause inconsistencies.

[SWS_EthTSyn_00415]
Status: DRAFT
Upstream requirements: RS_TS_20052, RS_TS_20053

⌈If Sync and `Follow_Up` messages are not received on dynamically set to Slave port for `HoldOverTime`, then all the ports which are dynamically set to either Master or Slave shall turn back to `DYNAMIC`.⌋

Note: This will stop transmission of Sync and `Follow_Up` on dynamically set to Master port and allow for a seamless change-over of the Slave port.

[SWS_EthTSyn_00416]
Status: DRAFT
Upstream requirements: RS_TS_20052, RS_TS_20053

⌈If Sync and `Follow_Up` are not received on a Slave port for `HoldOverTime`, then all the ports which are set to `TIME_MASTER` (dynamically or statically) shall stop transmission of Sync and `Follow_Up` messages.⌋

Note: It does not matter whether the Slave port was dynamically or statically configured as `TIME_SLAVE`; in any case transmission of Sync and `Follow_Up` messages shall be stopped after `HoldOverTime` to prevent discontinuities.

7.8.4 System Wide Global Time Master

If the `Time Base` referred to by `EthTSynSynchronizedTimeBaseRef` of the ports in `EthTSynPortRole` `TIME_MASTER` is not referenced by any other `TIME_SLAVE` port of any other `<Bus>TSyn` module, the master is considered a `System Wide Global Time Master` for the top level `Time Domain` of the [13, System Template].

Refer also to section 7.10 on the difference between an AR `Time Gateway` and an IEEE `Time-aware Bridge`.

7.9 Acting as Time Slave

Refer to chapter Acting as Time Slave in [1, PRS Time Synchronization Over Ethernet Protocol] for additional requirements.

7.9.1 Message processing

In addition to the Follow_Up message fields:

- `preciseOriginTimeStamp`
- `correctionField`

(refer to [1, PRS Time Synchronization Over Ethernet Protocol] in chapter `Message Processing` which are received by the Time Slave on the bus from the Time Master, this chapter defines and uses the following internal variables for calculation of the Rx Time Tuple of a Synchronized Time Base:

- $T1_{VLT}$: Ingress timestamp of SYNC message as captured by HW in the Ethernet controller or by SW in `EthTSyn_RxIndication`.
- T2: Global Time component of the Rx Time Tuple (equivalent to TG_Rx in the StbM).
- $T2_{VLT}$: Virtual Local Time component of the Rx Time Tuple (equivalent to TV_Rx in the StbM).
- $T3_{VLT}$: Current time read out from Ethernet controller hardware - used for correlation of StbM time and Ethernet HW clock.
- $T4_{VLT}$: Current virtual local time in StbM - used for correlation of StbM local time and Ethernet HW clock.
- T_{SRD}: SYNC reception delay as difference between $T3_{VLT}$ and $T1_{VLT}$.

Figure 7.2 illustrates the flow of actions to calculate the Rx Time Tuple from the data that is received in the Sync and in the Follow_Up messages. The diagram helps to understand the requirements in this chapter. Further details are given in sequence diagram Figure 9.5.

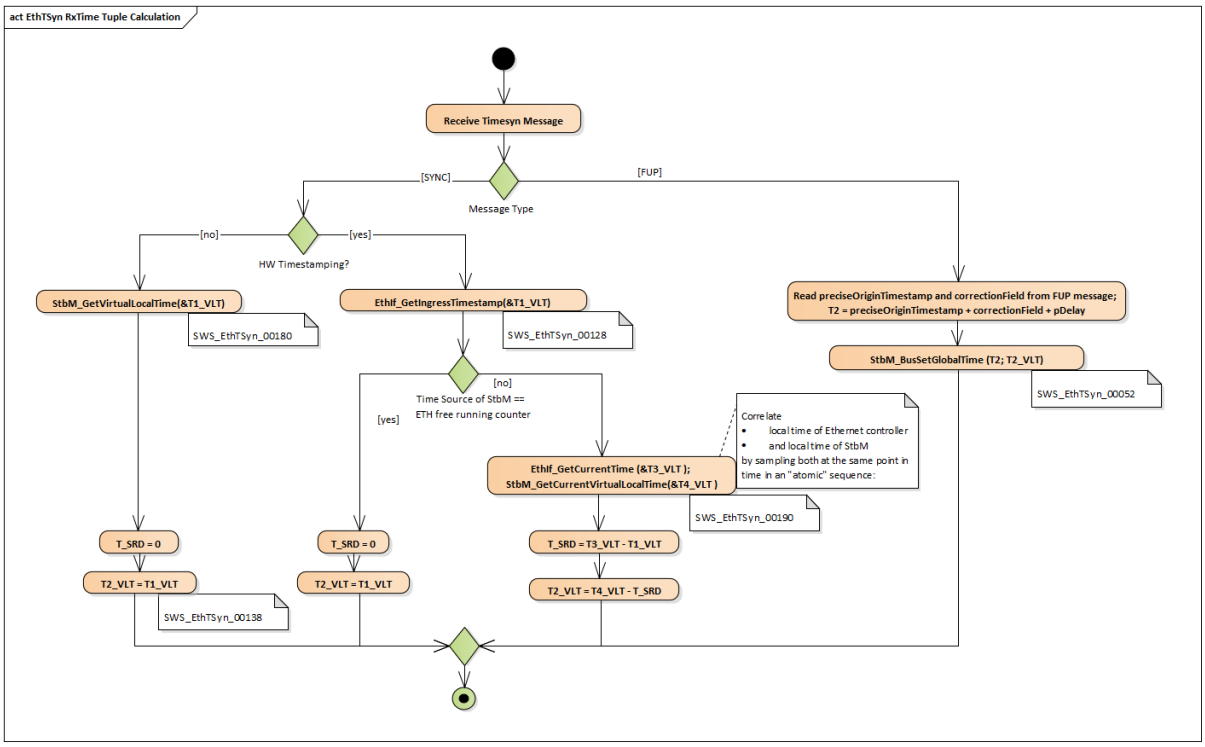


Figure 7.2: Evaluate Timesync message

[SWS_EthTSyn_00412]

Upstream requirements: RS_TS_20048

⌈If `EthTSynPortRole` is set to `TIME_SLAVE` or `DYNAMIC`, the Sync and `Follow_Up` message shall be processed in `EthTSyn_RxIndication` according to requirements [SWS_EthTSyn_00128], [SWS_EthTSyn_00138], [SWS_EthTSyn_00180], [SWS_EthTSyn_00190].⌋

[SWS_EthTSyn_00413]

Upstream requirements: RS_TS_20048

⌈If `EthTSynPortRole` is set to `TIME_SLAVE`, no other port shall accept the Sync and `Follow_Up` messages, i.e., time synchronization messages are only processed by the configured Slave port.⌋

Note: For [SWS_EthTSyn_00412] and [SWS_EthTSyn_00413]: When configuring more than one Slave port for the same Time Domain, inconsistencies may arise.

[SWS_EthTSyn_00128]

Upstream requirements: RS_TS_20048

⌈If `EthTSynHardwareTimestampSupport` is set to TRUE, when `EthTSyn_RxIndication` is called on ingress of a Sync message, then the EthTSyn shall retrieve the ingress time stamp $T1_{VLT}$ from the meta data item of type `TIMETUPLE_TYPE_PTR` of the PDU identified by parameter `RxPduId`.⌋

[SWS_EthTSyn_00138]

Upstream requirements: RS_TS_20048

⌈On invocation of EthTSyn_RxIndication for the Sync message and if EthTSyn-HardwareTimestampSupport is set to TRUE and if the StbM does use the Ethernet hardware counter as Virtual Local Time Source for the Time Base:

- The $T2_{VLT}$ part of the Rx Time Tuple shall be set to the value of $T1_{VLT}$ (i.e., $T2_{VLT} = T1_{VLT}$)
- The Sync reception delay T_{SRD} shall be set to 0

⌋

[SWS_EthTSyn_00180]

Upstream requirements: RS_TS_20048

⌈On invocation of EthTSyn_RxIndication and if EthTSynHardwareTimestamp-Support is set to FALSE the following sequence shall be applied:

- Immediately establish a protection against interruptions and run the next step directly afterwards:
- Retrieve the reference time $T1_{VLT}$ for the Sync message via StbM_Get-CurrentVirtualLocalTime from the StbM
- The protection against interruptions may be removed now.

The $T2_{VLT}$ part of the Rx Time Tuple shall be set to the value of $T1_{VLT}$ (i.e., $T2_{VLT} = T1_{VLT}$). The Sync reception delay T_{SRD} shall be set to 0.

⌋

Note: Immediately protecting against interruptions means that there shall be no frame checks before. If called in context of the Rx interrupt with interrupt nesting disabled, protection against interruptions is implicitly done by the controller. Once the interrupts are locked, it is ok to check whether the received message is a Sync message for which a snapshot of the Virtual Local Time shall be taken, but no other frame checks (e.g., SC validation) shall be done before taking the snapshot. Once the snapshot has been taken it is ok to remove the protection against interruptions and to make the necessary validations. This means that a snapshot of the Virtual Local Time shall be taken even if the succeeding validations fail and thus making the snapshot superfluous.

[SWS_EthTSyn_00190]

Upstream requirements: RS_TS_20048

⌈On invocation of EthTSyn_RxIndication, a reference time shall be retrieved on reception of the Sync message if EthTSynHardwareTimestampSupport is set to TRUE and if the StbM does not use the Ethernet hardware counter as Virtual Local Time Source for the Time Base by applying the following sequence:

- Protect the following two steps against interuptions:

- the current time of the Ethernet hardware counter shall be retrieved via via parameter `currentTimeTuplePtr ->timestampClockValue` of `EthIf_GetCurrentTimeTuple` from the `EthIf` and converted to the Virtual Local Time $T3_{VLT}$
- the current value of the Virtual Local Time of the Time Base shall be retrieved as $T4_{VLT}$ via `StbM_GetCurrentVirtualLocalTime`.
- the `Sync` reception delay T_{SRD} shall be calculated as $T3_{VLT} - T1_{VLT}$.
- $T2_{VLT}$ shall be calculated as $T4_{VLT}$ - T_{SRD}.

⌋

[SWS_EthTSyn_00052]

Upstream requirements: RS_TS_20048

⌈When a valid Follow-Up message is received,

the EthTSyn shall

- calculate T2 by adding the values
  - preciseOriginTimestamp (from the Follow-Up message),
  - correctionField (from the Follow-Up message),
  - Pdelay (calculated according to [SWS_EthTSyn_00264])
- and forward the resulting Rx Time Tuple $[T2; T2_{VLT}]$ to the StbM module via `StbM_BusSetGlobalTime`

⌋

Note: The Pdelay value is not influenced significantly by a RateRatio according to [4, IEEE Std 802.1AS-2011] Note-2 of chapter "11.2.15.2.4 computePropTime()".

[SWS_EthTSyn_00266]

Status: DRAFT
Upstream requirements: RS_TS_20048

⌈If

- configuration parameter `EthTSynNeighbourRateRatioConfig` is set to `CMLDS` or `TransportSpecific`
- and a valid `neighborRateRatio` has been calculated (refer to [SWS_EthTSyn_00263])
- and a new valid Follow-Up message has been received,

then EthTSyn shall calculate in the next main function the `rateRatio` as given in PRS_TS_00261 in [1, PRS Time Synchronization Over Ethernet Protocol].⌋

[SWS_EthTSyn_00267]
Status: DRAFT
Upstream requirements: RS_TS_20048

⌈If a `rateRatio` has been successfully calculated (refer to [SWS_EthTSyn_00266]), then when calling `StbM_BusSetGlobalTime`, EthTSyn shall set parameter `measureDataPtr->rateDeviation` as follows:

- calculate `rateDeviationValue` as
  - `rateDeviationValue` = (`rateRatio` - 1) * 2^{41}
  - and then truncate `rateDeviationValue` to the next smaller signed integer
- and set `rateDeviationStatus` to `ETH_RATE_OK`.

If the calculated rate deviation value exceeds the value range of `rateDeviationValue`, then EthTSyn shall

- set `rateDeviationValue` to `SINT32_MIN` or `SINT32_MAX`, respectively.
- set `rateDeviationStatus` to `ETH_RATE_EXCEEDED`.

⌋

Note: According to [4, IEEE Std 802.1AS-2011] assumption is that the fractional value of the rate deviation is within the range [-(2^{-10} - 2^{-41}), 2^{-10} - 2^{-41}], i.e., approximately [-9.766 * 10^{-4}, 9.766 * 10^{-4}].

[SWS_EthTSyn_00268]
Status: DRAFT
Upstream requirements: RS_TS_20048

⌈If a `rateRatio` has not yet been successfully calculated (refer to [SWS_EthTSyn_00266]), then when calling `StbM_BusSetGlobalTime`, EthTSyn shall set parameter `measureDataPtr->rateDeviation` as follows:

- `rateDeviationValue` to 0
- and `rateDeviationStatus` to `ETH_RATE_NOT_AVAILABLE`.

⌋

[SWS_EthTSyn_00150]
Upstream requirements: RS_TS_20058

⌈When calling `StbM_BusSetGlobalTime`, EthTSyn shall pass the current `linkDelay` value (refer [SWS_EthTSyn_00264]) by the parameter `measureDataPtr->pathDelay` to the StbM.⌋

[SWS_EthTSyn_00129]

Upstream requirements: RS_TS_20051

⌈When providing a new Global Time tuple to the StbM via `StbM_BusSetGlobalTime`, `EthTSyn` shall set the SYNC_TO_GATEWAY bit in timeBaseStatus (structure member, which is referenced by the parameter timeTuplePtr), according to the SGW value (refer to [PRS_TS_00156]). The remaining status bits shall be set to 0.⌋

[SWS_EthTSyn_00230]

Upstream requirements: RS_TS_20062

⌈If `EthTSynMessageCompliance` is either set to `TRUE` or if `EthTSynRxSubTLVUserData` is set to `FALSE`, `EthTSyn` shall pass a NULL pointer as parameter `UserData` of `StbM_BusSetGlobalTime`.⌋

[SWS_EthTSyn_00227]

Upstream requirements: RS_TS_20069

⌈On invocation of `EthTSyn_GetProtocolParam` `EthTSyn` shall return the following values received in the latest Follow_Up information TLV via argument protocolParam:

- cumulativeScaledRateOffset
- gmTimeBaseIndicator
- lastGmPhaseChange
- scaledLastGmFreqChange

Member protocolType of argument protocolParam shall be set to STBM_TIMESYNC_ETHERNET.⌋

7.9.1.1 Runtime Error detection

[SWS_EthTSyn_00146]

Upstream requirements: RS_TS_20051

⌈If `EthTSynMasterSlaveConflictDetection` is set to `TRUE` and if the Time Slave receives a Sync frame with different sourcePortIdentity (i.e., different MAC addresses), it shall report a runtime error by calling `Det_ReportRuntimeError` with error code ETHTSYN_E_TSCONFLICT and discard the received Sync frame.⌋

7.9.1.2 Frame Debouncing

Refer to chapter Frame Debouncing in [1, PRS Time Synchronization Over Ethernet Protocol] for additional requirements.

[SWS_EthTSyn_00232]

Upstream requirements: RS_TS_20072

⌈During the EthTSynGlobalTimeRxDebounceTime, if the sequence is reset, then the Time Slave shall report the security event SEV_TSYN_ETH_MSG_SEQUENCE_ERROR to the IdsM (refer to [SWS_EthTSyn_00261]).⌋

7.9.1.3 Secure Time Synchronization

Refer to the chapter in StbM [2] for the configuration details of FV referenced in each Time Domain.

[SWS_EthTSyn_00233]

Upstream requirements: RS_TS_20072

⌈When the FV is referenced (refer EthTSynIcvVerificationFvIdRef), FVL is valid and 'ICV with FV' bit is set in ICV_Flags of received Follow_Up message (refer [PRS_TS_00248]), the Time Slave shall call the StbM_GetRxFreshness Api in order to obtain the Freshness Value by using

- the StbMFreshnessValueId from the reference EthTSynIcvVerificationFvIdRef
- the StbMTruncatedFreshnessValue as received in the FV field of the Follow_Up message
- the StbMTruncatedFreshnessValueLength as received in the FVL field of the Follow_Up message
- the StbMAuthVerifyAttempts as the number of failed verification attempt counts for the current message (ICV verification attempt counter)
- the StbMFreshnessValueLength from the reference EthTSynIcvVerificationFvIdRef

⌋

[SWS_EthTSyn_00234]

Upstream requirements: RS_TS_20072

⌈If StbM_GetRxFreshness returns E_OK, the Time Slave shall use the FV in ICV verification.⌋

[SWS_EthTSyn_00236]

Upstream requirements: RS_TS_20072

⌈If

- StbM_GetRxFreshness returns non-recoverable error code (e.g., E_NOT_OK)

- or FVL is invalid and ICV with FV bit is set in ICV_Flags of received Follow_Up message (refer [PRS_TS_00250]),

the ICV verification of received Follow_Up message is considered to be failed, and the Time Slave shall :

- stop the ICV verification (refer to chapter "ICV Verification". see link in note below) and discard the received `Follow_Up` message,
- call `Det_ReportRuntimeError` with the parameter ErrorId := `ETHTSYN_E_FRESHNESSFAILURE` (refer [SWS_EthTSyn_00144]),
- report the security event `SEV_TSYN_ETH_FRESHNESS_NOT_AVAILABLE` to the IdsM (refer to [SWS_EthTSyn_00261]).

⌋

Refer to the chapter in [11] for the configuration details of CSM job used for ICV verification.

[SWS_EthTSyn_00237]

Upstream requirements: RS_TS_20072

⌈If `EthTSynIcvVerificationBase` for the Time Domain is configured to `ICV_MAC`, the Time Slave shall call `Csm_MacVerify` to verify the `ICV` value.⌋

[SWS_EthTSyn_00238]

Upstream requirements: RS_TS_20072

⌈If `EthTSynIcvVerificationBase` for the Time Domain is configured to `ICV_SIGNATURE`, the Time Slave shall call `Csm_SignatureVerify` to verify the `ICV` value.⌋

Note: 7.9.2.4

Note: The `mode` parameter is intentionally left open for the implementer to choose (i.e. `CRYPTO_OPERATIONMODE_SINGLECALL` would possibly be the best option since it does not require further calls to `Csm`).

The CSM job used to generate the ICV can be configured to synchronous or asynchronous behaviour.

[SWS_EthTSyn_00239]

Upstream requirements: RS_TS_20072

⌈The ICV verification timeout observation is disabled, when the CSM job to verify ICV is configured in synchronous behaviour. In this case, the `EthTSynIcvVerificationTimeout` shall be set to 0.⌋

[SWS_EthTSyn_00240]

Upstream requirements: RS_TS_20072

⌈If `Csm_MacVerify` or `Csm_SignatureVerify` returns `E_OK`, the Time Slave shall start the `EthTSynIcvVerificationTimeout`.⌋

[SWS_EthTSyn_00241]

Upstream requirements: RS_TS_20072

⌈If Csm_MacVerify or Csm_SignatureVerify returns recoverable error code (e.g., CRYPTO_E_BUSY, CRYPTO_QUEUE_FULL), the current verification of received Follow_Up message is considered to be failed , and the Time Slave shall increment the authentication build counter for this Follow_Up message.⌋

[SWS_EthTSyn_00242]

Upstream requirements: RS_TS_20072

⌈The `EthTSynIcvVerificationTimeout` shall be stopped with the notification of the `EthTSyn_IcvVerificationIndication` callback.⌋

[SWS_EthTSyn_00243]

Upstream requirements: RS_TS_20072

⌈If one of the following conditions is true:

- the authentication build counter has reached the configuration value `EthTSyn-RxAuthenticationBuildAttempts`,
- the ICV verification attempt counter has reached the configuration value EthTSynIcvVerificationAttempts,
- the verification of the ICV has returned a non-recoverable error such as returning E_NOT_OK or KEY_FAILURE,
- EthTSynIcvVerificationTimeout expires before the notification of the EthTSyn_IcvVerificationIndication callback,

the time slave shall:

- stop the ICV verification (refer to chapter ICV Verification, see link in note below) and discard the received `Follow_Up` message,
- report the security event `SEV_TSYN_ETH_ICV_VERIFICATION_FAILED` to the IdsM (refer to [SWS_EthTSyn_00261]).

⌋

Note: 7.9.2.4

[SWS_EthTSyn_00407]
Status: DRAFT
Upstream requirements: RS_TS_20072

⌈For every reception of messages that require ICV verification, EthTSyn shall maintain an authentication build counter (refer `EthTSynRxAuthenticationBuildAttempts`).⌋

[SWS_EthTSyn_00408]
Status: DRAFT
Upstream requirements: RS_TS_20072

⌈Upon the initial processing of messages that require ICV verification, the authentication build counter shall be set to 0.⌋

[SWS_EthTSyn_00409]
Status: DRAFT
Upstream requirements: RS_TS_20072

⌈If `StbM_GetRxFreshness` returns recoverable error code (e.g., STBM_E_BUSY), the authentication build counter shall be incremented and no attempt for verification of the ICV shall be executed.⌋

[SWS_EthTSyn_00410]
Status: DRAFT
Upstream requirements: RS_TS_20072

⌈If building the authenticated message verification has failed and the authentication build counter has not yet reached the configuration value `EthTSynRxAuthenticationBuildAttempts`, the freshness attempt and ICV verification shall be retried in the next call of the `EthTSyn_MainFunction`.⌋

[SWS_EthTSyn_00411]
Status: DRAFT
Upstream requirements: RS_TS_20072

⌈If the verification of the ICV could be successfully executed but the verification failed (e.g. the MAC verification has failed or the key was invalid), the ICV verification attempt counter shall be incremented and the authentication build counter shall be set to 0.⌋

Note: Resetting the authentication build counter shall prevent to drop the authentication process too early even though ICV verification attempts are still possible.

[SWS_EthTSyn_00244]
Upstream requirements: RS_TS_20072

⌈When the `EthTSyn_IcvVerificationIndication` callback is called and ICV verification result is successful, the Time Slave shall accept the `Follow_Up` message and call `StbM_BusSetGlobalTime` to forward the global time to StbM.⌋

[SWS_EthTSyn_00245]

Upstream requirements: RS_TS_20072

⌈When the `EthTSyn_IcvVerificationIndication` callback is called and ICV verification result is unsuccessful, the Time Slave shall discard the `Follow_Up` message.⌋

7.9.2 Message Field Validation and Disassembling

Additional content to this chapter can be found in [1, PRS Time Synchronization Over Ethernet Protocol] in chapter Message Field Validation and Disassembling.

7.9.2.1 SGW Calculation

Refer to chapter SGW Calculation in [1, PRS Time Synchronization Over Ethernet Protocol].

7.9.2.2 CRC Validation

[SWS_EthTSyn_00111]

Upstream requirements: RS_TS_20061

⌈The function `Crc_CalculateCRC8H2F` as defined in [12] shall be used to calculate the CRC if configured.⌋

Refer to chapter CRC Calculation in [1, PRS Time Synchronization Over Ethernet Protocol] for additional requirements.

7.9.2.2.1 AUTOSAR Sub-TLV: Time Secured

Refer to chapter AUTOSAR Sub-TLV: Time Secured in [1, PRS Time Synchronization Over Ethernet Protocol].

7.9.2.2.2 AUTOSAR Sub-TLV: Status secured

Refer to chapter AUTOSAR Sub-TLV: Status secured in [1, PRS Time Synchronization Over Ethernet Protocol].

7.9.2.2.3 AUTOSAR Sub-TLV: UserData secured

Refer to chapter AUTOSAR Sub-TLV: UserData secured in [1, PRS Time Synchronization Over Ethernet Protocol].

7.9.2.3 Sequence Counter (sequenceId) Validation

Refer to chapter Sequence Counter (sequenceId) Validation in AUTOSAR Time Synchronization Protocol Specification[1, PRS Time Synchronization Over Ethernet Protocol] for additional requirements.

7.9.2.4 ICV Verification

Refer to chapter ICV Verification in PRS-TimeSyncProtocol [1, PRS Time Synchronization Over Ethernet Protocol].

7.9.2.5 Message Disassembling

Refer to chapter Message Disassembling in [1, PRS Time Synchronization Over Ethernet Protocol].

7.10 Time Gateway

A `Time Gateway` synchronizes a `Time Base` of the [2, StbM] to a `TIME_SLAVE` port of a `<Bus>TSyn` module and uses the `TIME_MASTER` port(s) of potentially one ore more (different or same) `<Bus>TSyn` modules to distribute time information. The role of `Time Gateway` is an implicit configuration of the `EthTSyn` module only. It means there is potentially one port in `EthTSynPortRole TIME_SLAVE` (see 7.9) and potentially more than one port in `EthTSynPortRole TIME_MASTER` (see 7.8) with the same `EthTSynSynchronizedTimeBaseRef`.

In the context of the [13, System Template] a `EthTSynGlobalTimeDomain`, identified by its `EthTSynGlobalTimeDomainId`, represents a `sub domain`. The root of the top level `Time Domain` is defined by the `System Wide Global Time Master` (see 7.8.4).

7.10.1 Differences to a Time Aware Bridge

In a `Time-aware Bridge` (according to [4, IEEE Std 802.1AS-2011]) the Ethernet ports participating in the distribution of time are connected through a `PTP Relay Instance`. There is one relay instance per gPTP Time Domain as identified by its Time Domain ID in every Bridge. In contrast to an Autosar Time Gateway, the IEEE Time

Aware Bridge does not maintain a time base for the synchronized time. Instead it only uses a local time base for time stamping and to measure short time intervals between reception and transmission of gPTP messages.

The Grandmaster of [4, IEEE Std 802.1AS-2011] is not to be confused with the System Wide Global Time Master (see 7.8.4). As described in 7.10 above, Autosar defines a more abstract top level TimeDomain based on a hierarchy of TimeBases in [13, AUTOSAR System Template] Chapter 9.2 *"The big Picture"*.

7.11 Time Recording

7.11.1 Time Validation

[SWS_EthTSyn_00212]

Upstream requirements: RS_TS_00034

⌈The EthTSyn shall support Time Validation, if EthTSynTimeValidationSupport set to TRUE.⌋

[SWS_EthTSyn_00213]

Upstream requirements: RS_TS_00034

⌈If

- EthTSynTimeValidationSupport is enabled and
- EthTSynEnableTimeValidation for the Time Domain is enabled,

EthTSyn shall do time recording for Time Validation for that Time Domain.⌋

[SWS_EthTSyn_00214]

Upstream requirements: RS_TS_20048

⌈If time recording for Time Validation is enabled for a Master Port Domain of a Time Domain (refer to [SWS_EthTSyn_00212] and [SWS_EthTSyn_00213]). The EthTSyn shall call StbM_EthSetMasterTimingData upon successful transmission of a Sync message (refer to EthTSyn TimesyncSequence).⌋

Note: EthTSyn TimesyncSequence is shown in Figure 9.2.

[SWS_EthTSyn_00215]

Upstream requirements: RS_TS_20048

⌈Upon invocation of StbM_EthSetMasterTimingData (refer to [SWS_EthTSyn_00214]) the EthTSyn shall pass the following parameters

- The sequenceId of the sent Sync message,
- The sourcePortIdentity as sent in the Sync message and

- The Virtual Local Time $T2_{VLT}$ sampled on egress of the Sync message (refer to [SWS_EthTSyn_00127]),
- The preciseOriginTimestamp as copied to the Follow_Up message and (refer to [SWS_EthTSyn_00188]).
- The correctionField as copied to the Follow_Up message.

by the parameter measureDataPtr.⌋

[SWS_EthTSyn_00216]

Upstream requirements: RS_TS_00034

⌈If

- Time recording for Time Validation is enabled for a Time Domain (refer to [SWS_EthTSyn_00212] and [SWS_EthTSyn_00213]) and
- EthTSyn is configured as Time Slave for that Time Domain.

EthTSyn shall call StbM_EthSetSlaveTimingData upon successful reception of a FollowUp message (refer to EthTSyn TimesyncSequence).

StbM_EthSetSlaveTimingData shall be called after StbM_BusSetGlobalTime.⌋

Note: EthTSyn TimesyncSequence is shown in Figure 9.2.

Note: StbM_BusSetGlobalTime shall be called first, because it updates the Synclocal Time Tuple (refer to [2]), which is required by StbM_EthSetSlaveTimingData).

[SWS_EthTSyn_00217]

Upstream requirements: RS_TS_00034

⌈Upon invocation of StbM_EthSetSlaveTimingData EthTSyn shall pass following values

- The sequenceId received in the Follow_Up message,
- The sourcePortIdentity received in the Follow_Up message and
- The Virtual Local Time $T1_{VLT}$ sampled on ingress of the Sync message (refer to [SWS_EthTSyn_00128]),
- The preciseOriginTimestamp received in the Follow_Up message
- The correctionField received in the Follow_Up message and
- The current value of the Pdelay

to the function by the parameter measureDataPtr.

The struct members

- measureDataPtr->referenceLocalTimestamp and
- measureDataPtr->referenceGlobalTimestamp

shall be passed as 0.⌋

Note: The `EthTSyn` passes 0 to avoid undefined values. The StbM will calculate the structure members referenceLocalTimestamp and referenceGlobalTimestamp based on the Synclocal Time Tuple (refer to SWS_StbM_00471 in [2]).

7.11.1.1 Recording of Pdelay Measurement

[SWS_EthTSyn_00218]

Upstream requirements: RS_TS_00034

⌈If

- time recording for Time Validation is enabled for a Time Domain (refer to [SWS_EthTSyn_00212] and [SWS_EthTSyn_00213])
- and `EthTSyn` is configured as Time Master for that Time Domain

`EthTSyn` shall call `StbM_GetCurrentTime` to retrieve a Time Tuple [$T_{refPDResponder}$; $T_{VLT_refPDResponder}$] before sending the `Pdelay_Resp` message (refer to `EthTSyn PdelaySequence`).⌋

Note: The Time Tuple [$T_{refPDResponder}$; $T_{VLT_refPDResponder}$] will be used for coherent conversion of t2 or `requestReceiptTimestamp` and t3 or `responseOriginTimestamp` into Global Time values, i.e., of instances in Virtual Local Time values into instances in Global Time.

Note: `EthTSyn PdelaySequence` is shown in Figure 9.3

[SWS_EthTSyn_00219]

Upstream requirements: RS_TS_00034

⌈If

- time recording for Time Validation is enabled for the Time Domain (refer to [SWS_EthTSyn_00212] and [SWS_EthTSyn_00213])
- and `EthTSyn` is configured as Time Master for that Time Domain

`EthTSyn` shall call `StbM_EthSetPdelayResponderData` after the current Pdelay measurement is finished, i.e., upon transmission of the `Pdelay_Resp_Follow_Up` message (refer to `EthTSyn PdelaySequence`).⌋

Note: `EthTSyn PdelaySequence` is shown in Figure 9.3

[SWS_EthTSyn_00220]

Upstream requirements: RS_TS_00034

⌈The Time Master shall pass the following parameters

- the `sequenceId` of the received `Pdelay_Req` message and

- the `sourcePortIdentity` of the received `Pdelay_Req` message,
- the `sourcePortIdentity` of the sent `Pdelay_Resp` message
- t2 (refer to [SWS_EthTSyn_00160], [SWS_EthTSyn_00124])
- t3 (refer to [SWS_EthTSyn_00159] , [SWS_EthTSyn_00122]) and
- the sampled reference Time Tuple [$\mathtt{T}_{refPDResponder}$; $\mathtt{T}_{VLT_refPDResponder}$] (refer to [SWS_EthTSyn_00218])

to `StbM_EthSetPdelayResponderData` upon invocation by the parameter measure DataPtr.⌋

[SWS_EthTSyn_00223]

Upstream requirements: RS_TS_00034

⌈If time recording for Time Validation is enabled for the Time Domain (refer to [SWS_EthTSyn_00212] and [SWS_EthTSyn_00213]), the Time Slave shall call `StbM_GetCurrentTime` to retrieve a Time Tuple [$\mathtt{T}_{refPDInitiator}$; $\mathtt{T}_{VLT_refPDInitiator}$] before sending the `pDelay_Req` message (refer to `EthTSyn PdelaySequence`).⌋

Note: The Time Tuple Tuple [$\mathtt{T}_{refPDInitiator}$; $\mathtt{T}_{VLT_refPDInitiator}$] will be used for coherent conversion of t1 and t4 from Virtual Local Time values into Global Time values.

Note: `EthTSyn PdelaySequence` is shown in Figure 9.3

[SWS_EthTSyn_00221]

Upstream requirements: RS_TS_00034

⌈If

- time recording for Time Validation is enabled for the Time Domain (refer to [SWS_EthTSyn_00212] and [SWS_EthTSyn_00213]) and
- `EthTSyn` is configured as Time Slave for that Time Domain

`EthTSyn` shall call `StbM_EthSetPdelayInitiatorData` after the current Pdelay measurement is finished, i.e., upon reception of the `Pdelay_Resp_Follow_Up` message (refer to `EthTSyn PdelaySequence`).⌋

Note: `EthTSyn PdelaySequence` is shown in Figure 9.3

[SWS_EthTSyn_00222]

Upstream requirements: RS_TS_00034

⌈The Time Slave shall pass the following parameters

- the `sequenceId` of the sent `Pdelay_Req` message,
- the `sourcePortIdentity` of the sent `Pdelay_Req` message,
- the `sourcePortIdentity` of the received `Pdelay_Resp` message

- t1 (refer to [SWS_EthTSyn_00013]),

t4 (refer to[SWS_EthTSyn_00049]),

- the `requestReceiptTimestamp` from the `Pdelay_Resp` message,
- the `responseOriginTimestamp` from the `Pdelay_Resp_Follow_Up` message,
- the sampled reference Time Tuple [$\mathtt{T}_{refPDInitiator}$; $\mathtt{T}_{VLT_refPDInitiator}$] (refer [SWS_EthTSyn_00223])

to `StbM_EthSetPdelayInitiatorData` upon invocation by the parameter `measureDataPtr`.⌋

7.12 Security Events

[SWS_EthTSyn_00231]

Upstream requirements: SRS_BSW_00489, SRS_BSW_00491

⌈If security event reporting has been enabled for the EthTSyn module (`EthTSynEnableSecurityEventReporting` is set to true) the respective security events shall be reported to the IdsM [14] via the interfaces defined in BSWGeneral [3].⌋

The following table lists the security events which are standardized for the EthTSyn together with their trigger conditions.

[SWS_EthTSyn_00261] Security events for EthTSyn

Upstream requirements: SRS_BSW_00488, SRS_BSW_00489, SRS_BSW_00490

⌈

Name	*Description*	*ID*
SEV_TSYN_ETH_ICV_GENERATION_FAILED	ICV generation for a Follow_Up message failed.	73
SEV_TSYN_ETH_ICV_VERIFICATION_FAILED	ICV verification of a received Follow_Up message failed.	74
SEV_TSYN_ETH_FRESHNESS_NOT_AVAILABLE	Failed to get freshness value from FvM.	75
SEV_TSYN_ETH_MSG_SEQUENCE_ERROR	Failed to receive correct sequence of SYNC and FUP from the TimeMaster within (EthTSynGlobalTimeFollowUp Timeout).	76

⌋

The following tables lists the security events together with their context data.

[SWS_EthTSyn_92000] Security event context data definition: SEV_TSYN_ETH_ICV_GENERATION_FAILED

Upstream requirements: SRS_BSW_00488, SRS_BSW_00489

⌈

SEV Name	*SEV_TSYN_ETH_ICV_GENERATION_FAILED*	
ID	73	
Description	ICV generation for a Follow_Up message failed.	
Context Data Version	1	
Context Data	*Data Type*	*Allowed Values*
GlobalTimeDomainId	uint8	

⌋

[SWS_EthTSyn_92001] Security event context data definition: SEV_TSYN_ETH_ICV_VERIFICATION_FAILED

Upstream requirements: SRS_BSW_00488, SRS_BSW_00489

⌈

SEV Name	*SEV_TSYN_ETH_ICV_VERIFICATION_FAILED*	
ID	74	
Description	ICV verification of a received Follow_Up message failed.	
Context Data Version	1	
Context Data	*Data Type*	*Allowed Values*
GlobalTimeDomainId	uint8	

⌋

[SWS_EthTSyn_92002] Security event context data definition: SEV_TSYN_ETH_FRESHNESS_NOT_AVAILABLE

Upstream requirements: SRS_BSW_00488, SRS_BSW_00489

⌈

SEV Name	*SEV_TSYN_ETH_FRESHNESS_NOT_AVAILABLE*	
ID	75	
Description	Failed to get freshness value from FvM.	
Context Data Version	1	
Context Data	*Data Type*	*Allowed Values*
GlobalTimeDomainId	uint8	

⌋

[SWS_EthTSyn_92003] Security event context data definition: SEV_TSYN_ETH_MSG_SEQUENCE_ERROR

Upstream requirements: SRS_BSW_00488, SRS_BSW_00489

⌈

SEV Name	*SEV_TSYN_ETH_MSG_SEQUENCE_ERROR*	
ID	76	
Description	Failed to receive correct sequence of SYNC and FUP from the TimeMaster within (EthTSyn GlobalTimeFollowUpTimeout).	
Context Data Version	1	
Context Data	*Data Type*	*Allowed Values*
GlobalTimeDomainId	uint8	

⌋

7.13 Error Classification

Chapter [3, General Specification of Basic Software Modules] 7.2 *"Error Handling"* describes the error handling of the Basic Software in detail. Above all, it constitutes a classification scheme for the error types that may occur in BSW modules.

Based on this foundation, the following section specifies particular errors arranged in the respective subsections below.

[SWS_EthTSyn_00029]

Upstream requirements: RS_TS_20051, SRS_BSW_00323, SRS_BSW_00417

⌈On errors and exceptions, the EthTSyn module shall not modify its current module state but shall simply report the error event.⌋

7.13.1 Development Errors

The detection of development errors is configurable (refer EthTSynDevErrorDetect).

[SWS_EthTSyn_00030] Definition of development errors in module EthTSyn

Upstream requirements: SRS_BSW_00337, SRS_BSW_00385, SRS_BSW_00323, SRS_BSW_00406

⌈

Type of error	*Related error code*	*Error value*
API service used in un-initialized state	ETHTSYN_E_UNINIT	0x20
EthTSyn initialization failed	ETHTSYN_E_INIT_FAILED	0x21
API called with invalid controller index	ETHTSYN_E_CTRL_IDX	0x22
API called with invalid pointer	ETHTSYN_E_PARAM_POINTER	0x23

▽

△

Type of error	Related error code	Error value
API called with invalid parameter	ETHTSYN_E_PARAM	0x24

⌋

7.13.2 Runtime Errors

[SWS_EthTSyn_00144] Definition of runtime errors in module EthTSyn

Upstream requirements: SRS_BSW_00385

⌈

Type of error	Related error code	Error value
Time Master conflict	ETHTSYN_E_TMCONFLICT	0x01
Time Slave conflict	ETHTSYN_E_TSCONFLICT	0x02
No FV available from the FVM	ETHTSYN_E_FRESHNESSFAILURE	0x03

⌋

7.13.3 Production Errors

No Production Errors defined.

8 API specification

8.1 Imported types

In this section all types included from the following modules are listed:

[SWS_EthTSyn_00031] Definition of imported datatypes of module EthTSyn

Upstream requirements: RS_TS_20048, RS_TS_20059

⌈

Module	*Header File*	*Imported Type*
Comtype	ComStack_Types.h	PduIdType
	ComStack_Types.h	PduInfoType
	ComStack_Types.h	PduLengthType
	ComStack_Types.h	RateDeviationStatusType (draft)
	ComStack_Types.h	RateDeviationWithStatusType (draft)
	ComStack_Types.h	TimeStampQualType (draft)
	ComStack_Types.h	TimeStampType (draft)
	ComStack_Types.h	TimeTupleType (draft)
Csm	Rte_Csm_Type.h	Crypto_OperationModeType
	Rte_Csm_Type.h	Crypto_ResultType
	Rte_Csm_Type.h	Crypto_VerifyResultType
Eth	Eth_GeneralTypes.h	Eth_PhcCorrectionResultType
	Eth_GeneralTypes.h	Eth_PhcCorrectionType
EthSwt	Eth_GeneralTypes.h	EthSwt_MgmtInfoType
	Eth_GeneralTypes.h	EthSwt_MgmtObjectType
	Eth_GeneralTypes.h	EthSwt_MgmtObjectValidType
	Eth_GeneralTypes.h	EthSwt_MgmtOwner
EthTrcv	Eth_GeneralTypes.h	EthTrcv_LinkStateType
IdsM	IdsM.h	IdsM_SecurityEventIdType
	Rte_IdsM_Type.h	IdsM_TimestampDataType
StbM	Rte_StbM_Type.h	StbM_EthTimeMasterMeasurementType
	Rte_StbM_Type.h	StbM_EthTimeSlaveMeasurementType
	Rte_StbM_Type.h	StbM_PdelayInitiatorMeasurementType
	Rte_StbM_Type.h	StbM_PdelayResponderMeasurementType
	Rte_StbM_Type.h	StbM_PortIdType
	Rte_StbM_Type.h	StbM_ProtocolParamType
	Rte_StbM_Type.h	StbM_SynchronizedTimeBaseType
	Rte_StbM_Type.h	StbM_TimeBaseStatusType
	Rte_StbM_Type.h	StbM_TimeStampShortType
	Rte_StbM_Type.h	StbM_TimeStampType
	Rte_StbM_Type.h	StbM_TimeSyncType
	Rte_StbM_Type.h	StbM_TimeTupleType

▽

△

Module	Header File	Imported Type
	Rte_StbM_Type.h	StbM_UserDataType
	StbM.h	StbM_MeasurementType
	StbM.h	StbM_VirtualLocalTimeType
Std	Std_Types.h	Std_ReturnType

⌋

8.2 Type definitions

8.2.1 EthTSyn_ConfigType

[SWS_EthTSyn_00032] Definition of datatype EthTSyn_ConfigType

Upstream requirements: RS_TS_20048

⌈

Name	EthTSyn_ConfigType	
Kind	Structure	
Elements	implementation specific	
	Type	–
	Comment	–
Description	This is the base type for the configuration of the Global Time Synchronization over Ethernet. A pointer to an instance of this structure will be used in the initialization of the Global Time Synchronization over Ethernet. The content of this structure is defined in chapter 10 Configuration specification.	
Available via	EthTSyn.h	

⌋

8.2.2 EthTSyn_TransmissionModeType

[SWS_EthTSyn_00033] Definition of datatype EthTSyn_TransmissionModeType

Upstream requirements: RS_TS_20048

⌈

Name	EthTSyn_TransmissionModeType		
Kind	Enumeration		
Range	ETHTSYN_TX_OFF	0x00	Transmission Disabled
	ETHTSYN_TX_ON	0x01	Transmission Enabled
Description	Handles the enabling and disabling of the transmission mode		
Available via	EthTSyn.h		

⌋

8.3 Function definitions

This chapter contains all APIs provided by the module.

8.3.1 EthTSyn_Init

[SWS_EthTSyn_00035] Definition of API function EthTSyn_Init

Upstream requirements: RS_TS_20048

⌈

Service Name	EthTSyn_Init	
Syntax	`void EthTSyn_Init ( const EthTSyn_ConfigType* configPtr )`	
Service ID [hex]	0x01	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	configPtr	Pointer to selected configuration structure
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	This function initializes the Time Synchronization over Ethernet.	
Available via	EthTSyn.h	

⌋

See section 7.1.1 for details.

8.3.2 EthTSyn_SetTransmissionMode

[SWS_EthTSyn_00039] Definition of API function EthTSyn_SetTransmissionMode

Upstream requirements: RS_TS_20048

⌈

Service Name	EthTSyn_SetTransmissionMode	
Syntax	`void EthTSyn_SetTransmissionMode ( uint8 CtrlIdx, EthTSyn_TransmissionModeType Mode )`	
Service ID [hex]	0x05	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	CtrlIdx	Index of the Ethernet controller
	Mode	ETHTSYN_TX_OFF ETHTSYN_TX_ON

▽

△

Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	This API is used to turn on and off the TX capabilities of the EthTSyn.
Available via	EthTSyn.h

⌋

[SWS_EthTSyn_00172]

Upstream requirements: SRS_BSW_00323, SRS_BSW_00337, SRS_BSW_00386, SRS_BSW_00452

⌈The function `EthTSyn_SetTransmissionMode` shall inform the DET, if development error detection is enabled (EthTSynDevErrorDetect (ECUC_EthTSyn_00002 :) is set to TRUE) and if function call has failed because of the following reasons:

- CtrlIdx is invalid (ETHTSYN_E_CTRL_IDX)
- Mode is invalid (ETHTSYN_E_PARAM)

⌋

8.3.3 EthTSyn_SetProtocolParam

[SWS_EthTSyn_00330] Definition of API function EthTSyn_SetProtocolParam

Upstream requirements: RS_TS_20069

⌈

Service Name	EthTSyn_SetProtocolParam	
Syntax	`Std_ReturnType EthTSyn_SetProtocolParam ( StbM_SynchronizedTimeBaseType timeBaseId, const StbM_ProtocolParamType* protocolParam )`	
Service ID [hex]	0x0a	
Sync/Async	Synchronous	
Reentrancy	Conditionally Reentrant (non-reentrant for same TimeBaseId, reentrant for different TimeBase Ids)	
Parameters (in)	timeBaseId	ID of the synchronized time base
	protocolParam	structure with Follow_Up information TLV parameters
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: successful `E_NOT_OK`: failed
Description	This API is used to set FollowUp information TLV parameters of a Follow_Up message prior transmission. The API is called within StbM_SetBusProtocolParam which provides the content of the structure protocolParam.	
Available via	EthTSyn.h	

⌋

[SWS_EthTSyn_00228]

Upstream requirements: SRS_BSW_00323, SRS_BSW_00337, SRS_BSW_00386, SRS_BSW_00452

⌈The function EthTSyn_SetProtocolParam() shall inform the DET, if development error detection is enabled (EthTSynDevErrorDetect (ECUC_EthTSyn_00002 :) is set to TRUE) and if function call has failed because of the following reasons:

- timeBaseId does not belong to a Time Base, which is mapped to a Time Domain with ID 0 ..127 in EthTSyn (Development Error: ETHTSYN_E_PARAM)
- protocolParam is NULL (Development Error: ETHTSYN_E_PARAM_POINTER)

⌋

8.3.4 EthTSyn_GetProtocolParam

[SWS_EthTSyn_00331] Definition of API function EthTSyn_GetProtocolParam

Upstream requirements: RS_TS_20069

⌈

Service Name	EthTSyn_GetProtocolParam	
Syntax	`Std_ReturnType EthTSyn_GetProtocolParam ( StbM_SynchronizedTimeBaseType timeBaseId, StbM_ProtocolParamType* protocolParam )`	
Service ID [hex]	0x0b	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	timeBaseId	ID of the synchronized time base
Parameters (inout)	None	
Parameters (out)	protocolParam	structure to store received Follow_Up information TLV parameters
Return value	Std_ReturnType	`E_OK`: successful `E_NOT_OK`: failed
Description	This API is used to read FollowUp information TLV parameters from received Follow_Up message.	
Available via	EthTSyn.h	

⌋

[SWS_EthTSyn_00229]

Upstream requirements: SRS_BSW_00323, SRS_BSW_00337, SRS_BSW_00386, SRS_BSW_00452

⌈The function EthTSyn_GetProtocolParam() shall inform the DET, if development error detection is enabled (EthTSynDevErrorDetect (ECUC_EthTSyn_00002 :) is set to TRUE) and if function call has failed because of the following reasons:

- timeBaseId does not belong to a Time Base, which is mapped to a Time Domain with ID 0 ..127 in EthTSyn (Development Error: ETHTSYN_E_PARAM)
- protocolParam is NULL (Development Error: ETHTSYN_E_PARAM_POINTER)

⌋

8.3.5 EthTSyn_GetCurrentTimeTuple

[SWS_EthTSyn_91003] Definition of API function EthTSyn_GetCurrentTimeTuple ⌈

Service Name	EthTSyn_GetCurrentTimeTuple	
Syntax	`Std_ReturnType EthTSyn_GetCurrentTimeTuple ( uint8 timeDomain, TimeTupleType timeTuplePtr )`	
Service ID [hex]	0x0e	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	timeDomain	Index of the time domain.
Parameters (inout)	None	
Parameters (out)	timeTuplePtr	Current time information. timestampClockValue represents the free running clock.
Return value	Std_ReturnType	`E_OK`: OK: successful `E_NOT_OK`: failed
Description	Returns the current time tuple from the corresponding timestamping clock.	
Available via	EthTSyn.h	

⌋

8.3.6 EthTSyn_SetAndGetPhcCorrection

[SWS_EthTSyn_91004] Definition of API function EthTSyn_SetAndGetPhcCorrection ⌈

Service Name	EthTSyn_SetAndGetPhcCorrection	
Syntax	`Std_ReturnType EthTSyn_SetAndGetPhcCorrection ( uint8 timeDomain, Eth_PhcCorrectionType* phcCorrectionPtr, Eth_PhcCorrectionResultType phcCorrectionResultPtr )`	
Service ID [hex]	0x0f	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	timeDomain	Index of the time domain.
	phcCorrectionPtr	Rate deviation and offset time by which the PHC shall be corrected.
Parameters (inout)	None	
Parameters (out)	phcCorrectionResultPtr	Actually applied correction.
Return value	Std_ReturnType	`E_OK`: successful `E_NOT_OK`: failed

▽

△

Description	Triggers correction of the PHC used as reference time for the time domain and retrieves the performed correction.
Available via	EthTSyn.h

⌋

8.3.7 EthTSyn_RxIndication

[SWS_EthTSyn_00040] Definition of callback function EthTSyn_RxIndication

Upstream requirements: RS_TS_20048

⌈

Service Name	EthTSyn_RxIndication	

Syntax	`void EthTSyn_RxIndication (`
	`  PduIdType RxPduId,`
	`  const PduInfoType* PduInfoPtr`
	`)`	

Service ID [hex]	0x42	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in)	RxPduId	ID of the received PDU.
	PduInfoPtr	Contains the length (SduLength) of the received PDU, a pointer to a buffer (SduDataPtr) containing the PDU, and the MetaData related to this PDU.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Indication of a received PDU from a lower layer communication interface module.	
Available via	EthTSyn.h	

⌋

[SWS_EthTSyn_00041]

Upstream requirements: SRS_BSW_00337, SRS_BSW_00323

⌈The callback function EthTSyn_RxIndication() shall inform the DET, if development error detection is enabled (EthTSynDevErrorDetect (ECUC_EthTSyn_00002 :) is set to TRUE) and if the function call has failed because of the following reasons:

- `RxPduId` is invalid (`ETHTSYN_E_PARAM`)
- `PduInfoPtr` is invalid (`ETHTSYN_E_PARAM_POINTER`)

⌋

8.3.8 EthTSyn_TxConfirmation

[SWS_EthTSyn_00042] Definition of callback function EthTSyn_TxConfirmation

Upstream requirements: RS_TS_20048

⌈

Service Name	EthTSyn_TxConfirmation	
Syntax	`void EthTSyn_TxConfirmation ( PduIdType TxPduId, Std_ReturnType result )`	
Service ID [hex]	0x40	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in)	TxPduId	ID of the PDU that has been transmitted.
	result	`E_OK`: The PDU was transmitted successfully. `E_NOT_OK`: Transmission of the PDU failed.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	The lower layer communication interface module confirms the transmission of a PDU, or the failure to transmit a PDU.	
Available via	EthTSyn.h	

⌋

[SWS_EthTSyn_00175]

Upstream requirements: SRS_BSW_00323, SRS_BSW_00337

⌈The function EthTSyn_TxConfirmation() shall inform the DET, if development error detection is enabled (EthTSynDevErrorDetect (ECUC_EthTSyn_00002 :) is set to TRUE) and if function call has failed because of the following reasons:

- `TxPduId` is invalid (`ETHTSYN_E_PARAM`)

⌋

[SWS_EthTSyn_00176]

Upstream requirements: SRS_BSW_00323, SRS_BSW_00337

⌈On invocation of EthTSyn_TxConfirmation() with parameter 'Result' equal to E_NOT_OK the process of collection of synchronized time distribution shall be aborted and all intermediate result variables shall be reset to default value.⌋

8.3.9 EthTSyn_TrcvLinkStateChg

[SWS_EthTSyn_00043] Definition of callback function EthTSyn_TrcvLinkStateChg

Upstream requirements: RS_TS_20048

⌈

Service Name	EthTSyn_TrcvLinkStateChg	
Syntax	`void EthTSyn_TrcvLinkStateChg ( uint8 CtrlIdx, EthTrcv_LinkStateType TrcvLinkState )`	
Service ID [hex]	0x08	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	CtrlIdx	Index of the Ethernet controller
	TrcvLinkState	ETHTRCV_LINK_STATE_DOWN ETHTRCV_LINK_STATE_ACTIVE
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Allows resetting state machine in case of unexpected Link loss to avoid inconsistent Sync and Follow_Up sequences	
Available via	EthTSyn.h	

⌋

[SWS_EthTSyn_00174]

Upstream requirements: SRS_BSW_00323, SRS_BSW_00337

⌈If function `EthTSyn_TrcvLinkStateChg` is called with a `CtrlIdx` not referenced by the `EthTSyn`, then the `EthTSyn` shall ignore the call.

⌋

8.3.10 EthTSyn_IcvGenerationIndication

[SWS_EthTSyn_91001] Definition of API function EthTSyn_IcvGenerationIndication

Upstream requirements: RS_TS_20072

⌈

Service Name	EthTSyn_IcvGenerationIndication
Syntax	`void EthTSyn_IcvGenerationIndication ( uint32 jobId, Crypto_ResultType result )`
Service ID [hex]	0x0c

▽

△

Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	jobId	JobID of the operation that caused the callback.
	result	Contains the result of the cryptographic operation.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	By this API service the EthTSyn gets an indication and the result of ICV generation.	
Available via	EthTSyn.h	

⌋

[SWS_EthTSyn_00259]

Upstream requirements: SRS_BSW_00323, SRS_BSW_00337

⌈The function EthTSyn_IcvGenerationIndication() shall inform the DET, if development error detection is enabled (`EthTSynDevErrorDetect` is set to TRUE) and if function call has failed because of the following reasons:

- jobId is invalid (ETHTSYN_E_PARAM)

⌋

8.3.11 EthTSyn_IcvVerificationIndication

[SWS_EthTSyn_91002] Definition of API function EthTSyn_IcvVerificationIndication

Upstream requirements: RS_TS_20072

⌈

Service Name	EthTSyn_IcvVerificationIndication	
Syntax	`void EthTSyn_IcvVerificationIndication ( uint32 jobId, Crypto_ResultType result )`	
Service ID [hex]	0x0d	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	jobId	JobID of the operation that caused the callback.
	result	Contains the result of the cryptographic operation.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	By this API service the EthTSyn gets an indication and the result of ICV verification.	
Available via	EthTSyn.h	

⌋

[SWS_EthTSyn_00260]

Upstream requirements: SRS_BSW_00323, SRS_BSW_00337

⌈The function EthTSyn_IcvVerificationIndication() shall inform the DET, if development error detection is enabled (`EthTSynDevErrorDetect` is set to TRUE) and if function call has failed because of the following reasons:

- jobId is invalid (ETHTSYN_E_PARAM)

⌋

8.4 Scheduled functions

The Basic Software Scheduler directly calls these functions. The following functions shall have no return value and no parameters. All functions shall be non-reentrant.

8.4.1 EthTSyn_MainFunction

[SWS_EthTSyn_00044] Definition of scheduled function EthTSyn_MainFunction

Upstream requirements: RS_TS_20048

⌈

Service Name	EthTSyn_MainFunction
Syntax	`void EthTSyn_MainFunction ( void )`
Service ID [hex]	0x09
Description	Main function for cyclic call / resp. Sync, Follow_Up and Pdelay_Req transmissions
Available via	EthTSyn_SchM.h

⌋

[SWS_EthTSyn_00045]

Upstream requirements: RS_TS_20048

⌈The frequency of invocations of `EthTSyn_MainFunction` is determined by the configuration parameter EthTSynMainFunctionPeriod (ECUC_EthTSyn_00012 :).⌋

8.5 Expected Interfaces

In this section, all interfaces required from other modules are listed.

8.5.1 Mandatory Interfaces

There are no mandatory interfaces defined.

8.5.2 Optional Interfaces

This section defines all interfaces that are required to fulfill an optional functionality of the module.

[SWS_EthTSyn_00047] Definition of optional interfaces requested by module Eth TSyn

Upstream requirements: RS_TS_20048, RS_TS_20059

⌈

API Function	*Header File*	*Description*
Crc_CalculateCRC8H2F	Crc.h	This service makes a CRC8 calculation with the Polynomial 0x2F on Crc_Length
Csm_MacGenerate	Csm.h	Uses the given data to perform a MAC generation and stores the MAC in the memory location pointed to by the MAC pointer.
Csm_MacVerify	Csm.h	Verifies the given MAC by comparing if the MAC is generated with the given data.
Csm_SignatureGenerate	Csm.h	Uses the given data to perform the signature calculation and stores the signature in the memory location pointed by the result pointer.
Csm_SignatureVerify	Csm.h	Verifies the given signature by checking if it was generated with the given data.
Det_ReportError	Det.h	Service to report development errors.
Det_ReportRuntimeError	Det.h	Service to report runtime errors. If a callout has been configured then this callout shall be called.
EthIf_EnableEgressTimeStamp	EthIf.h	Activates egress time stamping on a dedicated message object. Some HW does store once the egress time stamp marker and some HW needs it always before transmission. There will be no “disable” functionality, due to the fact, that the message type is always "time stamped" by network design.
EthIf_GetEgressTimeStamp	EthIf.h	Reads back the egress time stamp on a dedicated message object. It must be called within the Tx Confirmation() function.
EthIf_GetTxMgmtObject	EthIf.h	Request the MgmtObject of the (in this context) unique BufIdx.
EthIf_SetAndGetPhcCorrection	EthIf.h	Triggers correction of the PHC used as reference time for the given controller and clock unit, and retrieves the performed correction.
EthIf_SetSwitchMgmtInfo	EthIf.h	Provides additional management information along to an Ethernet frame that requires special treatment within the Switch. For direct data provision, it has to be called before the transmit request is called. For indirect data provision, it can also be called in the context of the TriggerTransmit API.
EthIf_SwitchEnableTimeStamping	EthIf.h	Activates egress time stamping on a dedicated message object, addressed addressed by the PduId which is associated with an Ethernet controller index and an egress queue.
IdsM_ReportSecurityEvent	IdsM.h	This API is the interface to report security events to the IdsM.
LSduR_EthTSynTransmit (draft)	LSduR_EthTSyn.h	Requests transmission of a PDU.

▽

△

API Function	*Header File*	*Description*
StbM_BusSetGlobalTime	StbM.h	Allows the Time Base Provider Modules to forward the Rx Time Tuple to the StbM.
StbM_EthSetMasterTimingData	StbM_EthTSyn.h	Provides Ethernet Timesyn module specific data for a Time Master to the StbM.
StbM_EthSetPdelayInitiatorData	StbM_EthTSyn.h	–
StbM_EthSetPdelayResponderData	StbM_EthTSyn.h	–
StbM_EthSetSlaveTimingData	StbM_EthTSyn.h	Allows the EthTSyn Module to forward Ethernet specific details to the StbM.
StbM_GetCurrentTime	StbM.h	Returns a time tuple (Local time, Global time and Timebase status) and user data details Note: This API shall be called with locked interrupts / within an Exclusive Area to prevent interruption (i.e., the risk that the time stamp is outdated on return of the function call).
StbM_GetCurrentVirtualLocalTime	StbM.h	Returns the Virtual Local Time of the referenced Time Base.
StbM_GetRxFreshness	StbM.h	This interface is used by the StbM to query the current freshness value.
StbM_GetTimeBaseStatus	StbM.h	Returns detailed status information for a Synchronized (or Pure Local) Time Base.
StbM_GetTimeBaseUpdateCounter	StbM.h	Allows the Timesync Modules to detect, whether a Time Base should be transmitted immediately in the subsequent <Bus>TSyn_MainFunction() cycle.
StbM_GetTxFreshness	StbM.h	This API returns the freshness value from the Most Significant Bits in the first byte, of the Freshness array, in big endian format.
StbM_GetTxFreshnessTruncData	StbM.h	This interface is used by the StbM to obtain the current freshness value. The interface function provides also the truncated freshness transmitted in the secured time sync message.
StbM_SPduTxConfirmation	StbM.h	This interface is used by the StbM to indicate that the Secured Time Synchronization Message has been initiated for transmission.

⌋

8.5.3 Callout Interfaces

There are no callout interfaces defined.

8.5.4 Configurable Interfaces

There are no configurable interfaces defined.

8.6 Service Interfaces

There are no service interfaces defined.

9 Sequence diagrams

9.1 EthIf_EnableEgressTimeStamp

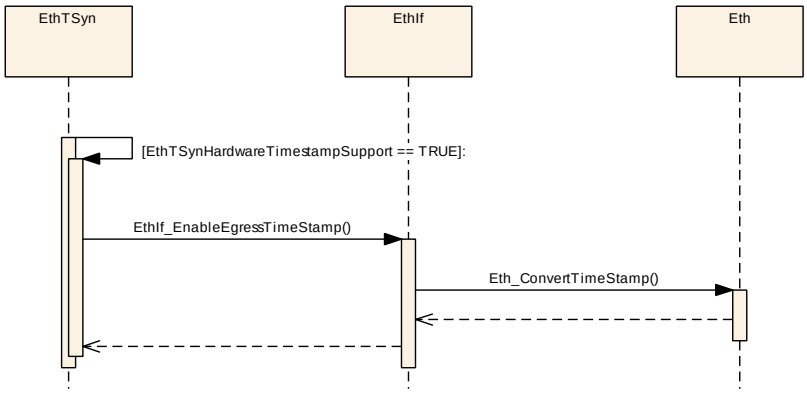


Figure 9.1: EnableEgressTimeStamp

9.2 Time Synchronization Sequence

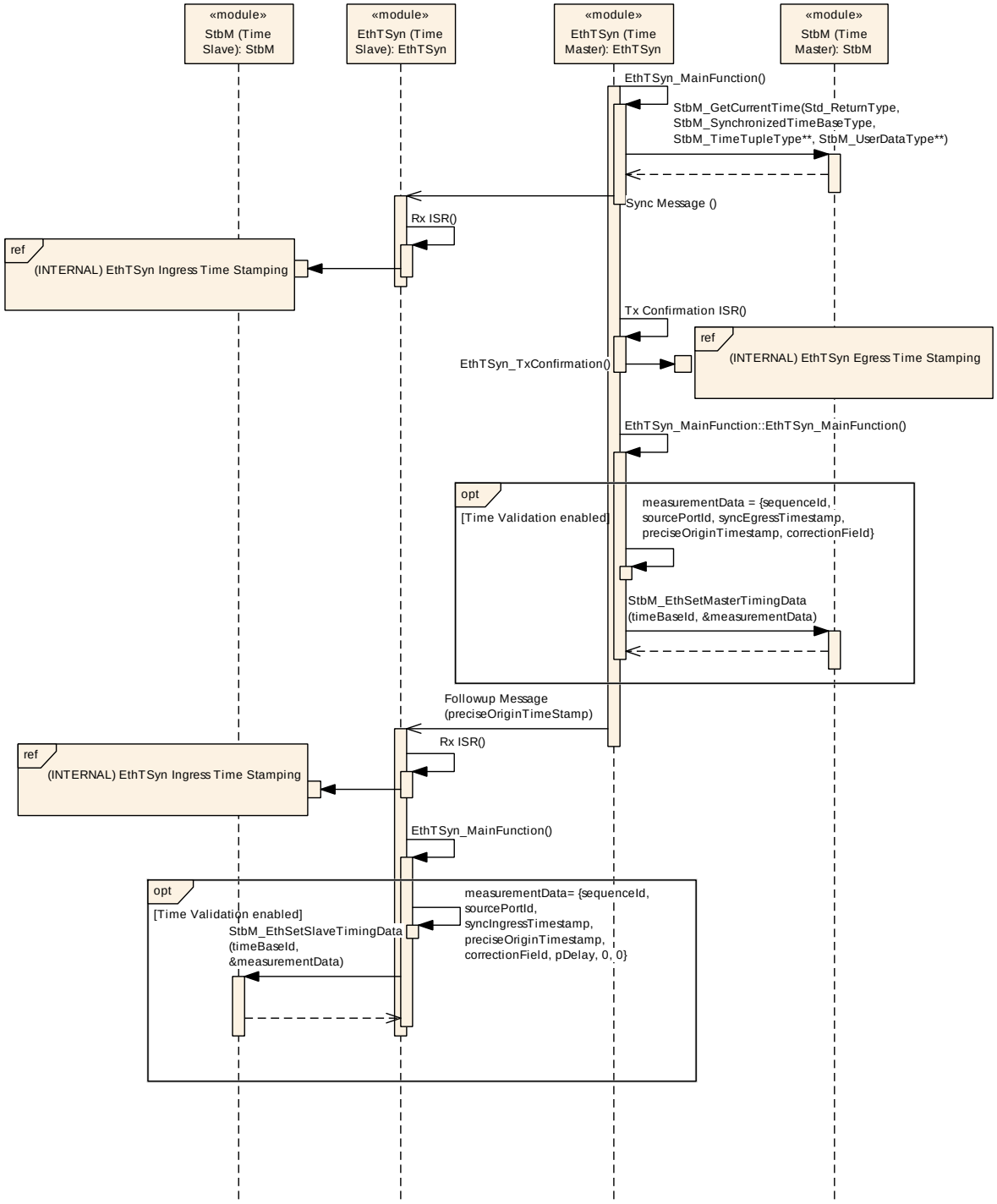


Figure 9.2: : Time Synchronization Sequence

9.3 Pdelay Measurement Sequence

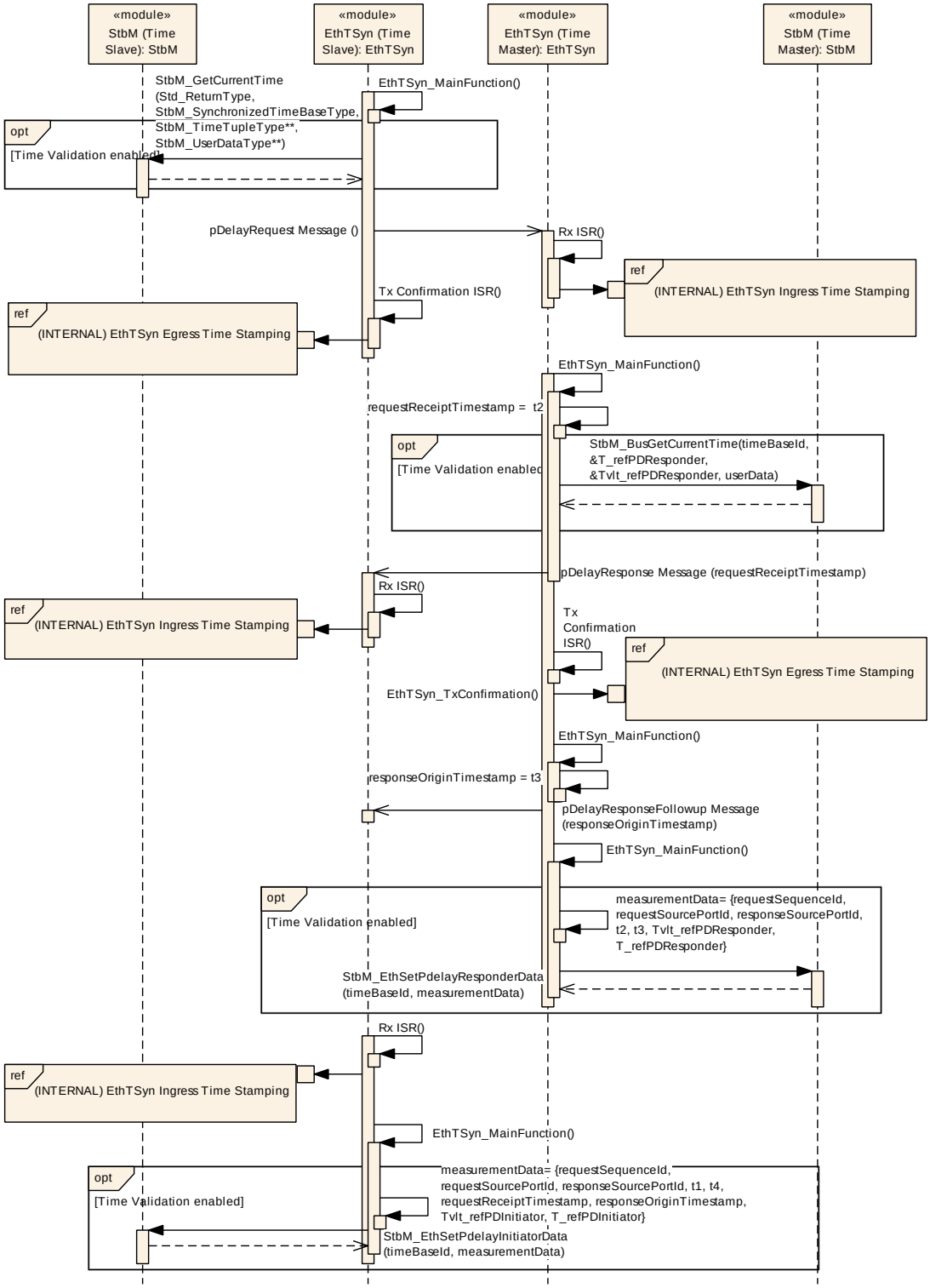


Figure 9.3: : Pdelay Sequence

9.4 EthTSyn Egress Timestamping

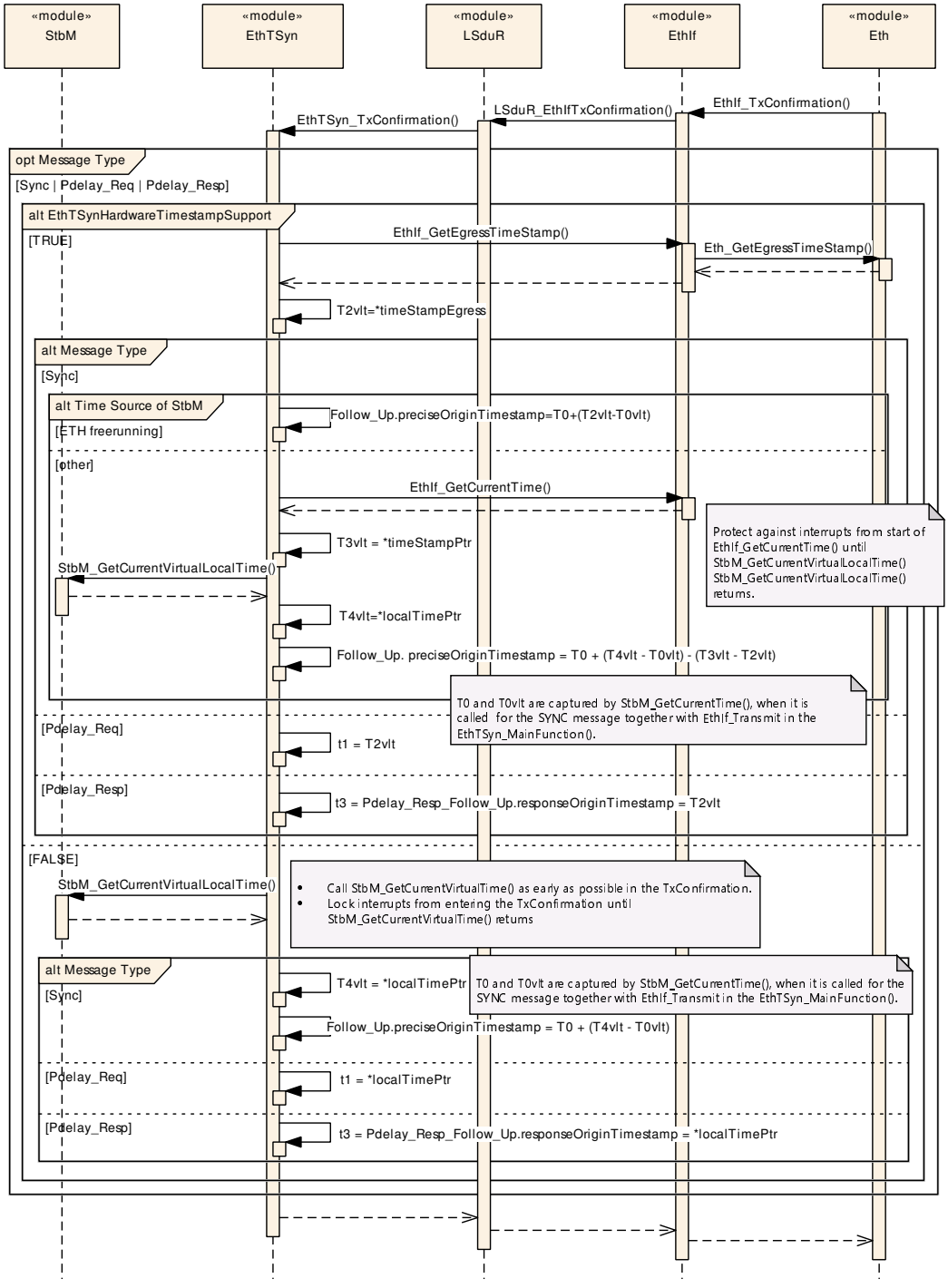


Figure 9.4: EthTSynEgressTimestamping

9.5 EthTSyn Ingress Timestamping

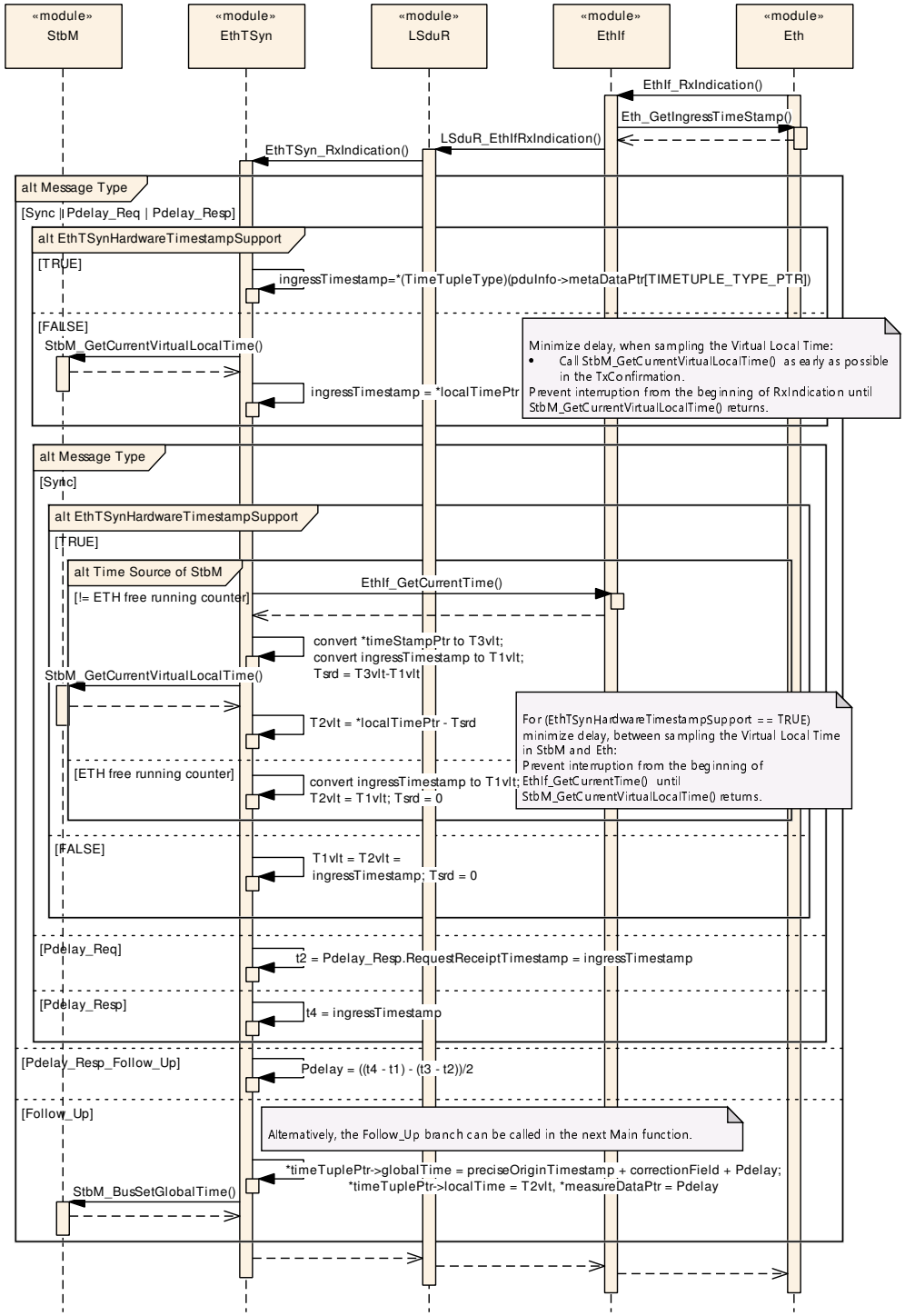


Figure 9.5: EthTSyn Ingress Timestamping

9.6 Time measurement with Switches

9.6.1 Time Aware Bridge with GTM as Management CPU - Tx

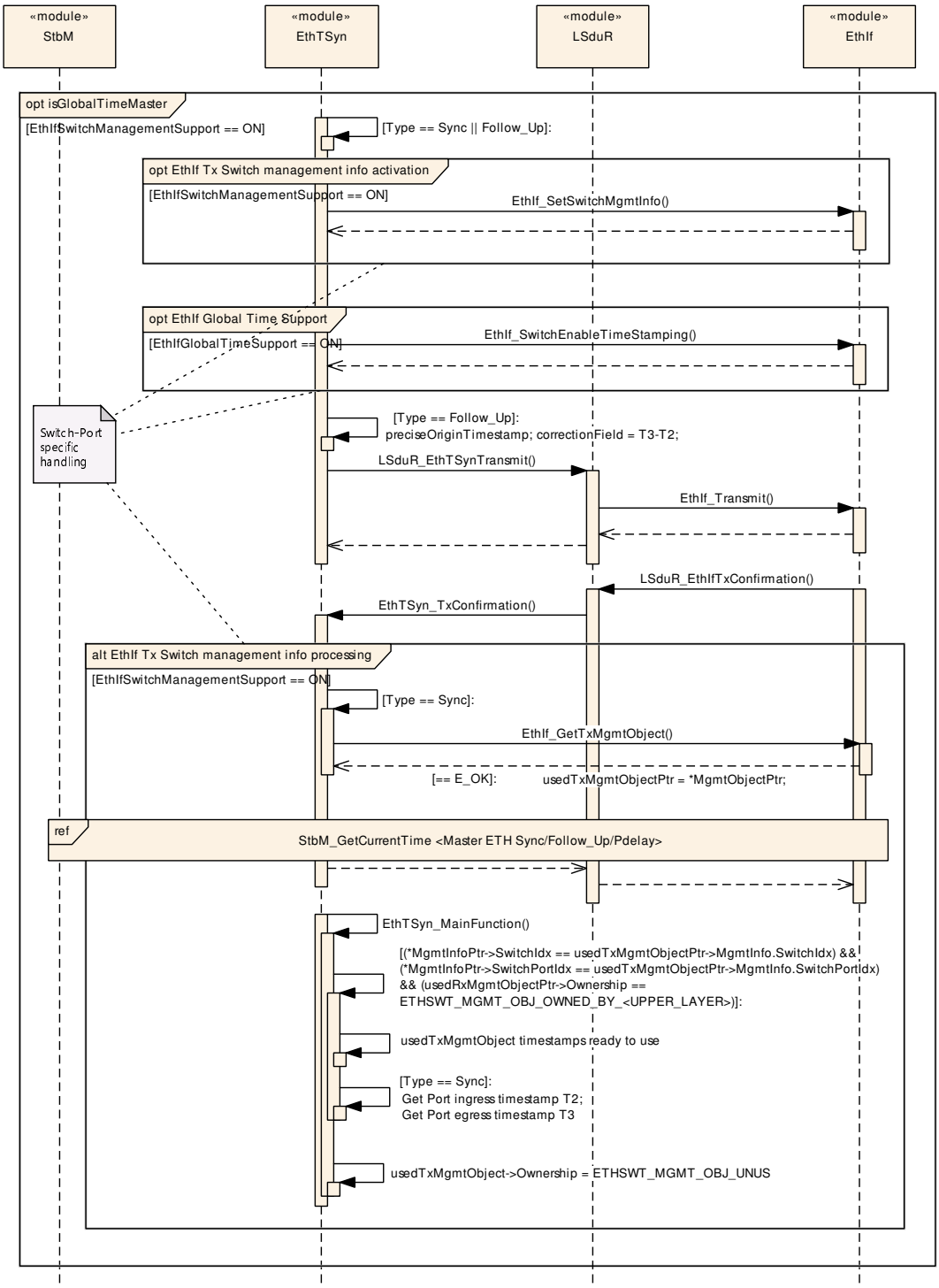


Figure 9.6: Time Aware Bridge with GTM as Management CPU Sync_Up Follow_Up Tx

9.6.2 Time Aware Bridge without GTM as Management CPU - Tx

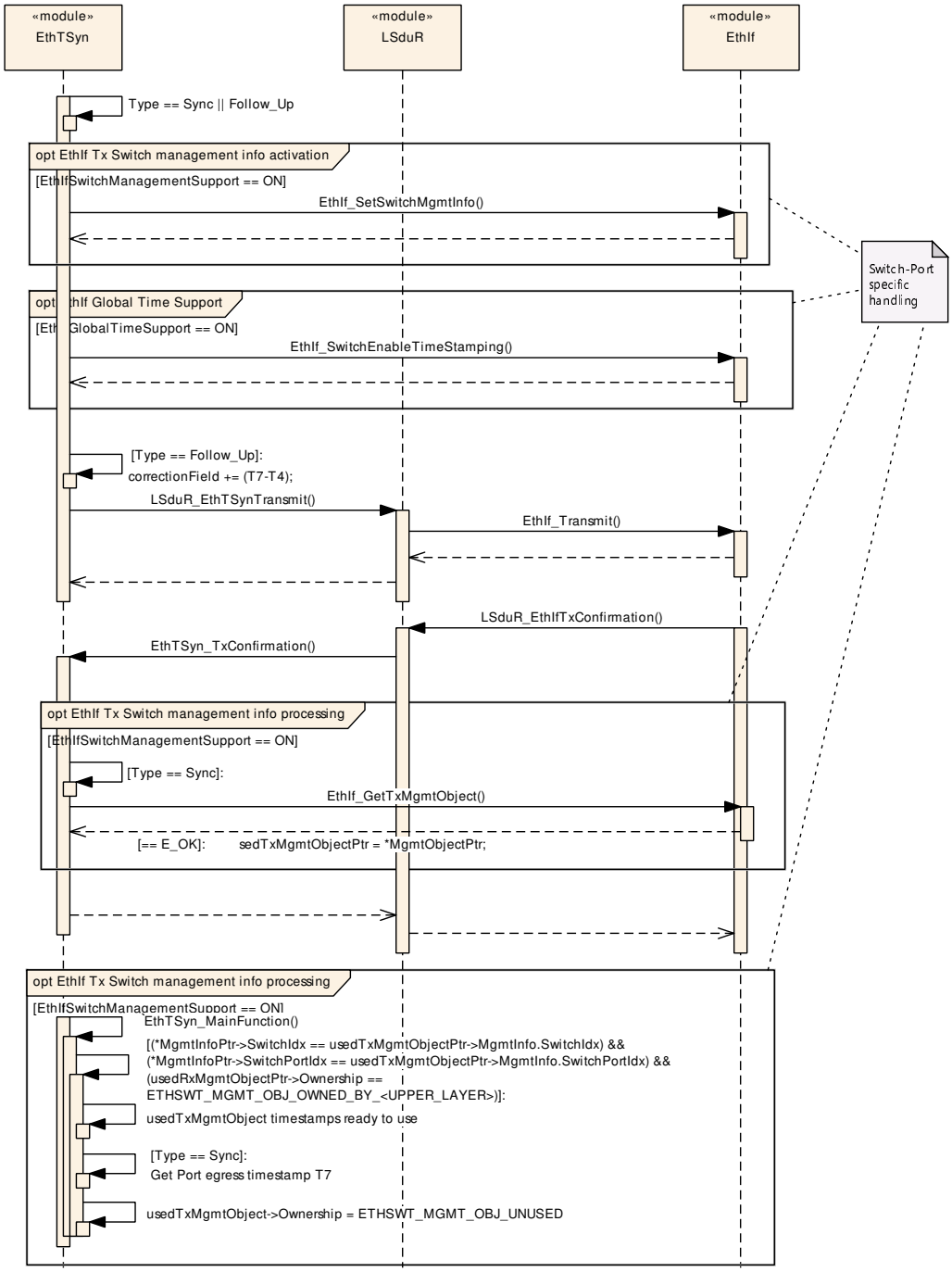


Figure 9.7: EthTSyn_SwitchWithoutGTM_Sync_Follow_Up_Tx

9.6.3 Time Aware Bridge without GTM as Management CPU - Rx

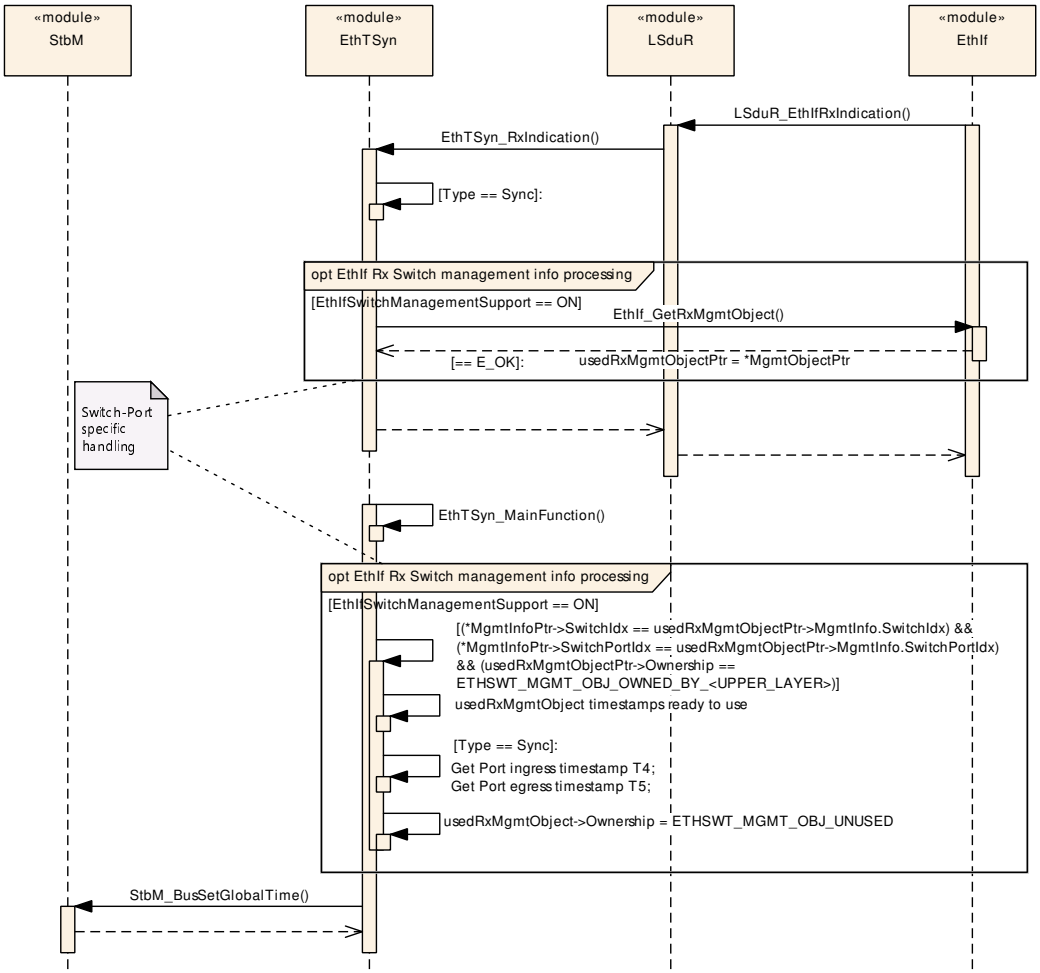


Figure 9.8: EthTSyn_SwitchWithoutGTM_Sync_Follow_Up_Rx

9.7 Secure Time Synchronization Sequence

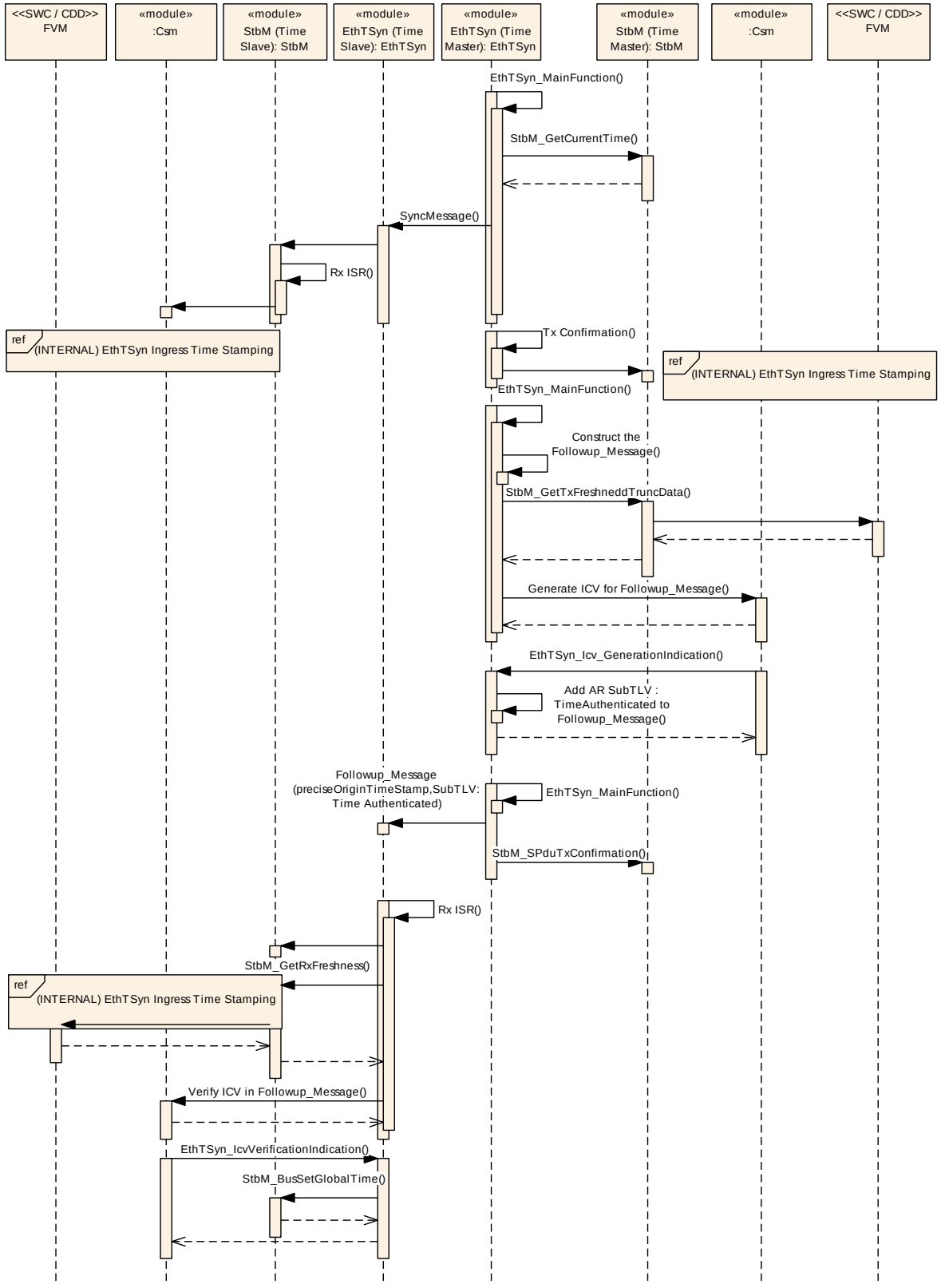


Figure 9.9: Secure Time Synchronization Sequence

10 Configuration specification

In general, this chapter defines configuration parameters and their clustering into containers. In order to support the specification chapter 10.1 describes fundamentals. It also specifies a template (table) you shall use for the parameter specification. We intend to leave chapter 10.1 in the specification to guarantee comprehension.

Chapter 10.2 specifies the structure (containers) and the parameters of the module EthTSyn.

Chapter 10.5 specifies published information of the module EthTSyn.

10.1 How to read this chapter

For details refer to [3] Chapter 10.1 *"Introduction to configuration specification"*.

[SWS_EthTSyn_00051]

Upstream requirements: RS_TS_20052, RS_TS_20053, RS_TS_20054

⌈The EthTSyn module shall support the configuration for Time Master, Time Slave and Time Gateway.⌋

10.2 Containers and configuration parameters

The following sections summarize all configuration parameters of the Global Time Synchronization over Ethernet. The detailed meaning of the parameters is described in chapters Chapter 7 and Chapter 8.

The module supports different post-build variants (previously known as post-build selectable configuration sets), but not post-build loadable configuration.

10.2.1 EthTSyn

[ECUC_EthTSyn_00001] Definition of EcucModuleDef EthTSyn ⌈

Module Name	EthTSyn
Description	Configuration of the Timesync over Ethernet (EthTSyn) module with respect to global time synchronization.
Post-Build Selectable Support	true
Post-Build Loadable Support	false

Included Containers		
Container Name	**Multiplicity**	**Description**
EthTSynCMLDS	0..*	Configuration of the Common Mean Link Delay Service (CMLDS). There is at most one instance of the CMLDS per physical port. For each physical port the CMLDS uses a single virtual port.
EthTSynGeneral	1	This container holds the general parameters of the Ethernet specific timesync module (EthTSyn).
EthTSynGlobalTimeDomain	1..*	This represents a global time domain on Ethernet. The EthTSyn module can manage several global time domains. Only one container shall be configured per global time domain ID. If the EthTSyn exists it is assumed that at least one global time domain exists.
EthTSynGlobalTimeMasterConfig	0..*	Configuration of a global-time master. The exact role of the port is defined by EthTSynPortRole. A global-time master may or may not exist on the configured ECU. Each global-time domain may have multiple ports in the time master role (see Time Gateway) to e.g. relay global-time from a global time slave port.
EthTSynGlobalTimeSlaveConfig	0..*	Configuration of a global-time slave. The exact role of the port is defined by EthTSynPortRole. A TIME_SLAVE port will exist on every configured ECU, except potentially on the SystemWideGlobalTimeMaster. Each global-time domain should have at least one port of one ECU in the TIME_MASTER role and one port of another ECU in the TIME_SLAVE role.
EthTSynPTPPortDetails	1..*	Configuration of a PTP Port. The same configuration can be used by multiple PTP ports through referencing this configuration.
EthTSynPdelayConfig	0..*	Configuration of cyclic propagation delay measurement. This can be used by either a PTP port or the CMLDS.
EthTSynPhysicalPortConfiguration	1..*	Configuration of a physical port. The same configuration can be used by multiple physical ports through referencing this configuration.
EthTSynVirtualPortConfiguration	1..*	Configuration of a virtual port (VLAN) associated with a single physical port and used by different PTP and/or CMLDS ports to transmit and receive PTP messages.

⌋

Module dependencies of the EthTSyn is shown in Figure 5.1.

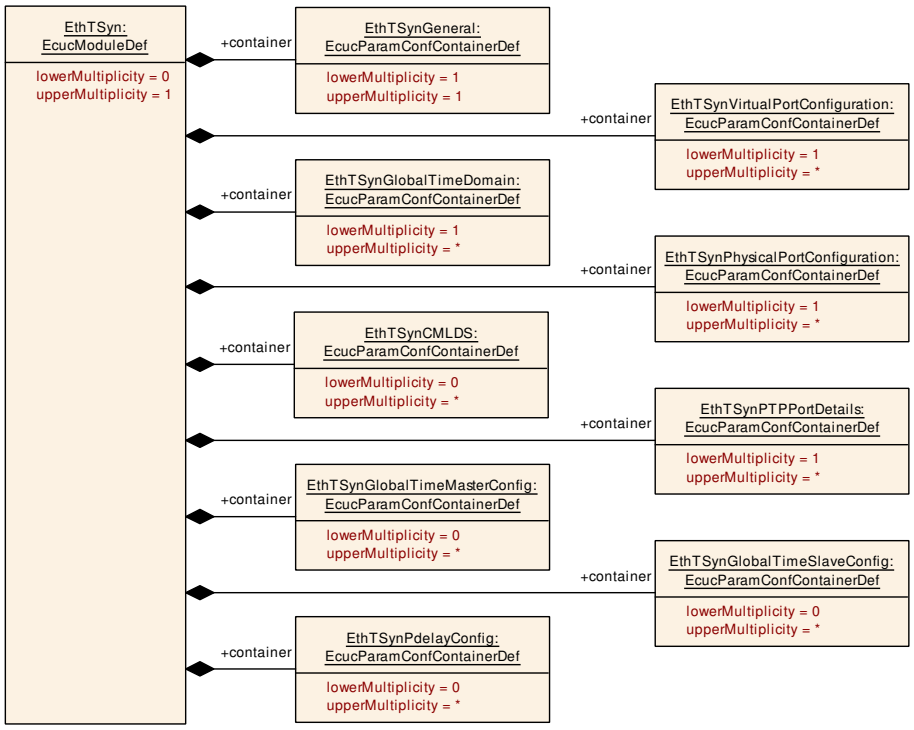


Figure 10.1: EthTSyn

10.2.2 EthTSynGeneral

[ECUC_EthTSyn_00003] Definition of EcucParamConfContainerDef EthTSyn General ⌈

Container Name	EthTSynGeneral
Parent Container	EthTSyn
Upstream mapping defined	–
Description	This container holds the general parameters of the Ethernet specific timesync module (EthTSyn).
Multiplicity	1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTSynDevErrorDetect	1	[ECUC_EthTSyn_00002]
EthTSynEnableSecurityEventReporting	1	[ECUC_EthTSyn_00089]
EthTSynMainFunctionPeriod	1	[ECUC_EthTSyn_00012]
EthTSynMaxNumberOfTransmitRetries	1	[ECUC_EthTSyn_00120]
EthTSynSwitchMgmtRxMessageBufferCount	0..1	[ECUC_EthTSyn_00059]
EthTSynTimeValidationSupport	1	[ECUC_EthTSyn_00081]
EthTSynEthIfFrameType	1	[ECUC_EthTSyn_00127]

Included Containers		
Container Name	**Multiplicity**	**Description**
EthTSynSecurityEventRefs	0..1	Container for the references to IdsMEvent elements representing the security events that the EthTSyn module shall report to the IdsM in case the coresponding security related event occurs (and if EthTSynEnableSecurityEventReportings set to "true"). The standardized security events in this container can be extended by vendor-specific security events.

⌋

[ECUC_EthTSyn_00002] Definition of EcucBooleanParamDef EthTSynDevError Detect ⌈

Parameter Name	EthTSynDevErrorDetect
Parent Container	EthTSynGeneral
Upstream mapping defined	–
Description	Switches the development error detection and notification on or off. • true: detection and notification is enabled. • false: detection and notification is disabled.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00089] Definition of EcucBooleanParamDef EthTSynEnableSecurityEventReporting ⌈

Parameter Name	EthTSynEnableSecurityEventReporting
Parent Container	EthTSynGeneral
Upstream mapping defined	–
Description	Switches the reporting of security events to the IdsM: - true: reporting is enabled. - false: reporting is disabled.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00012] Definition of EcucFloatParamDef EthTSynMainFunctionPeriod ⌈

Parameter Name	EthTSynMainFunctionPeriod	
Parent Container	EthTSynGeneral	
Upstream mapping defined	–	
Description	Schedule period of the main function EthTSyn_MainFunction. Unit: seconds.	
Multiplicity	1	
Type	EcucFloatParamDef	
Range	]0 .. INF[	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_EthTSyn_00120] Definition of EcucIntegerParamDef EthTSynMaxNumberOfTransmitRetries

Status: DRAFT

⌈

Parameter Name	EthTSynMaxNumberOfTransmitRetries	
Parent Container	EthTSynGeneral	
Upstream mapping defined	–	
Description	Provide the max number of re-transmission of frames in case of transmission failure(i.e., no buffer available or driver busy). A value of 0 will disable the retry mechanism. **Tags:** atp.Status=draft	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 20	
Default value	0	
Pre-Compile only	true	

⌋

[ECUC_EthTSyn_00059] Definition of EcucIntegerParamDef EthTSynSwitchMgmtRxMessageBufferCount ⌈

Parameter Name	EthTSynSwitchMgmtRxMessageBufferCount
Parent Container	EthTSynGeneral
Upstream mapping defined	–
Description	This parameter is used to determine the amount of Rx message buffers available in the EthTSyn when EthTSyn is used in a Bridge configuration.
Multiplicity	0..1
Type	EcucIntegerParamDef
Range	1 .. 254
Default value	10
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00081] Definition of EcucBooleanParamDef EthTSynTimeValidationSupport ⌈

Parameter Name	EthTSynTimeValidationSupport
Parent Container	EthTSynGeneral
Upstream mapping defined	–
Description	Switches support for time validation on or off. • true: time validation is enabled. • false: time validation is disabled.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00127] Definition of EcucReferenceDef EthTSynEthIfFrameType

Status: DRAFT

⌈

Parameter Name	EthTSynEthIfFrameType
Parent Container	EthTSynGeneral
Upstream mapping defined	–
Description	The chosen frame owner determines which frames (in respect to ethertype) are received. **Tags:** atp.Status=draft
Multiplicity	1
Type	Reference to EthIfFrameConfig
Pre-Compile only	true

⌋

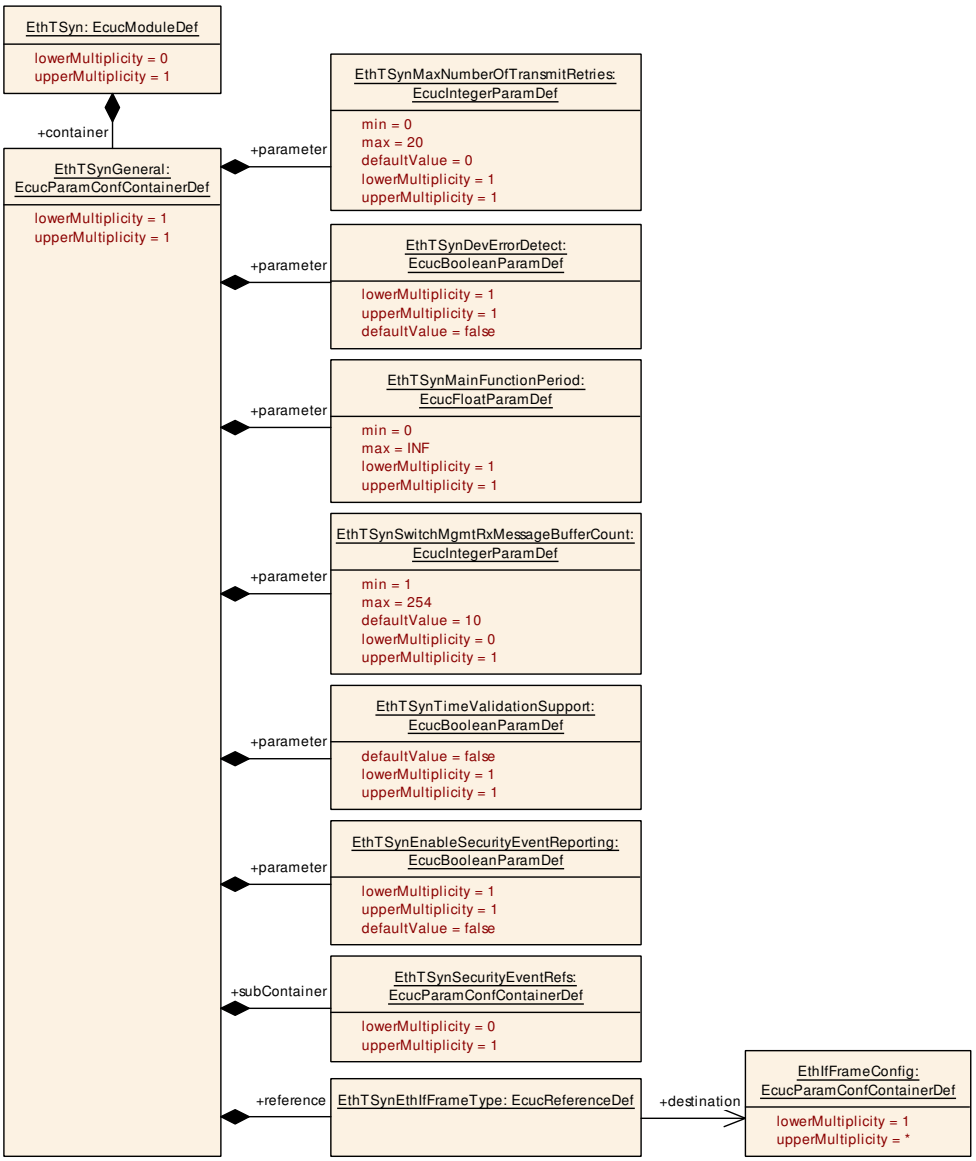


Figure 10.2: EthTSynGeneral

10.2.3 EthTSynSecurityEventRefs

[ECUC_EthTSyn_00090] Definition of EcucParamConfContainerDef EthTSynSecurityEventRefs ⌈

Container Name	EthTSynSecurityEventRefs
Parent Container	EthTSynGeneral
Upstream mapping defined	–

▽

△

Description	Container for the references to IdsMEvent elements representing the security events that the EthTSyn module shall report to the IdsM in case the coresponding security related event occurs (and if EthTSynEnableSecurityEventReportings set to "true"). The standardized security events in this container can be extended by vendor-specific security events.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
SEV_TSYN_ETH_FRESHNESS_NOT_AVAILABLE	0..1	[ECUC_EthTSyn_00093]
SEV_TSYN_ETH_ICV_GENERATION_FAILED	0..1	[ECUC_EthTSyn_00091]
SEV_TSYN_ETH_ICV_VERIFICATION_FAILED	0..1	[ECUC_EthTSyn_00092]
SEV_TSYN_ETH_SYNC_FUP_SEQUENCE_ERROR	0..1	[ECUC_EthTSyn_00114]

No Included Containers

⌋

[ECUC_EthTSyn_00093] Definition of EcucReferenceDef SEV_TSYN_ETH_FRESHNESS_NOT_AVAILABLE ⌈

Parameter Name	SEV_TSYN_ETH_FRESHNESS_NOT_AVAILABLE
Parent Container	EthTSynSecurityEventRefs
Upstream mapping defined	–
Description	FV not available from FVM. Context data provides the respective domain ID.
Multiplicity	0..1
Type	Symbolic name reference to IdsMEvent
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00091] Definition of EcucReferenceDef SEV_TSYN_ETH_ICV_GENERATION_FAILED ⌈

Parameter Name	SEV_TSYN_ETH_ICV_GENERATION_FAILED
Parent Container	EthTSynSecurityEventRefs
Upstream mapping defined	–
Description	ICV generation for Follow_Up message failed. Context data provides the respective domain ID
Multiplicity	0..1
Type	Symbolic name reference to IdsMEvent
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00092] Definition of EcucReferenceDef SEV_TSYN_ETH_ICV_VERIFICATION_FAILED ⌈

Parameter Name	SEV_TSYN_ETH_ICV_VERIFICATION_FAILED
Parent Container	EthTSynSecurityEventRefs
Upstream mapping defined	–
Description	ICV verification for Follow_Up message failed. Context data provides the respective domain ID.
Multiplicity	0..1
Type	Symbolic name reference to IdsMEvent
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00114] Definition of EcucReferenceDef SEV_TSYN_ETH_SYNC_FUP_SEQUENCE_ERROR ⌈

Parameter Name	SEV_TSYN_ETH_SYNC_FUP_SEQUENCE_ERROR
Parent Container	EthTSynSecurityEventRefs
Upstream mapping defined	–
Description	Failed to receive correct sequence of SYNC and Follow_Up from the TimeMaster within (EthTSynGlobalTimeFollowUpTimeout).
Multiplicity	0..1
Type	Symbolic name reference to IdsMEvent
Pre-Compile only	true

⌋

10.2.4 EthTSynGlobalTimeDomain

[ECUC_EthTSyn_00004] Definition of EcucParamConfContainerDef EthTSyn GlobalTimeDomain ⌈

Container Name	EthTSynGlobalTimeDomain	
Parent Container	EthTSyn	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	This represents a global time domain on Ethernet. The EthTSyn module can manage several global time domains. Only one container shall be configured per global time domain ID. If the EthTSyn exists it is assumed that at least one global time domain exists.	
Multiplicity	1..*	
Post-Build Multiplicity	-	Selectable
Configuration Parameters		

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTSynGlobalTimeDebounceTime	0..1	[ECUC_EthTSyn_00048]
EthTSynGlobalTimeDomainId	1	[ECUC_EthTSyn_00005]
EthTSynGlobalTimeRxDebounceTime	0..1	[ECUC_EthTSyn_00094]
EthTSynSynchronizedTimeBaseRef	1	[ECUC_EthTSyn_00013]

Included Containers		
Container Name	**Multiplicity**	**Description**
EthTSynGlobalTimeFollowUpDataIDList	0..1	The DataIDList for Follow_Up message ensures the identification of data elements due to CRC calculation and message authentication process.
EthTSynPTPPort	1..*	Configuration of the EthTSynPTPPorts within the TimeDomain.

⌋

[ECUC_EthTSyn_00048] Definition of EcucFloatParamDef EthTSynGlobalTime DebounceTime ⌈

Parameter Name	EthTSynGlobalTimeDebounceTime	
Parent Container	EthTSynGlobalTimeDomain	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	This represents the configuration of a TX debounce time for Sync, Follow_Up, and p Delay messages compared to a message before with the same PDU. Unit: seconds	
Multiplicity	0..1	
Type	EcucFloatParamDef	
Range	[0 .. 4]	
Default value	–	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00005] Definition of EcucIntegerParamDef EthTSynGlobalTimeDomainId ⌈

Parameter Name	EthTSynGlobalTimeDomainId	
Parent Container	EthTSynGlobalTimeDomain	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	The global time domain ID.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00094] Definition of EcucFloatParamDef EthTSynGlobalTimeRxDebounceTime ⌈

Parameter Name	EthTSynGlobalTimeRxDebounceTime	
Parent Container	EthTSynGlobalTimeDomain	
Upstream mapping defined	–	
Description	This represents the configuration of a RX debounce time for Sync and Follow_Up. Unit: seconds	
Multiplicity	0..1	
Type	EcucFloatParamDef	
Range	[0 .. 4]	
Default value	–	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00013] Definition of EcucReferenceDef EthTSynSynchronizedTimeBaseRef ⌈

Parameter Name	EthTSynSynchronizedTimeBaseRef
Parent Container	EthTSynGlobalTimeDomain
Upstream mapping defined	–
Description	Mandatory reference to the required synchronized time-base.
Multiplicity	1
Type	Symbolic name reference to StbMSynchronizedTimeBase
Pre-Compile only	true

⌋

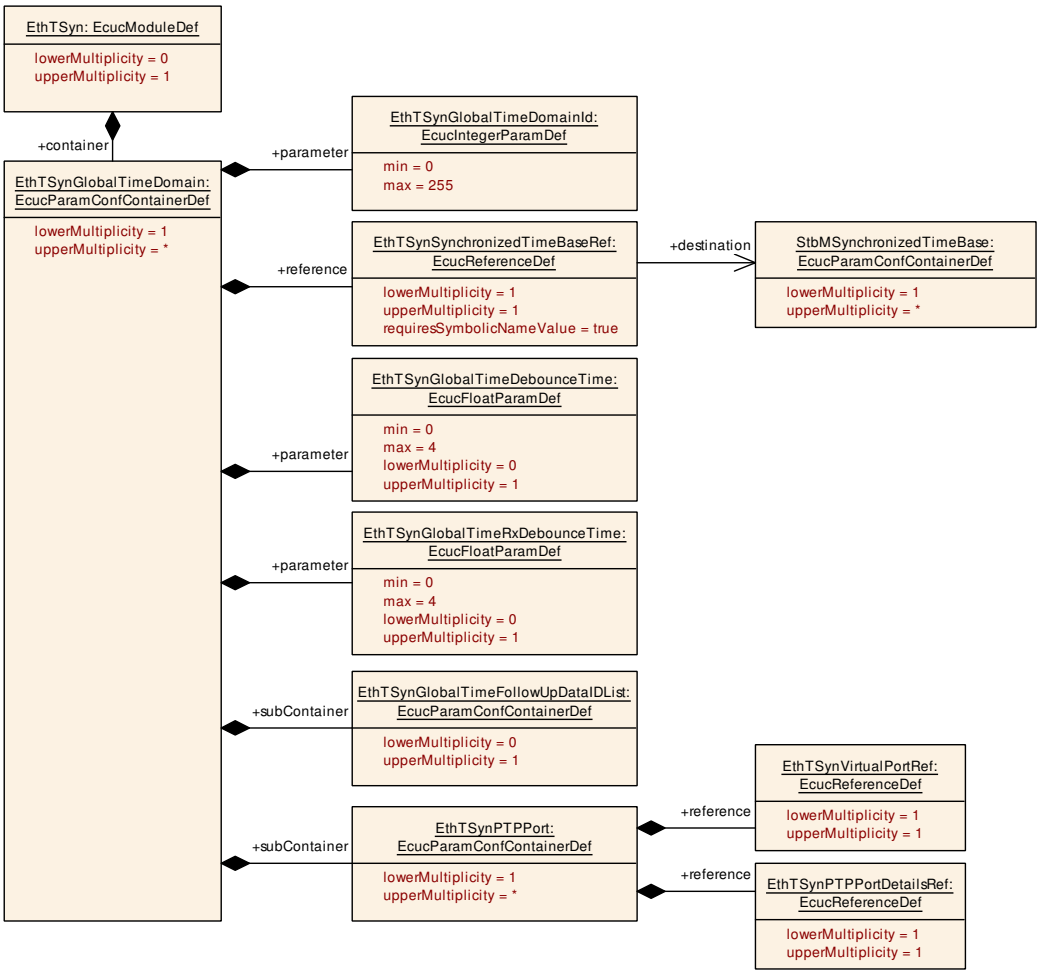


Figure 10.3: EthTSynGlobalTimeDomain

10.2.5 EthTSynGlobalTimeFollowUpDataIDList

[ECUC_EthTSyn_00030] Definition of EcucParamConfContainerDef EthTSynGlobalTimeFollowUpDataIDList ⌈

Container Name	EthTSynGlobalTimeFollowUpDataIDList	
Parent Container	EthTSynGlobalTimeDomain	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	The DataIDList for Follow_Up message ensures the identification of data elements due to CRC calculation and message authentication process.	
Multiplicity	0..1	
Post-Build Multiplicity	-	Selectable
Configuration Parameters		

No Included Parameters

Included Containers		
Container Name	**Multiplicity**	**Description**
EthTSynGlobalTimeFollowUpDataIDListElement	16	Element of the DataIDList for Follow_Up message ensures the identification of data elements due to CRC calculation and message authentication process.

⌋

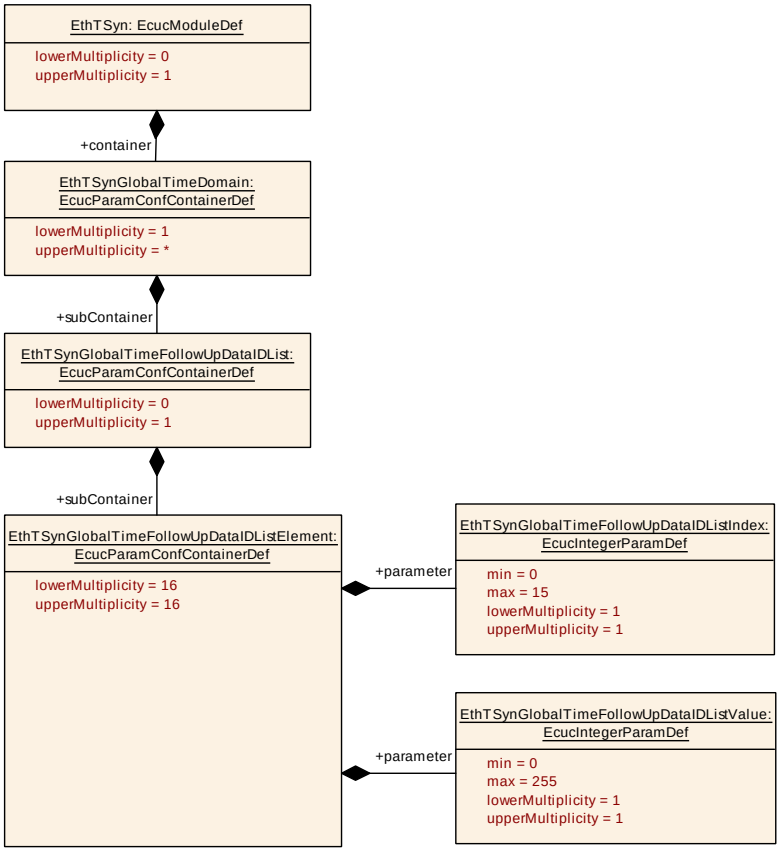


Figure 10.4: EthTSyn_GlobalTimeFollowUpDataIDList

10.2.6 EthTSynGlobalTimeFollowUpDataIDListElement

[ECUC_EthTSyn_00031] Definition of EcucParamConfContainerDef EthTSynGlobalTimeFollowUpDataIDListElement ⌈

Container Name	EthTSynGlobalTimeFollowUpDataIDListElement
Parent Container	EthTSynGlobalTimeFollowUpDataIDList
Upstream mapping defined	[13, CP_TPS_SystemTemplate]
Description	Element of the DataIDList for Follow_Up message ensures the identification of data elements due to CRC calculation and message authentication process.
Multiplicity	16
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTSynGlobalTimeFollowUpDataIDListIndex	1	[ECUC_EthTSyn_00032]
EthTSynGlobalTimeFollowUpDataIDListValue	1	[ECUC_EthTSyn_00033]

No Included Containers

⌋

[ECUC_EthTSyn_00032] Definition of EcucIntegerParamDef EthTSynGlobalTimeFollowUpDataIDListIndex ⌈

Parameter Name	EthTSynGlobalTimeFollowUpDataIDListIndex	
Parent Container	EthTSynGlobalTimeFollowUpDataIDListElement	
Upstream mapping defined	–	
Description	Index of the DataIDList for Follow_Up message ensures the identification of data elements due to CRC calculation and message authentication process.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 15	
Default value	–	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00033] Definition of EcucIntegerParamDef EthTSynGlobalTimeFollowUpDataIDListValue ⌈

Parameter Name	EthTSynGlobalTimeFollowUpDataIDListValue	
Parent Container	EthTSynGlobalTimeFollowUpDataIDListElement	
Upstream mapping defined	–	
Description	Value of the DataIDList for Follow_Up message ensures the identification of data elements due to CRC calculation and message authentication process.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
Default value	–	
Post-Build Value	-	Selectable

⌋

10.2.7 EthTSynPTPPort

[ECUC_EthTSyn_00063] Definition of EcucParamConfContainerDef EthTSynPTPPort ⌈

Container Name	EthTSynPTPPort	
Parent Container	EthTSynGlobalTimeDomain	
Upstream mapping defined	–	
Description	Configuration of the EthTSynPTPPorts within the TimeDomain.	
Multiplicity	1..*	
Post-Build Multiplicity	-	Selectable
Configuration Parameters		

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTSynConfiguredMasterSourcePortIdentity	0..1	[ECUC_EthTSyn_00128]
EthTSynLinkDelayConfig	1	[ECUC_EthTSyn_00131]
EthTSynNeighbourRateRatioConfig	1	[ECUC_EthTSyn_00130]
EthTSynPortRole	1	[ECUC_EthTSyn_00116]
EthTSynPTPPortDetailsRef	1	[ECUC_EthTSyn_00132]
EthTSynVirtualPortRef	1	[ECUC_EthTSyn_00133]

No Included Containers

⌋

[ECUC_EthTSyn_00128] Definition of EcucStringParamDef EthTSynConfigured MasterSourcePortIdentity

Status: DRAFT

⌈

Parameter Name	EthTSynConfiguredMasterSourcePortIdentity
Parent Container	EthTSynPTPPort
Upstream mapping defined	–
Description	The SourcePortIdentity (clockIdentity + portNumber) which the sourcePortIdentity of the received Sync frame is compared to. Input format has to match xx:xx:xx:xx:xx:xx:xx:xx:xx:xx, where x stands for a hex value between 0 and F. **Tags:** atp.Status=draft
Multiplicity	0..1
Type	EcucStringParamDef
Default value	–
Regular Expression	[0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){9}
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00131] Definition of EcucEnumerationParamDef EthTSynLink DelayConfig ⌈

Parameter Name	EthTSynLinkDelayConfig
Parent Container	EthTSynPTPPort
Upstream mapping defined	–

▽

△

Description	Selects the mechanism through which this PTPPort determines the LinkDelay. Different mechanisms may use the same configuration as given in the PTPPortDetails.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	CMLDS	The link delay is determined through the Common Mean Link Delay Service (CMLDS) for this virtual port.
	OFF	No LinkDelay is determined for this PTPPort.
	TransportSpecific	The LinkDelay is determined through the TransportSpecific pDelay mechanism.
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00130] Definition of EcucEnumerationParamDef EthTSyn NeighbourRateRatioConfig ⌈

Parameter Name	EthTSynNeighbourRateRatioConfig	
Parent Container	EthTSynPTPPort	
Upstream mapping defined	–	
Description	Sets the mechanism through which the NeighbourRateRatio is determined.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	CMLDS	The NeighbourRateRatio is determined through the Common Mean Link Delay Service (CMLDS).
	DriftTrackingTLV	The NeighbourRateRatio is determined through the DriftTrackingTLV mechanism.
	OFF	No NeighbourRateRatio is determined on this PTPPort.
	PHC	The NeighbourRateRatio is determined through a mechanism for a PHC.
	TransportSpecific	The NeighbourRateRatio is determined through the TransportSpecific pDelay mechanism.
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00116] Definition of EcucEnumerationParamDef EthTSynPort Role ⌈

Parameter Name	EthTSynPortRole	
Parent Container	EthTSynPTPPort	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	Parameter to set the port's role to TimeSlave, TimeMaster, or Dynamic (i.e., TimeSlave or TimeMaster is determined at runtime).	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DYNAMIC	The port's role is determined at runtime to be either TIME_MASTER or TIME_SLAVE.

▽

△

	TIME_MASTER	This PTPPort transmits timing information for the Domain it is associated with.
	TIME_SLAVE	This PTPPort receives timing information for the Domain it is associated with. A Domain can only receive timing information on one single Port on a specific ECU.
Default value	DYNAMIC	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00132] Definition of EcucReferenceDef EthTSynPTPPortDetailsRef ⌈

Parameter Name	EthTSynPTPPortDetailsRef	
Parent Container	EthTSynPTPPort	
Upstream mapping defined	–	
Description	This represents the reference to the Ethernet interface taken to fetch the global time information. If EthTSynPortRole = DYNAMIC, then two EthTSynPTPPortDetails may be referenced in parallel to reflect dedicated configurations for the TimeMaster and the TimeSlave role.	
Multiplicity	1	
Type	Reference to EthTSynPTPPortDetails	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00133] Definition of EcucReferenceDef EthTSynVirtualPortRef ⌈

Parameter Name	EthTSynVirtualPortRef
Parent Container	EthTSynPTPPort
Upstream mapping defined	–
Description	This represents the reference to the virtual Ethernet interface taken to fetch the global time information.
Multiplicity	1
Type	Reference to EthTSynVirtualPortConfiguration
Pre-Compile only	true

⌋

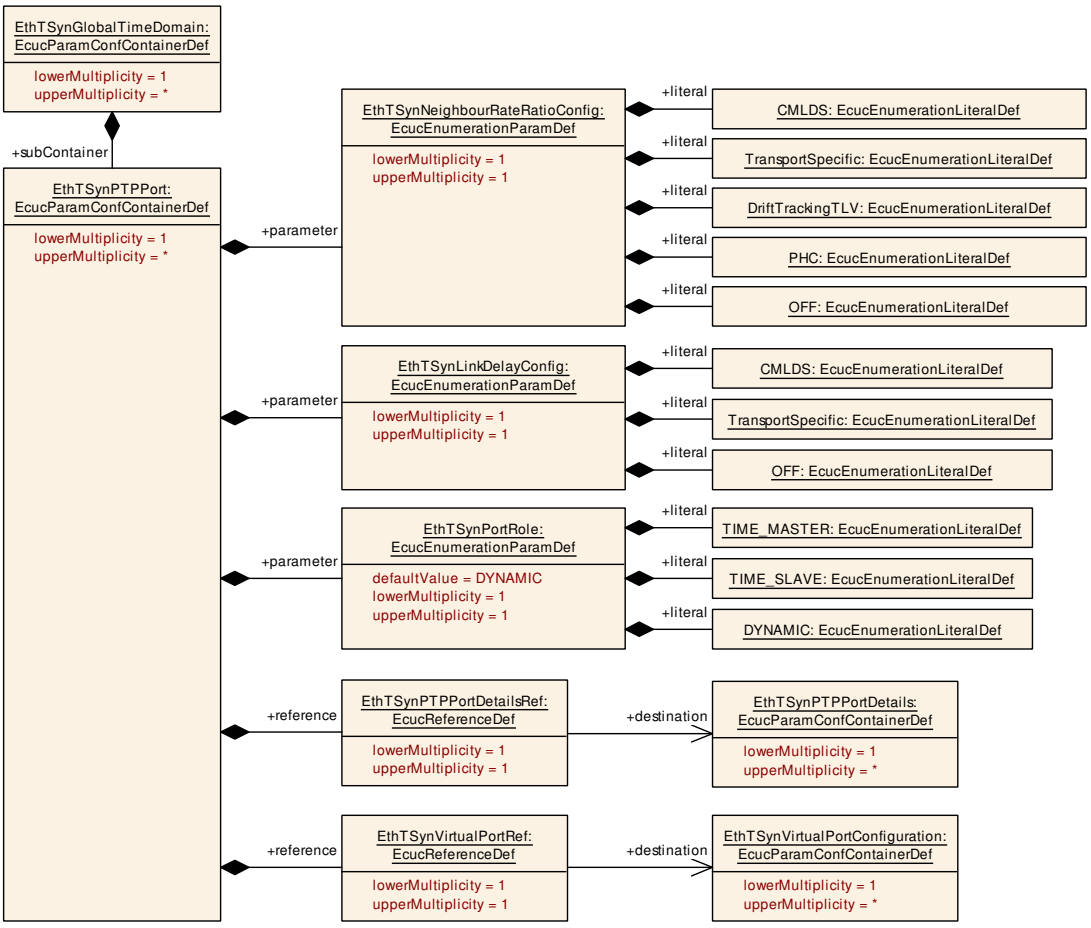


Figure 10.5: EthTSynPTPPort

10.2.8 EthTSynPTPPortDetails

[ECUC_EthTSyn_00145] Definition of EcucParamConfContainerDef EthTSynPTPPortDetails ⌈

Container Name	EthTSynPTPPortDetails	
Parent Container	EthTSyn	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	Configuration of a PTP Port. The same configuration can be used by multiple PTP ports through referencing this configuration.	
Multiplicity	1..*	
Post-Build Multiplicity	-	Selectable
Configuration Parameters		

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTSynDestPhyAddr	1	[ECUC_EthTSyn_00058]
EthTSynEnableTimeValidation	1	[ECUC_EthTSyn_00082]
EthTSynFramePrio	0..1	[ECUC_EthTSyn_00034]
EthTSynMasterSlaveConflictDetection	1	[ECUC_EthTSyn_00075]
EthTSynMessageCompliance	1	[ECUC_EthTSyn_00029]
EthTSynClkUnitRef	0..1	[ECUC_EthTSyn_00119]
EthTSynPdelayConfigRef	0..1	[ECUC_EthTSyn_00134]
EthTSynTimeMasterConfigRef	0..1	[ECUC_EthTSyn_00135]
EthTSynTimeSlaveConfigRef	0..1	[ECUC_EthTSyn_00136]

Included Containers		
Container Name	**Multiplicity**	**Description**
EthTSynGlobalTimeRxIcv Verification	0..1	This container collects configuration that shall be used for ICV verification.
EthTSynGlobalTimeTxIcv Generation	0..1	This container collects configuration that shall be used for ICV generation.

⌋

[ECUC_EthTSyn_00058] Definition of EcucStringParamDef EthTSynDestPhyAddr ⌈

Parameter Name	EthTSynDestPhyAddr
Parent Container	EthTSynPTPPortDetails
Upstream mapping defined	[13, CP_TPS_SystemTemplate]
Description	Destination Physical Address (MAC-Address). Destination Physical Hardware Address (MAC-Address) of EthTSyn-gPTP Frames. Input format has to match xx:xx:xx:xx:xx:xx, where x stands for a hex value between 0 and F.
Multiplicity	1
Type	EcucStringParamDef
Default value	01:80:C2:00:00:0E
Regular Expression	–
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00082] Definition of EcucBooleanParamDef EthTSynEnableTimeValidation ⌈

Parameter Name	EthTSynEnableTimeValidation
Parent Container	EthTSynPTPPortDetails
Upstream mapping defined	–
Description	Enables/disables time recording for time validation for a port of a specific Time Domain. • true: recording is enabled. • false: recording is disabled.
Multiplicity	1

▽

△

Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00034] Definition of EcucIntegerParamDef EthTSynFramePrio ⌈

Parameter Name	EthTSynFramePrio	
Parent Container	EthTSynPTPPortDetails	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	This optional parameter, if present, indicates the priority of outgoing EthTSyn message (i.e., it equals the 3-bit PCP field of a tagged VLAN message). If a VLAN is not configured, this parameter is also not configured.	
Multiplicity	0..1	
Type	EcucIntegerParamDef	
Range	0 .. 7	
Default value	–	
Post-Build Multiplicity	-	Selectable
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00075] Definition of EcucBooleanParamDef EthTSynMasterSlaveConflictDetection ⌈

Parameter Name	EthTSynMasterSlaveConflictDetection
Parent Container	EthTSynPTPPortDetails
Upstream mapping defined	–
Description	Enables master / slave conflict detection and notification. • true: detection and notification is enabled. • false: detection and notification is disabled.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00029] Definition of EcucBooleanParamDef EthTSynMessageCompliance ⌈

Parameter Name	EthTSynMessageCompliance
Parent Container	EthTSynPTPPortDetails
Upstream mapping defined	[13, CP_TPS_SystemTemplate]
Description	• true: IEEE 802.1AS compliant message format will be used. • false: IEEE 802.1AS message format with AUTOSAR extension will be used.

▽

△

Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00119] Definition of EcucReferenceDef EthTSynClkUnitRef

Status: DRAFT

⌈

Parameter Name	EthTSynClkUnitRef
Parent Container	EthTSynPTPPortDetails
Upstream mapping defined	–
Description	Reference to a HW clock unit used for time stamping PTP event messages. **Tags:** atp.Status=draft
Multiplicity	0..1
Type	Symbolic name reference to EthIfClkUnit
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00134] Definition of EcucReferenceDef EthTSynPdelayConfig Ref ⌈

Parameter Name	EthTSynPdelayConfigRef
Parent Container	EthTSynPTPPortDetails
Upstream mapping defined	–
Description	This represents the reference to the configuration for the pDelay mechanism used.
Multiplicity	0..1
Type	Reference to EthTSynPdelayConfig
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00135] Definition of EcucReferenceDef EthTSynTimeMaster ConfigRef ⌈

Parameter Name	EthTSynTimeMasterConfigRef
Parent Container	EthTSynPTPPortDetails
Upstream mapping defined	–
Description	This represents the reference to the configuration used by the port that ends up running in TIME_MASTER role.
Multiplicity	0..1
Type	Reference to EthTSynGlobalTimeMasterConfig
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00136] Definition of EcucReferenceDef EthTSynTimeSlaveConfigRef ⌈

Parameter Name	EthTSynTimeSlaveConfigRef
Parent Container	EthTSynPTPPortDetails
Upstream mapping defined	–
Description	This represents the reference to the configuration used by the port that ends up running in TIME_SLAVE role.
Multiplicity	0..1
Type	Reference to EthTSynGlobalTimeSlaveConfig
Pre-Compile only	true

⌋

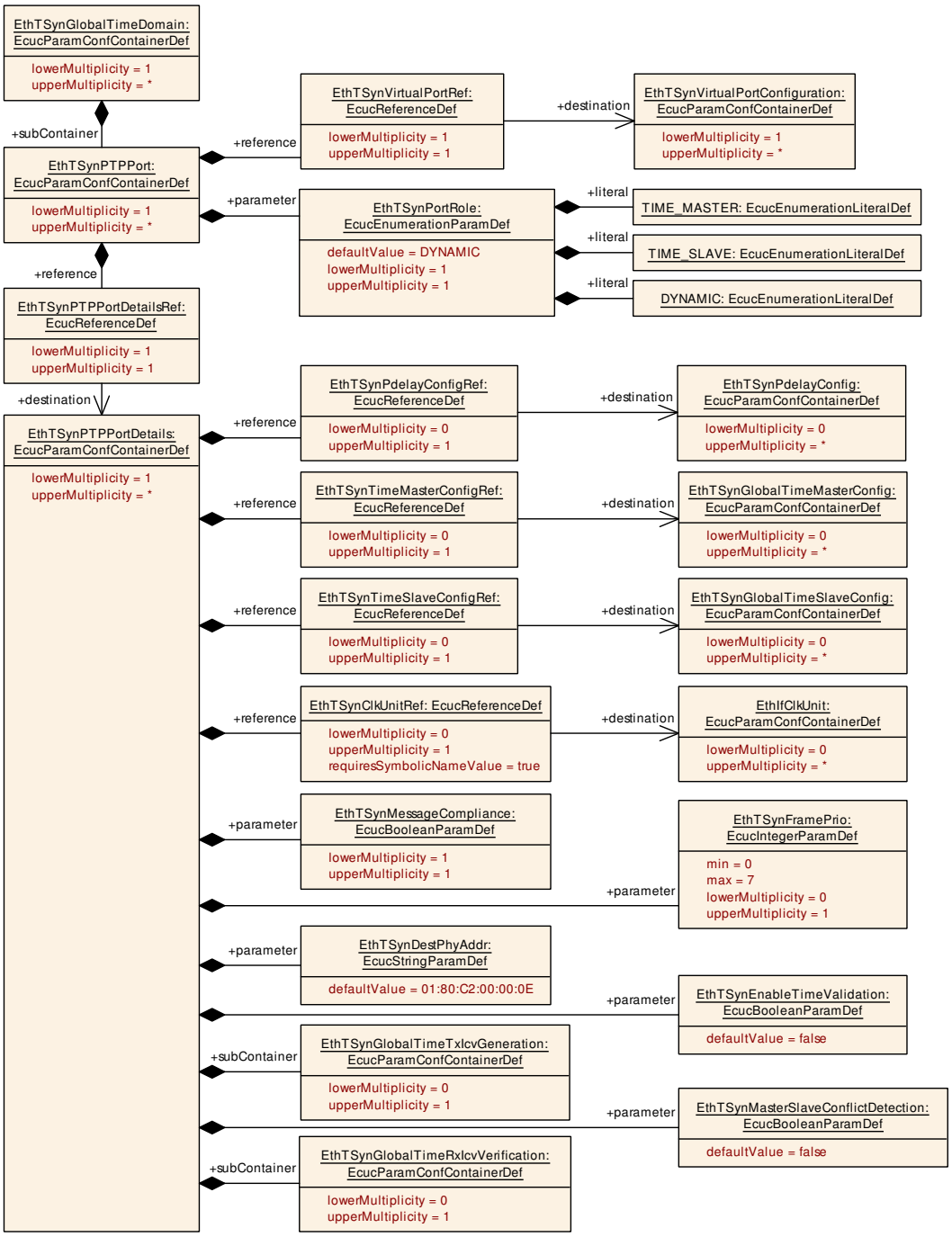


Figure 10.6: EthTSynPTPPortDetails

10.2.9 EthTSynCMLDS

[ECUC_EthTSyn_00141] Definition of EcucParamConfContainerDef EthTSyn CMLDS ⌈

Container Name	EthTSynCMLDS	
Parent Container	EthTSyn	
Upstream mapping defined	–	
Description	Configuration of the Common Mean Link Delay Service (CMLDS). There is at most one instance of the CMLDS per physical port. For each physical port the CMLDS uses a single virtual port.	
Multiplicity	0..*	
Post-Build Multiplicity	-	Selectable
Configuration Parameters		

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
EthTSynCMLDSPdelayConfigRef	1	[ECUC_EthTSyn_00142]
EthTSynCMLDSVirtualPortRef	1	[ECUC_EthTSyn_00143]

No Included Containers

⌋

[ECUC_EthTSyn_00142] Definition of EcucReferenceDef EthTSynCMLDSPdelay ConfigRef ⌈

Parameter Name	EthTSynCMLDSPdelayConfigRef
Parent Container	EthTSynCMLDS
Upstream mapping defined	–
Description	Reference to the pDelay configuration used for the CMLDS. Using a reference here allows to not have to change the pDelay configuration, when changing from transport specific to CMLDS.
Multiplicity	1
Type	Reference to EthTSynPdelayConfig
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00143] Definition of EcucReferenceDef EthTSynCMLDSVirtual PortRef ⌈

Parameter Name	EthTSynCMLDSVirtualPortRef
Parent Container	EthTSynCMLDS
Upstream mapping defined	–
Description	This represents the reference to the virtual port (VLAN) used to transmit and receive p Delay messages. For each physical port the CMLDS uses a single virtual port.
Multiplicity	1
Type	Reference to EthTSynVirtualPortConfiguration
Pre-Compile only	true

⌋

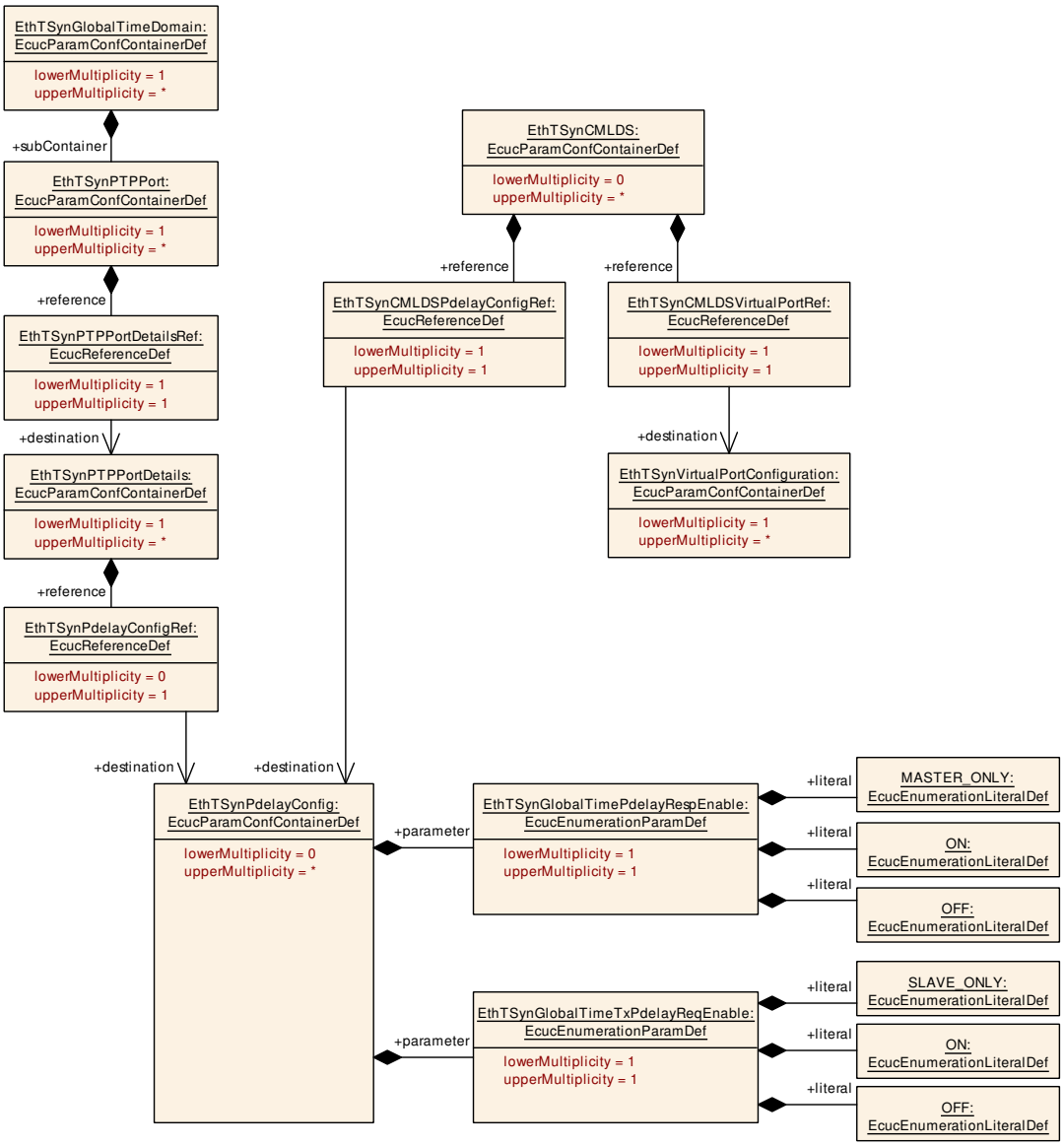


Figure 10.7: EthTSynCMLDS

10.2.10 EthTSynPhysicalPortConfiguration

[ECUC_EthTSyn_00139] Definition of EcucParamConfContainerDef EthTSynPhysicalPortConfiguration ⌈

Container Name	EthTSynPhysicalPortConfiguration
Parent Container	EthTSyn
Upstream mapping defined	[13, CP_TPS_SystemTemplate]

▽

△

Description	Configuration of a physical port. The same configuration can be used by multiple physical ports through referencing this configuration.	
Multiplicity	1..*	
Post-Build Multiplicity	-	Selectable
Configuration Parameters		

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTSynGlobalTimeRxToUplinkSwitchResidenceTime	0..1	[ECUC_EthTSyn_00060]
EthTSynGlobalTimeUplinkToTxSwitchResidenceTime	0..1	[ECUC_EthTSyn_00061]
EthTSynHardwareTimestampSupport	1	[ECUC_EthTSyn_00018]
EthTSynPhysicalPortType	1	[ECUC_EthTSyn_00140]
EthTSynSwitchManagementEthSwitchPortRef	0..1	[ECUC_EthTSyn_00066]

No Included Containers

⌋

[ECUC_EthTSyn_00060] Definition of EcucFloatParamDef EthTSynGlobalTimeRxToUplinkSwitchResidenceTime ⌈

Parameter Name	EthTSynGlobalTimeRxToUplinkSwitchResidenceTime	
Parent Container	EthTSynPhysicalPortConfiguration	
Upstream mapping defined	–	
Description	This parameter is specifying the default value used for the residence time of the Ethernet Switch [Ingress to Uplink]. This value is used by the EthTSyn if the calculation of the residence time failed. Unit: seconds	
Multiplicity	0..1	
Type	EcucFloatParamDef	
Range	[0 .. 4[	
Default value	0	
Pre-Compile only	true	

⌋

[ECUC_EthTSyn_00061] Definition of EcucFloatParamDef EthTSynGlobalTimeUplinkToTxSwitchResidenceTime ⌈

Parameter Name	EthTSynGlobalTimeUplinkToTxSwitchResidenceTime
Parent Container	EthTSynPhysicalPortConfiguration
Upstream mapping defined	–
Description	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 EthTSyn if the calculation of the residence time failed. Unit: seconds
Multiplicity	0..1

▽

△

Type	EcucFloatParamDef	
Range	[0 .. 4[	
Default value	0	
Pre-Compile only	true	

⌋

[ECUC_EthTSyn_00018] Definition of EcucBooleanParamDef EthTSynHardwareTimestampSupport ⌈

Parameter Name	EthTSynHardwareTimestampSupport
Parent Container	EthTSynPhysicalPortConfiguration
Upstream mapping defined	–
Description	Activate/Deactivate the hardware time stamping functionality of the Ethernet hardware. True: Timestamp is retrieved from the Ethernet hardware False: Timestamp is retrieved from the StbM
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00140] Definition of EcucEnumerationParamDef EthTSynPhysicalPortType ⌈

Parameter Name	EthTSynPhysicalPortType	
Parent Container	EthTSynPhysicalPortConfiguration	
Upstream mapping defined	–	
Description	Definition of the physical port type.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	FullDuplex	Full duplex
	HalfDuplex	Half duplex
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00066] Definition of EcucReferenceDef EthTSynSwitchManagementEthSwitchPortRef ⌈

Parameter Name	EthTSynSwitchManagementEthSwitchPortRef
Parent Container	EthTSynPhysicalPortConfiguration
Upstream mapping defined	[13, CP_TPS_SystemTemplate]
Description	In an AVB-Bridge config, this reference is used to assign the EthTSynPTPPort to an Ethernet Switch-Port.
Multiplicity	0..1
Type	Symbolic name reference to EthSwtPort
Pre-Compile only	true

⌋

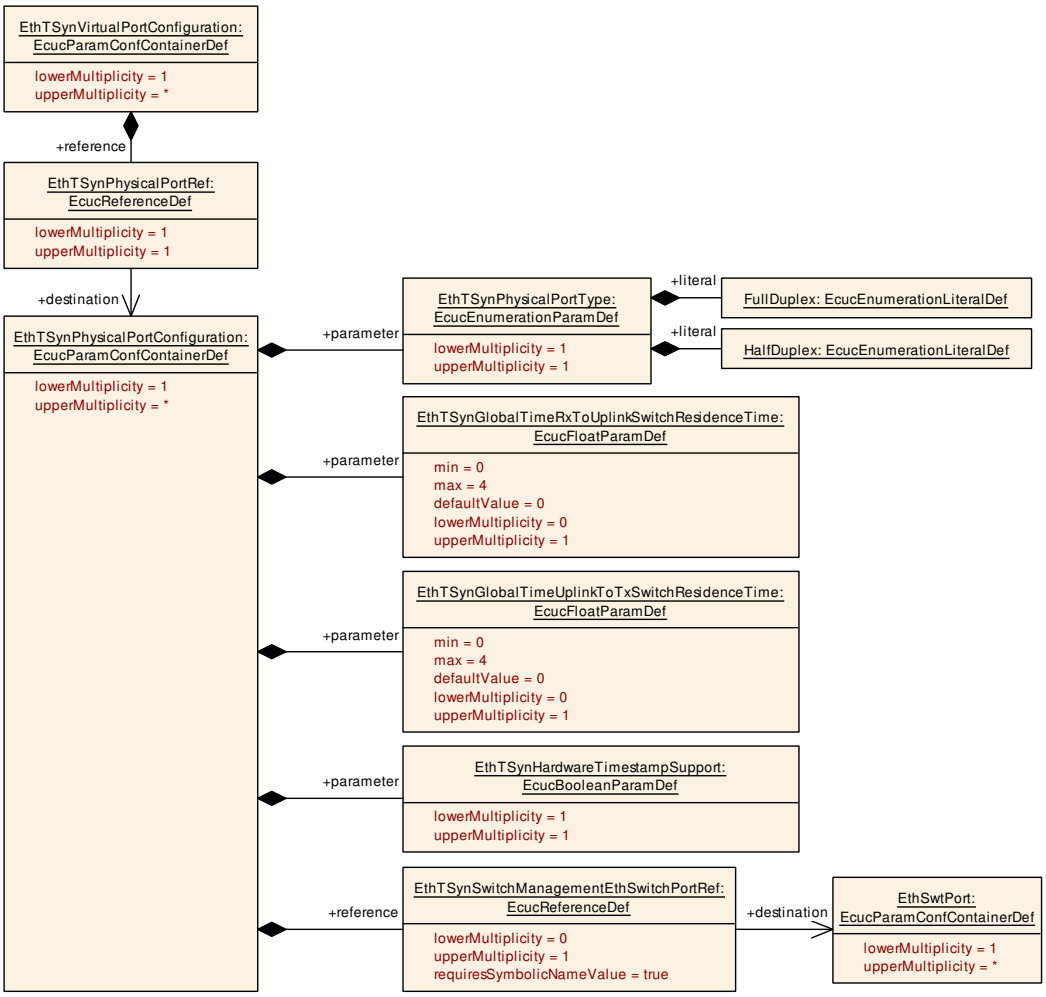


Figure 10.8: EthTSynPhysicalPortConfiguration

10.2.11 EthTSynVirtualPortConfiguration

[ECUC_EthTSyn_00137] Definition of EcucParamConfContainerDef EthTSynVirtualPortConfiguration ⌈

Container Name	EthTSynVirtualPortConfiguration	
Parent Container	EthTSyn	
Upstream mapping defined	–	
Description	Configuration of a virtual port (VLAN) associated with a single physical port and used by different PTP and/or CMLDS ports to transmit and receive PTP messages.	
Multiplicity	1..*	
Post-Build Multiplicity	-	Selectable
Configuration Parameters		

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTSynGlobalTimeEthIfRef	1	[ECUC_EthTSyn_00065]
EthTSynPhysicalPortRef	1	[ECUC_EthTSyn_00138]

Included Containers		
Container Name	**Multiplicity**	**Description**
EthTSynRxPdu	0..1	PDU used for reception. Supported MetaData entry: TIMETUPLE_TYPE_PTR **Tags:** atp.Status=draft
EthTSynTxPdu	0..1	PDU used for transmission. **Tags:** atp.Status=draft

⌋

[ECUC_EthTSyn_00065] Definition of EcucReferenceDef EthTSynGlobalTimeEthIfRef ⌈

Parameter Name	EthTSynGlobalTimeEthIfRef
Parent Container	EthTSynVirtualPortConfiguration
Upstream mapping defined	–
Description	This represents the reference to the Ethernet interface used to exchange global time information.
Multiplicity	1
Type	Symbolic name reference to EthIfController
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00138] Definition of EcucReferenceDef EthTSynPhysicalPortRef ⌈

Parameter Name	EthTSynPhysicalPortRef
Parent Container	EthTSynVirtualPortConfiguration
Upstream mapping defined	–
Description	Reference to the physical port configuration.
Multiplicity	1
Type	Reference to EthTSynPhysicalPortConfiguration
Pre-Compile only	true

⌋

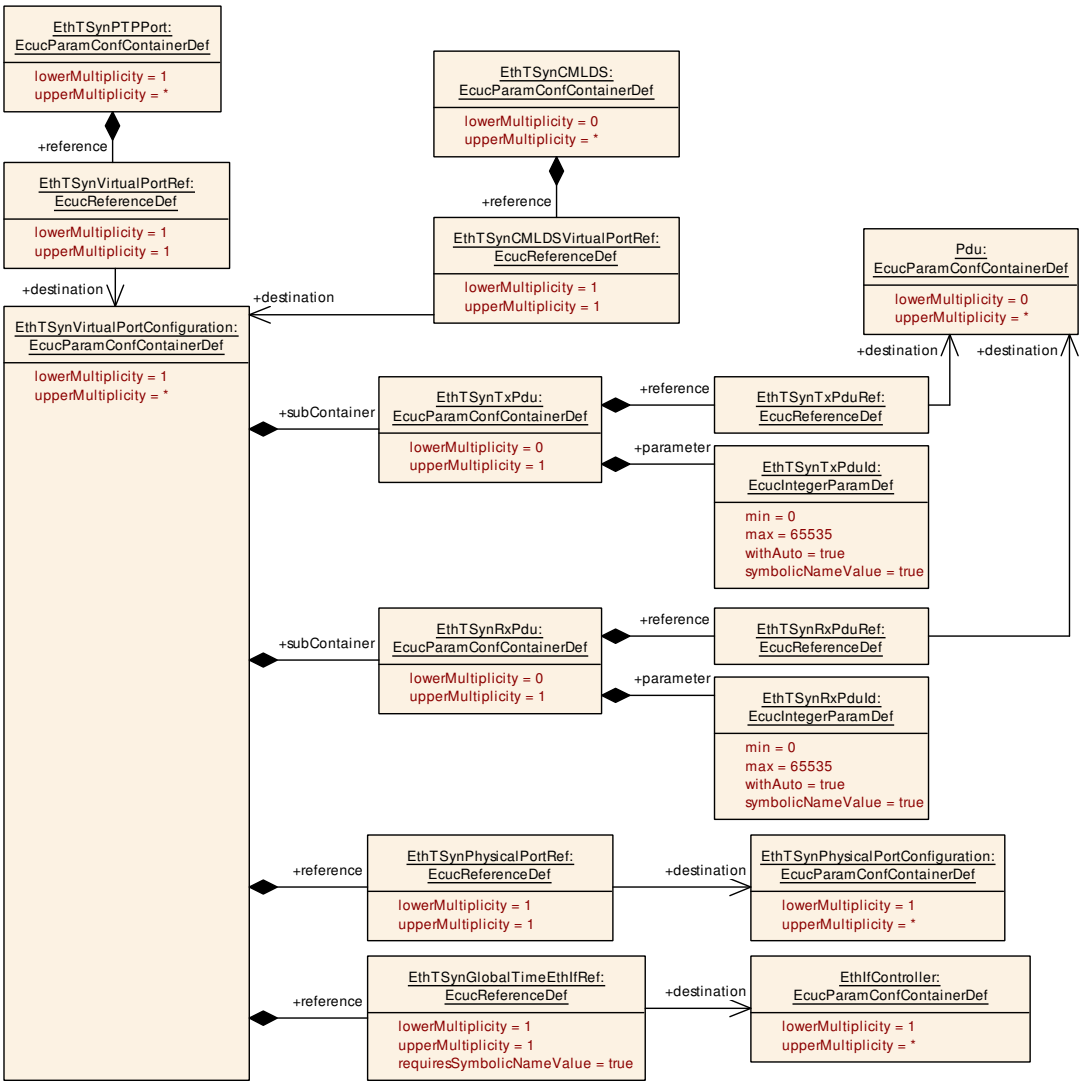


Figure 10.9: EthTSynVirtualPortConfiguration

10.2.12 EthTSynGlobalTimeTxIcvGeneration

[ECUC_EthTSyn_00096] Definition of EcucParamConfContainerDef EthTSynGlobalTimeTxIcvGeneration ⌈

Container Name	EthTSynGlobalTimeTxIcvGeneration
Parent Container	EthTSynPTPPortDetails
Upstream mapping defined	[13, CP_TPS_SystemTemplate]
Description	This container collects configuration that shall be used for ICV generation.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTSynIcvGenerationBase	1	[ECUC_EthTSyn_00098]
EthTSynIcvGenerationTimeout	1	[ECUC_EthTSyn_00101]
EthTSynIcvTxLength	1	[ECUC_EthTSyn_00099]
EthTSynTxAuthenticationBuildAttempts	1	[ECUC_EthTSyn_00113]
EthTSynIcvGenerationFvIdRef	0..1	[ECUC_EthTSyn_00097]
EthTSynIcvGenerationJobRef	1	[ECUC_EthTSyn_00100]

No Included Containers

⌋

[ECUC_EthTSyn_00098] Definition of EcucEnumerationParamDef EthTSynIcvGenerationBase ⌈

Parameter Name	EthTSynIcvGenerationBase	
Parent Container	EthTSynGlobalTimeTxIcvGeneration	
Upstream mapping defined	–	
Description	Symmetric or asymmetric cryptography selection for the ICV generation	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	ICV_MAC	Symmetric cryptography selection for the ICV generation.
	ICV_SIGNATURE	Asymmetric cryptography selection for the ICV generation.
Pre-Compile only	true	

⌋

[ECUC_EthTSyn_00101] Definition of EcucFloatParamDef EthTSynIcvGenerationTimeout ⌈

Parameter Name	EthTSynIcvGenerationTimeout	
Parent Container	EthTSynGlobalTimeTxIcvGeneration	
Upstream mapping defined	–	
Description	Timeout of ICV generation (respective CSM job completion in asynchronous behaviour). A value of 0 disables the ICV timeout monitoring. Unit: Seconds	
Multiplicity	1	
Type	EcucFloatParamDef	
Range	[0 .. INF[	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_EthTSyn_00099] Definition of EcucIntegerParamDef EthTSynIcvTx Length ⌈

Parameter Name	EthTSynIcvTxLength	
Parent Container	EthTSynGlobalTimeTxIcvGeneration	
Upstream mapping defined	–	
Description	Length of ICV to be transmitted within Follow_Up Message on the bus (in bytes).	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 1061	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_EthTSyn_00113] Definition of EcucIntegerParamDef EthTSynTxAuthenticationBuildAttempts ⌈

Parameter Name	EthTSynTxAuthenticationBuildAttempts	
Parent Container	EthTSynGlobalTimeTxIcvGeneration	
Upstream mapping defined	–	
Description	This parameter specifies the number of authentication build attempts that are to be carried out when the generation of the ICV failed for a given Follow_Up message. If zero is set, then only one ICV generation attempt is done.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 65535	
Default value	0	
Pre-Compile only	true	

⌋

[ECUC_EthTSyn_00097] Definition of EcucReferenceDef EthTSynIcvGeneration FvIdRef ⌈

Parameter Name	EthTSynIcvGenerationFvIdRef
Parent Container	EthTSynGlobalTimeTxIcvGeneration
Upstream mapping defined	[13, CP_TPS_SystemTemplate]
Description	This represents the reference to the FV taken to generate the ICV generation.
Multiplicity	0..1
Type	Symbolic name reference to StbMFreshnessValue
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00100] Definition of EcucReferenceDef EthTSynIcvGeneration JobRef ⌈

Parameter Name	EthTSynIcvGenerationJobRef
Parent Container	EthTSynGlobalTimeTxIcvGeneration
Upstream mapping defined	[13, CP_TPS_SystemTemplate]
Description	This represents the reference to the CSM job to fetch the CSM job ID.
Multiplicity	1
Type	Symbolic name reference to CsmJob
Pre-Compile only	true

⌋

10.2.13 EthTSynGlobalTimeRxIcvVerification

[ECUC_EthTSyn_00104] Definition of EcucParamConfContainerDef EthTSyn GlobalTimeRxIcvVerification ⌈

Container Name	EthTSynGlobalTimeRxIcvVerification
Parent Container	EthTSynPTPPortDetails
Upstream mapping defined	[13, CP_TPS_SystemTemplate]
Description	This container collects configuration that shall be used for ICV verification.
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTSynIcvRxLength	1	[ECUC_EthTSyn_00107]
EthTSynIcvVerificationAttempts	1	[ECUC_EthTSyn_00110]
EthTSynIcvVerificationBase	1	[ECUC_EthTSyn_00106]
EthTSynIcvVerificationTimeout	1	[ECUC_EthTSyn_00109]
EthTSynRxAuthenticationBuildAttempts	1	[ECUC_EthTSyn_00112]
EthTSynIcvVerificationFvIdRef	0..1	[ECUC_EthTSyn_00105]
EthTSynIcvVerificationJobRef	1	[ECUC_EthTSyn_00108]

No Included Containers

⌋

[ECUC_EthTSyn_00107] Definition of EcucIntegerParamDef EthTSynIcvRx Length ⌈

Parameter Name	EthTSynIcvRxLength
Parent Container	EthTSynGlobalTimeRxIcvVerification
Upstream mapping defined	–

▽

△

Description	Length of ICV to be transmitted within Follow_Up Message on the bus (in bytes).	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 1061	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_EthTSyn_00110] Definition of EcucIntegerParamDef EthTSynIcvVerificationAttempts ⌈

Parameter Name	EthTSynIcvVerificationAttempts	
Parent Container	EthTSynGlobalTimeRxIcvVerification	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	This parameter specifies the number of ICV verification attempts that are to be carried out when the verification of the ICV failed for a given secured Follow_Up message. If zero is set, then only one ICV verification attempt is done.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 65535	
Default value	0	
Pre-Compile only	true	

⌋

[ECUC_EthTSyn_00106] Definition of EcucEnumerationParamDef EthTSynIcvVerificationBase ⌈

Parameter Name	EthTSynIcvVerificationBase	
Parent Container	EthTSynGlobalTimeRxIcvVerification	
Upstream mapping defined	–	
Description	Symmetric or asymmetric cryptography selection for the ICV generation	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	ICV_MAC	Symmetric cryptography selection for the ICV generation.
	ICV_SIGNATURE	Asymmetric cryptography selection for the ICV generation.
Pre-Compile only	true	

⌋

[ECUC_EthTSyn_00109] Definition of EcucFloatParamDef EthTSynIcvVerificationTimeout ⌈

Parameter Name	EthTSynIcvVerificationTimeout
Parent Container	EthTSynGlobalTimeRxIcvVerification
Upstream mapping defined	–

▽

△

Description	Timeout of ICV verification (respective CSM job completion in asynchronous behavior). A value of 0 disables the ICV timeout monitoring. Unit: Seconds	
Multiplicity	1	
Type	EcucFloatParamDef	
Range	[0 .. INF[	
Default value	–	
Pre-Compile only	true	

⌋

[ECUC_EthTSyn_00112] Definition of EcucIntegerParamDef EthTSynRxAuthenticationBuildAttempts ⌈

Parameter Name	EthTSynRxAuthenticationBuildAttempts	
Parent Container	EthTSynGlobalTimeRxIcvVerification	
Upstream mapping defined	–	
Description	This parameter specifies the number of authentication build attempts that are to be carried out when the verification of the ICV failed for a given Follow_Up message. If zero is set, then only one ICV verification attempt is done.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 65535	
Default value	0	
Pre-Compile only	true	

⌋

[ECUC_EthTSyn_00105] Definition of EcucReferenceDef EthTSynIcvVerificationFvIdRef ⌈

Parameter Name	EthTSynIcvVerificationFvIdRef
Parent Container	EthTSynGlobalTimeRxIcvVerification
Upstream mapping defined	[13, CP_TPS_SystemTemplate]
Description	This represents the reference to the FV taken to verify the ICV.
Multiplicity	0..1
Type	Symbolic name reference to StbMFreshnessValue
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00108] Definition of EcucReferenceDef EthTSynIcvVerificationJobRef ⌈

Parameter Name	EthTSynIcvVerificationJobRef
Parent Container	EthTSynGlobalTimeRxIcvVerification
Upstream mapping defined	[13, CP_TPS_SystemTemplate]
Description	This represents the reference to the CSM job to fetch the CSM job ID.
Multiplicity	1

▽

△

Type	Symbolic name reference to CsmJob
Pre-Compile only	true

⌋

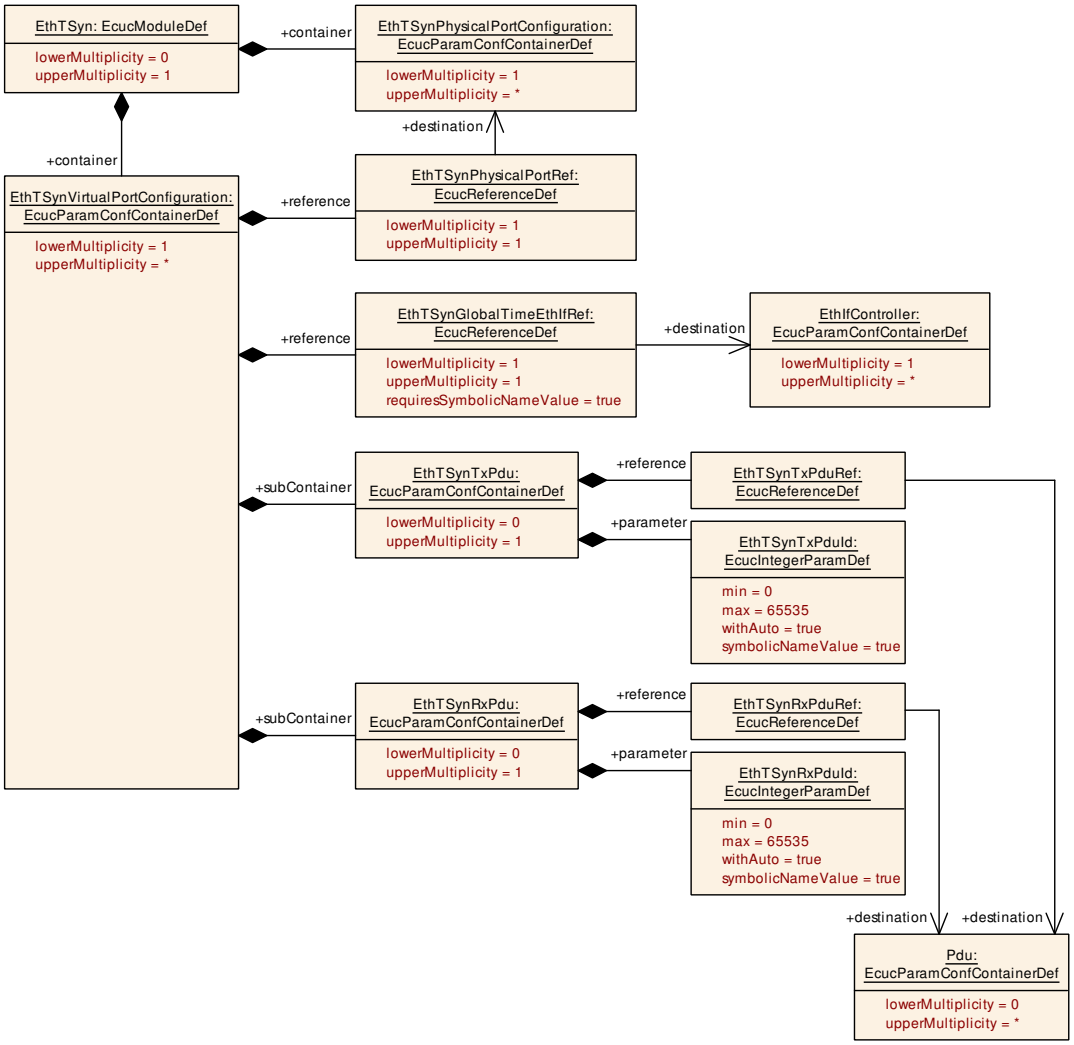


Figure 10.10: EthTSyn_GlobalTimePdu

10.2.14 EthTSynTxPdu

[ECUC_EthTSyn_00122] Definition of EcucParamConfContainerDef EthTSynTx Pdu

Status: DRAFT

⌈

Container Name	EthTSynTxPdu
Parent Container	EthTSynVirtualPortConfiguration
Upstream mapping defined	–
Description	PDU used for transmission. **Tags:** atp.Status=draft
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTSynTxPduId	1	[ECUC_EthTSyn_00121]
EthTSynTxPduRef	1	[ECUC_EthTSyn_00123]

No Included Containers

⌋

[ECUC_EthTSyn_00121] Definition of EcucIntegerParamDef EthTSynTxPduId

Status: DRAFT

⌈

Parameter Name	EthTSynTxPduId	
Parent Container	EthTSynTxPdu	
Upstream mapping defined	–	
Description	PDU identifier used for TxConfirmation from LSduR. **Tags:** atp.Status=draft	
Multiplicity	1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 65535	
Default value	–	
With Auto	true	
Pre-Compile only	true	

⌋

[ECUC_EthTSyn_00123] Definition of EcucReferenceDef EthTSynTxPduRef

Status: DRAFT

⌈

Parameter Name	EthTSynTxPduRef
Parent Container	EthTSynTxPdu
Upstream mapping defined	–
Description	Reference to the Pdu object representing the PDU. **Tags:** atp.Status=draft
Multiplicity	1

▽

△

Type	Reference to Pdu
Pre-Compile only	true

⌋

10.2.15 EthTSynRxPdu

[ECUC_EthTSyn_00124] Definition of EcucParamConfContainerDef EthTSynRxPdu

Status: DRAFT

⌈

Container Name	EthTSynRxPdu
Parent Container	EthTSynVirtualPortConfiguration
Upstream mapping defined	–
Description	PDU used for reception. Supported MetaData entry: TIMETUPLE_TYPE_PTR **Tags:** atp.Status=draft
Multiplicity	0..1
Pre-Compile only	true
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTSynRxPduId	1	[ECUC_EthTSyn_00125]
EthTSynRxPduRef	1	[ECUC_EthTSyn_00126]

No Included Containers

⌋

[ECUC_EthTSyn_00125] Definition of EcucIntegerParamDef EthTSynRxPduId

Status: DRAFT

⌈

Parameter Name	EthTSynRxPduId	
Parent Container	EthTSynRxPdu	
Upstream mapping defined	–	
Description	PDU identifier used for RxIndication from LSduR. **Tags:** atp.Status=draft	
Multiplicity	1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 65535	

▽

△

Default value	–
With Auto	true
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00126] Definition of EcucReferenceDef EthTSynRxPduRef

Status: DRAFT

⌈

Parameter Name	EthTSynRxPduRef
Parent Container	EthTSynRxPdu
Upstream mapping defined	–
Description	Reference to the Pdu object representing the PDU. **Tags:** atp.Status=draft
Multiplicity	1
Type	Reference to Pdu
Pre-Compile only	true

⌋

10.2.16 EthTSynPdelayConfig

[ECUC_EthTSyn_00068] Definition of EcucParamConfContainerDef EthTSynPdelayConfig ⌈

Container Name	EthTSynPdelayConfig	
Parent Container	EthTSyn	
Upstream mapping defined	–	
Description	Configuration of cyclic propagation delay measurement. This can be used by either a PTP port or the CMLDS.	
Multiplicity	0..*	
Post-Build Multiplicity	-	Selectable
Configuration Parameters		

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTSynGlobalTimePdelayRespEnable	1	[ECUC_EthTSyn_00129]
EthTSynGlobalTimePropagationDelay	1	[ECUC_EthTSyn_00070]
EthTSynGlobalTimeTxPdelayReqEnable	1	[ECUC_EthTSyn_00144]
EthTSynGlobalTimeTxPdelayReqPeriod	1	[ECUC_EthTSyn_00071]
EthTSynPdelayLatencyThreshold	0..1	[ECUC_EthTSyn_00076]
EthTSynPdelayRespAndRespFollowUpTimeout	1	[ECUC_EthTSyn_00074]
EthTSynRateRatioMeasurementCount	1	[ECUC_EthTSyn_00117]

No Included Containers

⌋

[ECUC_EthTSyn_00129] Definition of EcucEnumerationParamDef EthTSyn GlobalTimePdelayRespEnable ⌈

Parameter Name	EthTSynGlobalTimePdelayRespEnable	
Parent Container	EthTSynPdelayConfig	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	Determines if a Pdelay Response message is transmitted upon reception of a Pdelay Request message.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	MASTER_ONLY	only if Port is in TimeMaster role
	OFF	never
	ON	always
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00070] Definition of EcucFloatParamDef EthTSynGlobalTime PropagationDelay ⌈

Parameter Name	EthTSynGlobalTimePropagationDelay	
Parent Container	EthTSynPdelayConfig	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	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. Unit: seconds	
Multiplicity	1	
Type	EcucFloatParamDef	
Range	[0 .. 4]	
Default value	0	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00144] Definition of EcucEnumerationParamDef EthTSyn GlobalTimeTxPdelayReqEnable ⌈

Parameter Name	EthTSynGlobalTimeTxPdelayReqEnable	
Parent Container	EthTSynPdelayConfig	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	Determines if Pdelay Request messages are transmitted.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	OFF	never
	ON	always

▽

△

	SLAVE_ONLY	only if Port is in TimeSlave role
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00071] Definition of EcucFloatParamDef EthTSynGlobalTimeTxPdelayReqPeriod ⌈

Parameter Name	EthTSynGlobalTimeTxPdelayReqPeriod	
Parent Container	EthTSynPdelayConfig	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	This represents configuration of the TX period for Pdelay_Req messages. A value of 0 disables the cyclic Pdelay measurement. Unit: seconds	
Multiplicity	1	
Type	EcucFloatParamDef	
Range	[0 .. 4]	
Default value	0	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00076] Definition of EcucFloatParamDef EthTSynPdelayLatencyThreshold ⌈

Parameter Name	EthTSynPdelayLatencyThreshold	
Parent Container	EthTSynPdelayConfig	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	Threshold for calculated Pdelay. If a measured Pdelay exceeds EthTSynPdelayLatency Threshold, this value is discarded. Unit: seconds	
Multiplicity	0..1	
Type	EcucFloatParamDef	
Range	]0 .. INF[	
Default value	1E-5	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00074] Definition of EcucFloatParamDef EthTSynPdelayRespAndRespFollowUpTimeout ⌈

Parameter Name	EthTSynPdelayRespAndRespFollowUpTimeout
Parent Container	EthTSynPdelayConfig
Upstream mapping defined	[13, CP_TPS_SystemTemplate]

▽

△

Description	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. Unit: seconds	
Multiplicity	1	
Type	EcucFloatParamDef	
Range	[0 .. 4]	
Default value	–	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00117] Definition of EcucIntegerParamDef EthTSynRateRatioMeasurementCount

Status: DRAFT

⌈

Parameter Name	EthTSynRateRatioMeasurementCount	
Parent Container	EthTSynPdelayConfig	
Upstream mapping defined	–	
Description	This parameter defines the number of successful pDelay measurements used to calculate the neighbor rate ratio according to IEEE 802.1AS. **Tags:** atp.Status=draft	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	1 .. 255	
Default value	–	
Post-Build Value	-	Selectable

⌋

10.2.17 EthTSynGlobalTimeMasterConfig

[ECUC_EthTSyn_00008] Definition of EcucParamConfContainerDef EthTSynGlobalTimeMasterConfig ⌈

Container Name	EthTSynGlobalTimeMasterConfig
Parent Container	EthTSyn
Upstream mapping defined	[13, CP_TPS_SystemTemplate]
Description	Configuration of a global-time master. The exact role of the port is defined by EthTSynPortRole. A global-time master may or may not exist on the configured ECU. Each global-time domain may have multiple ports in the time master role (see Time Gateway) to e.g. relay global-time from a global time slave port.
Multiplicity	0..*

▽

△

Post-Build Multiplicity	-	Selectable
Configuration Parameters		

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
EthTSynCyclicMsgResumeTime	1	[ECUC_EthTSyn_00047]
EthTSynGlobalTimeTxCrcSecured	1	[ECUC_EthTSyn_00039]
EthTSynGlobalTimeTxPeriod	0..1	[ECUC_EthTSyn_00010]
EthTSynHoldOverTime	0..1	[ECUC_EthTSyn_00115]
EthTSynImmediateTimeSync	1	[ECUC_EthTSyn_00046]
EthTSynTLVFollowUpICVSubTLV	1	[ECUC_EthTSyn_00095]
EthTSynTxSubTLVStatus	1	[ECUC_EthTSyn_00036]
EthTSynTxSubTLVTime	1	[ECUC_EthTSyn_00035]
EthTSynTxSubTLVUserData	1	[ECUC_EthTSyn_00037]

Included Containers		
Container Name	**Multiplicity**	**Description**
EthTSynCrcTimeFlagsTxSecured	0..1	This container collects definitions which parts of the Follow_Up message elements shall be used for CRC calculation.

⌋

[ECUC_EthTSyn_00047] Definition of EcucFloatParamDef EthTSynCyclicMsgResumeTime ⌈

Parameter Name	EthTSynCyclicMsgResumeTime	
Parent Container	EthTSynGlobalTimeMasterConfig	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	Defines the time where the 1st regular cycle time based message transmission takes place, after an immediate transmission before. Unit: seconds	
Multiplicity	1	
Type	EcucFloatParamDef	
Range	[0 .. INF[	
Default value	–	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00039] Definition of EcucEnumerationParamDef EthTSynGlobalTimeTxCrcSecured ⌈

Parameter Name	EthTSynGlobalTimeTxCrcSecured
Parent Container	EthTSynGlobalTimeMasterConfig
Upstream mapping defined	[13, CP_TPS_SystemTemplate]
Description	This represents the configuration of whether or not CRC is supported.
Multiplicity	1
Type	EcucEnumerationParamDef

▽

△

Range	CRC_NOT_SUPPORTED	This represents a configuration where CRC is not supported.
	CRC_SUPPORTED	This represents a configuration where CRC is supported.
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00010] Definition of EcucFloatParamDef EthTSynGlobalTimeTx Period ⌈

Parameter Name	EthTSynGlobalTimeTxPeriod	
Parent Container	EthTSynGlobalTimeMasterConfig	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	This represents configuration of the TX period. Unit: seconds	
Multiplicity	0..1	
Type	EcucFloatParamDef	
Range	[0 .. INF[	
Default value	–	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00115] Definition of EcucFloatParamDef EthTSynHoldOver Time ⌈

Parameter Name	EthTSynHoldOverTime	
Parent Container	EthTSynGlobalTimeMasterConfig	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	Parameter to define timeout for transmission of Sync and Follow_Up messages on Master ports in absence of reception of Sync and Follow_Up messages on Slave port.Unit: seconds	
Multiplicity	0..1	
Type	EcucFloatParamDef	
Range	[0 .. INF[	
Default value	3	
Post-Build Multiplicity	-	Selectable
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00046] Definition of EcucBooleanParamDef EthTSynImmediate TimeSync ⌈

Parameter Name	EthTSynImmediateTimeSync
Parent Container	EthTSynGlobalTimeMasterConfig
Upstream mapping defined	–
Description	Enables/Disables the cyclic polling of StbM_GetTimeBaseUpdateCounter() within EthTSyn_MainFunction().

▽

△

Multiplicity	1	
Type	EcucBooleanParamDef	
Default value	–	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00095] Definition of EcucBooleanParamDef EthTSynTLVFollowUpICVSubTLV ⌈

Parameter Name	EthTSynTLVFollowUpICVSubTLV
Parent Container	EthTSynGlobalTimeMasterConfig
Upstream mapping defined	–
Description	This represents the configuration of whether an AUTOSAR Follow_Up ICV Sub-TLV is used or not. - true: This represents a configuration where an AUTOSAR Follow_Up ICV Sub-TLV is used. - false: This represents a configuration where an AUTOSAR Follow_Up ICV Sub-TLV is not used.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	–
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00036] Definition of EcucBooleanParamDef EthTSynTxSubTLVStatus ⌈

Parameter Name	EthTSynTxSubTLVStatus	
Parent Container	EthTSynGlobalTimeMasterConfig	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	Definition of whether (true) or not (false) a Sub-TLV:Status Secured or Sub-TLV:Status Not Secured shall be sent in the AUTOSAR TLV of a Follow_Up message.	
Multiplicity	1	
Type	EcucBooleanParamDef	
Default value	–	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00035] Definition of EcucBooleanParamDef EthTSynTxSubTLV-Time ⌈

Parameter Name	EthTSynTxSubTLVTime
Parent Container	EthTSynGlobalTimeMasterConfig
Upstream mapping defined	[13, CP_TPS_SystemTemplate]
Description	Definition of whether (true) or not (false) a Sub-TLV:Time Secured shall be sent in the AUTOSAR TLV of a Follow_Up message.
Multiplicity	1
Type	EcucBooleanParamDef

▽

△

Default value	–	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00037] Definition of EcucBooleanParamDef EthTSynTxSubTLVUserData ⌈

Parameter Name	EthTSynTxSubTLVUserData	
Parent Container	EthTSynGlobalTimeMasterConfig	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	Definition of whether (true) or not (false) a Sub-TLV:UserData Secured or Sub-TLV:User Data Not Secured shall be sent in the AUTOSAR TLV of a Follow_Up message.	
Multiplicity	1	
Type	EcucBooleanParamDef	
Default value	–	
Post-Build Value	-	Selectable

⌋

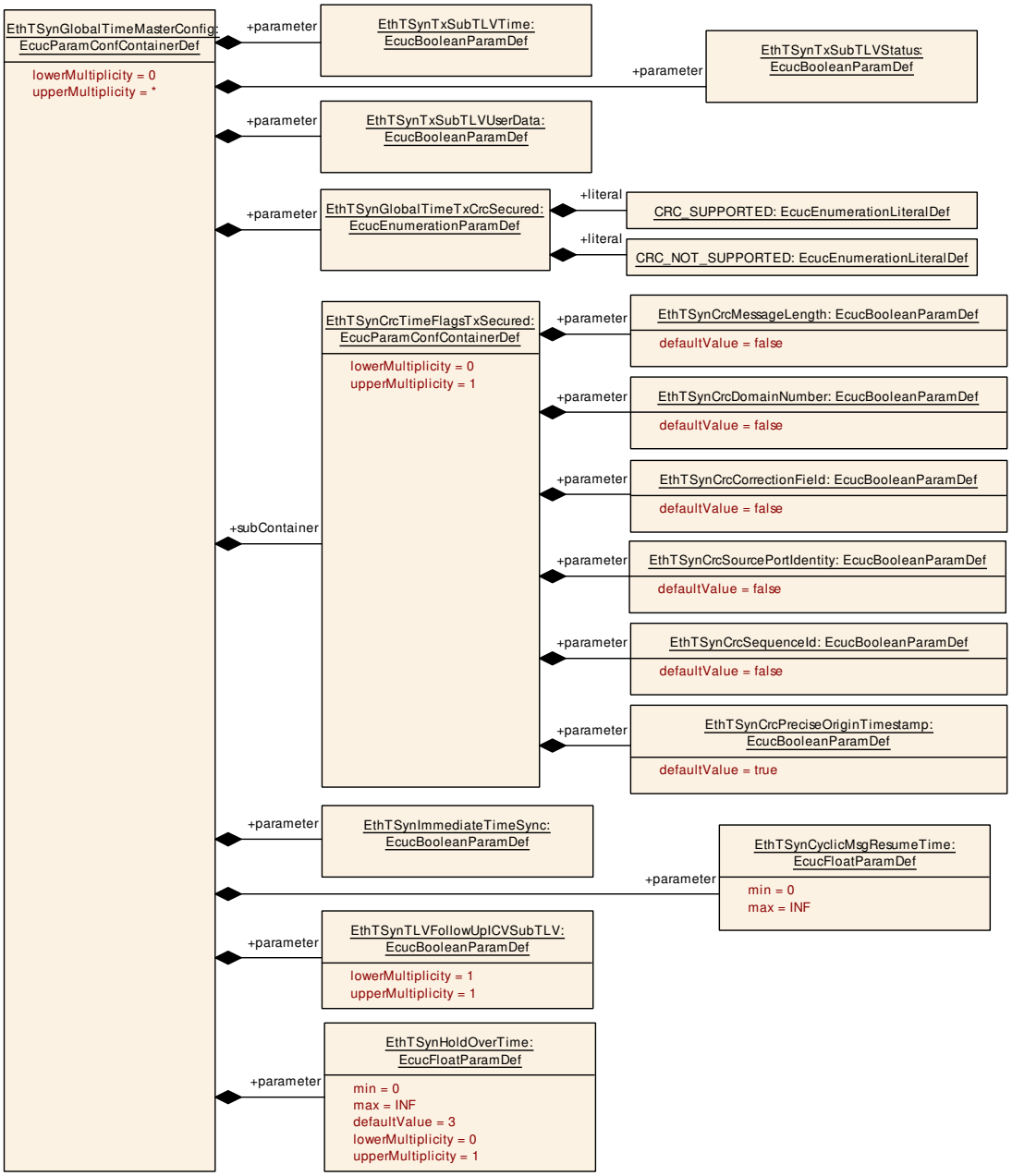


Figure 10.11: EthTSyn_GlobalTimeMaster

10.2.18 EthTSynCrcTimeFlagsTxSecured

[ECUC_EthTSyn_00057] Definition of EcucParamConfContainerDef EthTSynCrc TimeFlagsTxSecured ⌈

Container Name	EthTSynCrcTimeFlagsTxSecured	
Parent Container	EthTSynGlobalTimeMasterConfig	
Upstream mapping defined	–	
Description	This container collects definitions which parts of the Follow_Up message elements shall be used for CRC calculation.	
Multiplicity	0..1	
Post-Build Multiplicity	-	Selectable
Configuration Parameters		

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
EthTSynCrcCorrectionField	1	[ECUC_EthTSyn_00042]
EthTSynCrcDomainNumber	1	[ECUC_EthTSyn_00041]
EthTSynCrcMessageLength	1	[ECUC_EthTSyn_00040]
EthTSynCrcPreciseOriginTimestamp	1	[ECUC_EthTSyn_00045]
EthTSynCrcSequenceId	1	[ECUC_EthTSyn_00044]
EthTSynCrcSourcePortIdentity	1	[ECUC_EthTSyn_00043]

No Included Containers

⌋

For parameter table [ECUC_EthTSyn_00042] EthTSynCrcCorrectionField, see definition below container EthTSynCrcFlagsRxValidated.

For parameter table [ECUC_EthTSyn_00041] EthTSynCrcDomainNumber, see definition below container EthTSynCrcFlagsRxValidated.

For parameter table [ECUC_EthTSyn_00040] EthTSynCrcMessageLength, see definition below container EthTSynCrcFlagsRxValidated.

For parameter table [ECUC_EthTSyn_00045] EthTSynCrcPreciseOriginTimestamp, see definition below container EthTSynCrcFlagsRxValidated.

For parameter table [ECUC_EthTSyn_00044] EthTSynCrcSequenceId, see definition below container EthTSynCrcFlagsRxValidated.

For parameter table [ECUC_EthTSyn_00043] EthTSynCrcSourcePortIdentity, see definition below container EthTSynCrcFlagsRxValidated.

10.2.19 EthTSynGlobalTimeSlaveConfig

[ECUC_EthTSyn_00009] Definition of EcucParamConfContainerDef EthTSynGlobalTimeSlaveConfig ⌈

Container Name	EthTSynGlobalTimeSlaveConfig	
Parent Container	EthTSyn	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	Configuration of a global-time slave. The exact role of the port is defined by EthTSynPortRole. A TIME_SLAVE port will exist on every configured ECU, except potentially on the SystemWideGlobalTimeMaster. Each global-time domain should have at least one port of one ECU in the TIME_MASTER role and one port of another ECU in the TIME_SLAVE role.	
Multiplicity	0..*	
Post-Build Multiplicity	-	Selectable
Configuration Parameters		

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
EthTSynGlobalTimeFollowUpTimeout	1	[ECUC_EthTSyn_00007]
EthTSynGlobalTimeSequenceCounterHysteresis	1	[ECUC_EthTSyn_00084]
EthTSynGlobalTimeSequenceCounterJumpWidth	1	[ECUC_EthTSyn_00083]
EthTSynRxCrcValidated	1	[ECUC_EthTSyn_00049]
EthTSynRxIcvVerificationType	1	[ECUC_EthTSyn_00103]
EthTSynRxSubTLVStatus	1	[ECUC_EthTSyn_00086]
EthTSynRxSubTLVTime	1	[ECUC_EthTSyn_00085]
EthTSynRxSubTLVUserData	1	[ECUC_EthTSyn_00087]

Included Containers		
Container Name	Multiplicity	Description
EthTSynCrcFlagsRxValidated	0..1	This container collects definitions which parts of the Follow_Up message elements shall be included in CRC validation.

⌋

[ECUC_EthTSyn_00007] Definition of EcucFloatParamDef EthTSynGlobalTimeFollowUpTimeout ⌈

Parameter Name	EthTSynGlobalTimeFollowUpTimeout	
Parent Container	EthTSynGlobalTimeSlaveConfig	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	Timeout value of the Follow_Up message (of the subsequent Sync message). A value of 0 deactivates this timeout observation. Unit: seconds	
Multiplicity	1	
Type	EcucFloatParamDef	
Range	[0 .. 4]	
Default value	–	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00084] Definition of EcucIntegerParamDef EthTSynGlobalTimeSequenceCounterHysteresis ⌈

Parameter Name	EthTSynGlobalTimeSequenceCounterHysteresis	
Parent Container	EthTSynGlobalTimeSlaveConfig	
Upstream mapping defined	–	
Description	EthTSynGlobalTimeSequenceCounterHysteresis specifies the number of consecutive valid message pairs that are required by the Time Slave while being in Timeout state until a Time Tuple is forwarded to the StbM.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 15	
Default value	0	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00083] Definition of EcucIntegerParamDef EthTSynGlobalTimeSequenceCounterJumpWidth ⌈

Parameter Name	EthTSynGlobalTimeSequenceCounterJumpWidth	
Parent Container	EthTSynGlobalTimeSlaveConfig	
Upstream mapping defined	–	
Description	The SequenceCounterJumpWidth specifies the maximum allowed jump of the Sequence Counter between two consecutive Sync messages.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 65535	
Default value	0	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00049] Definition of EcucEnumerationParamDef EthTSynRxCrcValidated ⌈

Parameter Name	EthTSynRxCrcValidated	
Parent Container	EthTSynGlobalTimeSlaveConfig	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	Definition of whether or not validation of the CRC takes place.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	CRC_IGNORED	EthTSyn ignores any CRC inside the Sub-TLVs.
	CRC_NOT_VALIDATED	If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x50 or 0x60.
	CRC_OPTIONAL	If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x50 or 0x60, that contain an incorrect CRC value.

▽

△

	CRC_VALIDATED	If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x50 or 0x60, that contain an incorrect CRC value. EthTSyn rejects Follow_Up messages with Sub-TLVs of Type 0x51 or 0x61.
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00103] Definition of EcucEnumerationParamDef EthTSynRxIcvVerificationType ⌈

Parameter Name	EthTSynRxIcvVerificationType	
Parent Container	EthTSynGlobalTimeSlaveConfig	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	This parameter controls whether or not ICV verification shall be supported.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	ICV_IGNORED	The Timesync module shall not verify the ICV.
	ICV_NOT_VERIFIED	The Timesync module accepts only Time Synchronization messages, which are not ICV secured. All other Time Synchronization messages are ignored.
	ICV_OPTIONAL	The Timesync module accepts only Time Synchronization messages which are not ICV secured and Time Synchronization messages which are ICV secured and have the correct ICV. All other Time Synchronization messages are ignored.
	ICV_VERIFIED	The Timesync module accepts only Time Synchronization messages, which are ICV secured and have the correct ICV. All other Time Synchronization messages are ignored.
Pre-Compile only	true	

⌋

[ECUC_EthTSyn_00086] Definition of EcucBooleanParamDef EthTSynRxSubTLVStatus ⌈

Parameter Name	EthTSynRxSubTLVStatus
Parent Container	EthTSynGlobalTimeSlaveConfig
Upstream mapping defined	–
Description	Definition of whether or not a Sub-TLV:Status Secured or Sub-TLV:Status Not Secured shall be present and shall be evaluated when processing a received Follow_Up message.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00085] Definition of EcucBooleanParamDef EthTSynRxSubTLVTime ⌈

Parameter Name	EthTSynRxSubTLVTime
Parent Container	EthTSynGlobalTimeSlaveConfig
Upstream mapping defined	–
Description	Definition of whether or not a Sub-TLV:Time Secured shall be present and shall be evaluated when processing a received Follow_Up message
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

[ECUC_EthTSyn_00087] Definition of EcucBooleanParamDef EthTSynRxSubTLVUserData ⌈

Parameter Name	EthTSynRxSubTLVUserData
Parent Container	EthTSynGlobalTimeSlaveConfig
Upstream mapping defined	–
Description	Definition of whether or not a Sub-TLV:UserData Secured or Sub-TLV:UserData Not Secured shall be present and shall be evaluated when processing a received Follow_Up message
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Pre-Compile only	true

⌋

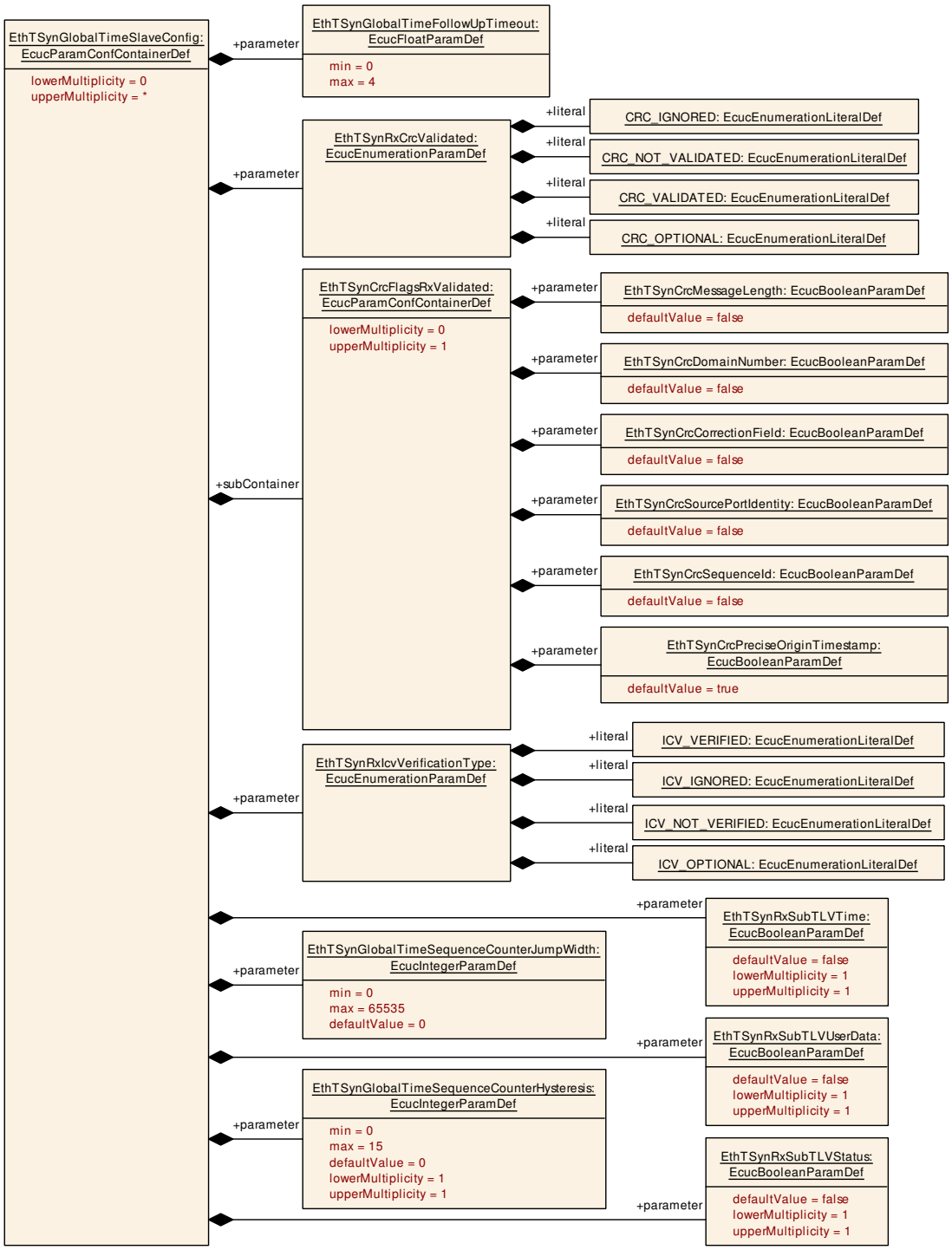


Figure 10.12: EthTSyn_GlobalTimeSlave

10.2.20 EthTSynCrcFlagsRxValidated

[ECUC_EthTSyn_00050] Definition of EcucParamConfContainerDef EthTSynCrc FlagsRxValidated ⌈

Container Name	EthTSynCrcFlagsRxValidated	
Parent Container	EthTSynGlobalTimeSlaveConfig	
Upstream mapping defined	–	
Description	This container collects definitions which parts of the Follow_Up message elements shall be included in CRC validation.	
Multiplicity	0..1	
Post-Build Multiplicity	-	Selectable
Configuration Parameters		

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
EthTSynCrcCorrectionField	1	[ECUC_EthTSyn_00042]
EthTSynCrcDomainNumber	1	[ECUC_EthTSyn_00041]
EthTSynCrcMessageLength	1	[ECUC_EthTSyn_00040]
EthTSynCrcPreciseOriginTimestamp	1	[ECUC_EthTSyn_00045]
EthTSynCrcSequenceId	1	[ECUC_EthTSyn_00044]
EthTSynCrcSourcePortIdentity	1	[ECUC_EthTSyn_00043]

No Included Containers

⌋

[ECUC_EthTSyn_00042] Definition of EcucBooleanParamDef EthTSynCrcCorrectionField ⌈

Parameter Name	EthTSynCrcCorrectionField	
Parent Container	EthTSynCrcFlagsRxValidated, EthTSynCrcTimeFlagsTxSecured	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	correctionField from the Follow_Up Message Header shall be included in CRC calculation.	
Multiplicity	1	
Type	EcucBooleanParamDef	
Default value	false	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00041] Definition of EcucBooleanParamDef EthTSynCrcDomainNumber ⌈

Parameter Name	EthTSynCrcDomainNumber	
Parent Container	EthTSynCrcFlagsRxValidated, EthTSynCrcTimeFlagsTxSecured	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	domainNumber from the Follow_Up Message Header shall be included in CRC calculation.	
Multiplicity	1	
Type	EcucBooleanParamDef	
Default value	false	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00040] Definition of EcucBooleanParamDef EthTSynCrcMessageLength ⌈

Parameter Name	EthTSynCrcMessageLength	
Parent Container	EthTSynCrcFlagsRxValidated, EthTSynCrcTimeFlagsTxSecured	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	messageLength from the Follow_Up Message Header shall be included in CRC calculation.	
Multiplicity	1	
Type	EcucBooleanParamDef	
Default value	false	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00045] Definition of EcucBooleanParamDef EthTSynCrcPreciseOriginTimestamp ⌈

Parameter Name	EthTSynCrcPreciseOriginTimestamp	
Parent Container	EthTSynCrcFlagsRxValidated, EthTSynCrcTimeFlagsTxSecured	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	preciseOriginTimestamp from the Follow_Up Message Field shall be included in CRC calculation.	
Multiplicity	1	
Type	EcucBooleanParamDef	
Default value	true	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00044] Definition of EcucBooleanParamDef EthTSynCrcSequenceId ⌈

Parameter Name	EthTSynCrcSequenceId	
Parent Container	EthTSynCrcFlagsRxValidated, EthTSynCrcTimeFlagsTxSecured	
Upstream mapping defined	[13, CP_TPS_SystemTemplate]	
Description	sequenceId from the Follow_Up Message Header shall be included in CRC calculation.	
Multiplicity	1	
Type	EcucBooleanParamDef	
Default value	false	
Post-Build Value	-	Selectable

⌋

[ECUC_EthTSyn_00043] Definition of EcucBooleanParamDef EthTSynCrcSourcePortIdentity ⌈

Parameter Name	EthTSynCrcSourcePortIdentity
Parent Container	EthTSynCrcFlagsRxValidated, EthTSynCrcTimeFlagsTxSecured
Upstream mapping defined	[13, CP_TPS_SystemTemplate]

▽

△

Description	sourcePortIdentity from the Follow_Up Message Header shall be included in CRC calculation.	
Multiplicity	1	
Type	EcucBooleanParamDef	
Default value	false	
Post-Build Value	-	Selectable

⌋

10.3 Constraints

Note: If a Time Master transmits Timesync messages for a Time Domain via multiple Ethernet controllers, the EthTSyn allows for different configuration options:

1. a Time Domain container (EthTSynGlobalTimeDomain) references multiple Ethernet controllers ((EthTSynGlobalTimeEthIfRef).
2. a Time Domain container references only one Ethernet controller. In this case one Time Domain container needs to be configured per Ethernet controller and each Time Domain is configured using the same Time Domain Id (EthTSyn-GlobalTimeDomainId).

[SWS_EthTSyn_CONSTR_00003] EthTSynGlobalTimeMasterConfig Container Validity for Time Domains ⌈The EthTSynGlobalTimeMasterConfig container exists for Synchronized Time Domains (EthTSynGlobalTimeDomain 0 .. 127) only.⌋

[SWS_EthTSyn_CONSTR_00004] ICV Generation Timeout for Synchronous CSM Jobs ⌈If the CSM job used to generate ICV is configured in synchronous behaviour, the EthTSynIcvGenerationTimeout shall be set to 0.⌋

[SWS_EthTSyn_CONSTR_00005] ICV Verification Timeout for Synchronous CSM Jobs ⌈If the CSM job used to verify ICV is configured in synchronous behavior, the EthTSynIcvVerificationTimeout shall be set to 0.⌋

[SWS_EthTSyn_CONSTR_00006] Tx Period Availability Based on Port Role ⌈The parameter EthTSynGlobalTimeTxPeriod shall determine the transmission interval of Sync messages. It shall be available if and only if the parameter EthTSynPortRole is set to TIME_MASTER or DYNAMIC.⌋

Note: Configuring different EthTSynGlobalTimeTxPeriod for different port requires the involvement of StbM.

[SWS_EthTSyn_CONSTR_00007] Port Role Conflict Within Time Domain ⌈

The parameter EthTSynPortRole shall not be configured as TIME_SLAVE for two ports under same EthTSynGlobalTimeDomain.⌋

[SWS_EthTSyn_CONSTR_00008] Configuration for VLAN Support ⌈

If parameter EthTSynFramePrio exists for a Time Domain, at least one Virtual Ethernet Controller (EthTSynGlobalTimeEthIfRef) referenced by the Time Domain shall refer to a tagged VLAN.⌋

[SWS_EthTSyn_CONSTR_00009] Support of PDUs with `KeepLocalPduBuffer` set to `FALSE` ⌈

The configuration of EthTSynTxPdu and EthTSynRxPdu shall refer to PDUs where KeepLocalPduBuffer is set to FALSE. Otherwise the configuration shall be rejected as invalid.⌋

10.4 Configuration Constraints

An EthTSynGlobalTimeDomain represents the concept of a PTP Instance from [4, IEEE Std 802.1AS-2011] and operates on multiple EthTSynPTPPorts. Since every EthTSynGlobalTimeDomain requires an EthTSynSynchronizedTimeBaseRef only Time Gateways (Chapter 7.10) are possible, i.e. a Time Aware Bridge (or PTP Relay Instance per [4, IEEE Std 802.1AS-2011]) can not be configured here.

In the context of the [13, AUTOSAR System Template] Chapter 9.2 *"The big Picture"* the EthTSynGlobalTimeDomain constitutes a globalTimeSubDomain, since the overarching GlobalTimeDomain there is defined through a hierarchy of TimeBases.

[constr_0002] ICV Generation Timeout for Synchronous CSM Jobs ⌈If the CSM job used to generate ICV is configured in synchronous behavior, the EthTSynIcvGenerationTimeout shall be set to 0.⌋

[constr_0003] ICV Verification Timeout for Synchronous CSM Jobs ⌈If the CSM job used to verify ICV is configured in synchronous behavior, the EthTSynIcvVerificationTimeout shall be set to 0.⌋

[constr_0004] Tx Period Availability Based on Port Role ⌈The parameter EthTSynGlobalTimeTxPeriod shall determine the transmission interval of Sync messages. It shall be available if and only if the parameter EthTSynPortRole is set to TIME_MASTER or DYNAMIC.⌋

[constr_0005] Port Role Conflict Within Time Domain ⌈The parameter EthTSynPortRole shall not be configured as TIME_SLAVE for two ports under same EthTSynGlobalTimeDomain.⌋

[constr_0006] Configuration for VLAN Support ⌈If parameter EthTSynFramePrio exists for a Time Domain, at least one Virtual Ethernet Controller (EthTSynGlobalTimeEthIfRef) referenced by the Time Domain shall refer to a tagged VLAN.⌋

[constr_0007] Support of PDUs with `KeepLocalPduBuffer` set to `FALSE`

Status: DRAFT

⌈The configuration of EthTSynTxPdu and EthTSynRxPdu shall refer to PDUs where KeepLocalPduBuffer is set to FALSE. Otherwise the configuration shall be rejected as invalid.⌋

[constr_0008] Multiple Domains reference a single StbM Time Base

Status: DRAFT

⌈If multiple EthTSynGlobalTimeDomains use the same EthTSynSynchronizedTimeBaseRef, only one of them can have a EthTSynPTPPort in EthTSynPortRole set to TIME_SLAVE. All other EthTSynPTPPorts must have EthTSynPortRole set to TIME_MASTER.⌋

[constr_0009] Multiple StbM Time Bases referenced by single Domain

Status: DRAFT

⌈While technically multiple EthTSynSynchronizedTimeBaseRef could be updated by a single EthTSynGlobalTimeDomain, if no EthTSynPTPPort has EthTSynPortRole set to TIME_MASTER, the multiplicity of EthTSynSynchronizedTimeBaseRef prohibits this.⌋

Note: To transfer the time information of a EthTSynGlobalTimeDomain to multiple TimeBases, [2, AUTOSAR Synchronized Time-Base Manager] Chapter 7.12 *"Time Base Cloning"* should be used.

[constr_0010] Single Pdelay mechanism per Domain

Status: DRAFT

⌈Both EthTSynNeighbourRateRatioConfig and EthTSynLinkDelayConfig technically allow to be configured for CMLDS or TransportSpecific. This could lead to two instances of the Pdelay mechanism to be operating for the same EthTSynGlobalTimeDomain.

- If EthTSynNeighbourRateRatioConfig is configured for CMLDS, EthTSynLinkDelayConfig must not be configured for TransportSpecific
- If EthTSynLinkDelayConfig is configured for CMLDS, EthTSynNeighbourRateRatioConfig must not be configured for TransportSpecific,

⌋

[constr_0011] Consistent Pdelay configuration with CMLDS

Status: DRAFT

⌈

- If either EthTSynNeighbourRateRatioConfig is configured for CMLDS, or EthTSynLinkDelayConfig is configured for CMLDS, then no EthTSynPdelayConfigRef must be configured for the EthTSynPTPPort.

- If either EthTSynNeighbourRateRatioConfig is configured for TransportSpecific, or EthTSynLinkDelayConfig is configured for TransportSpecific, then no EthTSynCMLDSVirtualPortRef must be configured for the same EthTSynVirtualPortRef.

⌋

10.5 Published Information

For details refer to [3] Chapter 10.3 *"Published Information"*.

A Not applicable requirements

[SWS_EthTSyn_NA_00996]

Upstream requirements: RS_TS_00003, RS_TS_00004, RS_TS_00005, RS_TS_00006, RS_TS_00007, RS_TS_00008, RS_TS_00009, RS_TS_00010, RS_TS_00011, RS_TS_00014, RS_TS_00015, RS_TS_00016, RS_TS_00017, RS_TS_00018, RS_TS_00019, RS_TS_00021, RS_TS_00024, RS_TS_00025, RS_TS_00026, RS_TS_00027, RS_TS_00029, RS_TS_00030, RS_TS_00031, RS_TS_00032, RS_TS_00033, RS_TS_00035, RS_TS_00037, RS_TS_00038, RS_TS_00039, RS_TS_20040, RS_TS_20041, RS_TS_20042, RS_TS_20043, RS_TS_20045, RS_TS_20046, RS_TS_20060, RS_TS_20068, RS_TS_20070, RS_TS_20071, RS_TS_20073, RS_TS_20074

⌈This specification item references requirements from RS Time Synchronization [1] that are not applicable to EthTSyn, because they are allocated either to other network specific Time Sync modules (CAN, FlexRay) or to the Synchronized Time-Base Manager (StbM) module.⌋

[SWS_EthTSyn_NA_00997]

Upstream requirements: SRS_BSW_00388, SRS_BSW_00389, SRS_BSW_00390, SRS_BSW_00392, SRS_BSW_00393, SRS_BSW_00395, SRS_BSW_00396, SRS_BSW_00403, SRS_BSW_00478

⌈This specification item references requirements, that cannot be traced because they apply only to EcuC elements.⌋

Note: EcuC elements do not support trace links

[SWS_EthTSyn_NA_00998]

Upstream requirements: SRS_BSW_00345, SRS_BSW_00369, SRS_BSW_00383, SRS_BSW_00384, SRS_BSW_00399, SRS_BSW_00419

⌈This specification item references requirements, that cannot be traced to a specific spec item in the EthTSyn.⌋

Note: These requirements are generic in nature and would affect all or very many requirements or cannot be traced to any requirement at all (but just explanatory chapters of the SWS).

[SWS_EthTSyn_NA_00999]

Upstream requirements: SRS_BSW_00168, SRS_BSW_00170, SRS_BSW_00336, SRS_BSW_00375, SRS_BSW_00416, SRS_BSW_00422, SRS_BSW_00425, SRS_BSW_00432, SRS_BSW_00458, SRS_BSW_00461, SRS_BSW_00469, SRS_BSW_00470, SRS_BSW_00471, SRS_BSW_00472

⌈These requirements are not applicable to EthTSyn.⌋

B Change history of AUTOSAR traceable items

Please note that the lists in this chapter also include traceable items that have been removed from the specification in a later version. These items do not appear as hyperlinks in the document.

B.1 Traceable item history of this document according to AUTOSAR Release R26-11

B.1.1 Added Specification Items in R26-11

[SWS_EthTSyn_00425] [SWS_EthTSyn_00426] [SWS_EthTSyn_00427] [SWS_EthTSyn_00428] [SWS_EthTSyn_00429] [SWS_EthTSyn_00430] [SWS_EthTSyn_00431] [SWS_EthTSyn_00432] [SWS_EthTSyn_00433] [SWS_EthTSyn_00434] [SWS_EthTSyn_00435] [SWS_EthTSyn_00436] [SWS_EthTSyn_91003] [SWS_EthTSyn_91004]

B.1.2 Changed Specification Items in R26-11

[ECUC_EthTSyn_00001] [ECUC_EthTSyn_00002] [ECUC_EthTSyn_00003] [ECUC_EthTSyn_00004] [ECUC_EthTSyn_00005] [ECUC_EthTSyn_00007] [ECUC_EthTSyn_00008] [ECUC_EthTSyn_00009] [ECUC_EthTSyn_00010] [ECUC_EthTSyn_00012] [ECUC_EthTSyn_00013] [ECUC_EthTSyn_00018] [ECUC_EthTSyn_00029] [ECUC_EthTSyn_00030] [ECUC_EthTSyn_00031] [ECUC_EthTSyn_00032] [ECUC_EthTSyn_00033] [ECUC_EthTSyn_00034] [ECUC_EthTSyn_00035] [ECUC_EthTSyn_00036] [ECUC_EthTSyn_00037] [ECUC_EthTSyn_00039] [ECUC_EthTSyn_00040] [ECUC_EthTSyn_00041] [ECUC_EthTSyn_00042] [ECUC_EthTSyn_00043] [ECUC_EthTSyn_00044] [ECUC_EthTSyn_00045] [ECUC_EthTSyn_00046] [ECUC_EthTSyn_00047] [ECUC_EthTSyn_00048] [ECUC_EthTSyn_00049] [ECUC_EthTSyn_00050] [ECUC_EthTSyn_00057] [ECUC_EthTSyn_00058] [ECUC_EthTSyn_00059] [ECUC_EthTSyn_00060] [ECUC_EthTSyn_00061] [ECUC_EthTSyn_00063] [ECUC_EthTSyn_00065] [ECUC_EthTSyn_00066] [ECUC_EthTSyn_00068] [ECUC_EthTSyn_00070] [ECUC_EthTSyn_00071] [ECUC_EthTSyn_00074] [ECUC_EthTSyn_00075] [ECUC_EthTSyn_00076] [ECUC_EthTSyn_00081] [ECUC_EthTSyn_00082] [ECUC_EthTSyn_00083] [ECUC_EthTSyn_00084] [ECUC_EthTSyn_00085] [ECUC_EthTSyn_00086] [ECUC_EthTSyn_00087] [ECUC_EthTSyn_00089] [ECUC_EthTSyn_00090] [ECUC_EthTSyn_00091] [ECUC_EthTSyn_00092] [ECUC_EthTSyn_00093] [ECUC_EthTSyn_00094] [ECUC_EthTSyn_00095] [ECUC_EthTSyn_00096] [ECUC_EthTSyn_00097] [ECUC_EthTSyn_00098] [ECUC_EthTSyn_00099] [ECUC_EthTSyn_00100] [ECUC_EthTSyn_00101] [ECUC_EthTSyn_00103] [ECUC_EthTSyn_00104] [ECUC_EthTSyn_00105] [ECUC_EthTSyn_00106] [ECUC_EthTSyn_00107] [ECUC_EthTSyn_00108] [ECUC_EthTSyn_00109] [ECUC_EthTSyn_00110] [ECUC_EthTSyn_00112] [ECUC_EthTSyn_00113] [ECUC_EthTSyn_00114]

[ECUC_EthTSyn_00115] [ECUC_EthTSyn_00116] [ECUC_EthTSyn_00117] [ECUC_EthTSyn_00119] [ECUC_EthTSyn_00120] [ECUC_EthTSyn_00121] [ECUC_EthTSyn_00122] [ECUC_EthTSyn_00123] [ECUC_EthTSyn_00124] [ECUC_EthTSyn_00125] [ECUC_EthTSyn_00126] [ECUC_EthTSyn_00127] [ECUC_EthTSyn_00128] [ECUC_EthTSyn_00129] [ECUC_EthTSyn_00130] [ECUC_EthTSyn_00131] [ECUC_EthTSyn_00132] [ECUC_EthTSyn_00133] [ECUC_EthTSyn_00134] [ECUC_EthTSyn_00135] [ECUC_EthTSyn_00136] [ECUC_EthTSyn_00137] [ECUC_EthTSyn_00138] [ECUC_EthTSyn_00139] [ECUC_EthTSyn_00140] [ECUC_EthTSyn_00141] [ECUC_EthTSyn_00142] [ECUC_EthTSyn_00143] [ECUC_EthTSyn_00144] [ECUC_EthTSyn_00145] [SWS_EthTSyn_00031] [SWS_EthTSyn_00047] [SWS_EthTSyn_00162] [SWS_EthTSyn_00233] [SWS_EthTSyn_00236] [SWS_EthTSyn_00330]

B.1.3 Deleted Specification Items in R26-11

[ECUC_EthTSyn_00015] [SWS_EthTSyn_00036]

B.1.4 Added Constraints in R26-11

none

B.1.5 Changed Constraints in R26-11

none

B.1.6 Deleted Constraints in R26-11

none

B.2 Traceable item history of this document according to AUTOSAR Release R25-11

B.2.1 Added Specification Items in R25-11

[ECUC_EthTSyn_00128] [ECUC_EthTSyn_00129] [ECUC_EthTSyn_00130] [ECUC_EthTSyn_00131] [ECUC_EthTSyn_00132] [ECUC_EthTSyn_00133] [ECUC_EthTSyn_00134] [ECUC_EthTSyn_00135] [ECUC_EthTSyn_00136] [ECUC_EthTSyn_00137] [ECUC_EthTSyn_00138] [ECUC_EthTSyn_00139] [ECUC_EthTSyn_00140] [ECUC_EthTSyn_00141] [ECUC_EthTSyn_00142] [ECUC_EthTSyn_00143] [ECUC_EthTSyn_00144] [ECUC_EthTSyn_00145] [SWS_EthTSyn_00421] [SWS_EthTSyn_00422] [SWS_EthTSyn_00423] [SWS_EthTSyn_00424]

B.2.2 Changed Specification Items in R25-11

[ECUC_EthTSyn_00001] [ECUC_EthTSyn_00002] [ECUC_EthTSyn_00003] [ECUC_EthTSyn_00004] [ECUC_EthTSyn_00005] [ECUC_EthTSyn_00007] [ECUC_EthTSyn_00008] [ECUC_EthTSyn_00009] [ECUC_EthTSyn_00010] [ECUC_EthTSyn_00012] [ECUC_EthTSyn_00013] [ECUC_EthTSyn_00015] [ECUC_EthTSyn_00018] [ECUC_EthTSyn_00029] [ECUC_EthTSyn_00030] [ECUC_EthTSyn_00031] [ECUC_EthTSyn_00032] [ECUC_EthTSyn_00033] [ECUC_EthTSyn_00034] [ECUC_EthTSyn_00035] [ECUC_EthTSyn_00036] [ECUC_EthTSyn_00037] [ECUC_EthTSyn_00039] [ECUC_EthTSyn_00040] [ECUC_EthTSyn_00041] [ECUC_EthTSyn_00042] [ECUC_EthTSyn_00043] [ECUC_EthTSyn_00044] [ECUC_EthTSyn_00045] [ECUC_EthTSyn_00046] [ECUC_EthTSyn_00047] [ECUC_EthTSyn_00048] [ECUC_EthTSyn_00049] [ECUC_EthTSyn_00050] [ECUC_EthTSyn_00057] [ECUC_EthTSyn_00058] [ECUC_EthTSyn_00059] [ECUC_EthTSyn_00060] [ECUC_EthTSyn_00061] [ECUC_EthTSyn_00063] [ECUC_EthTSyn_00065] [ECUC_EthTSyn_00066] [ECUC_EthTSyn_00068] [ECUC_EthTSyn_00070] [ECUC_EthTSyn_00071] [ECUC_EthTSyn_00074] [ECUC_EthTSyn_00075] [ECUC_EthTSyn_00076] [ECUC_EthTSyn_00081] [ECUC_EthTSyn_00082] [ECUC_EthTSyn_00083] [ECUC_EthTSyn_00084] [ECUC_EthTSyn_00085] [ECUC_EthTSyn_00086] [ECUC_EthTSyn_00087] [ECUC_EthTSyn_00089] [ECUC_EthTSyn_00090] [ECUC_EthTSyn_00091] [ECUC_EthTSyn_00092] [ECUC_EthTSyn_00093] [ECUC_EthTSyn_00094] [ECUC_EthTSyn_00095] [ECUC_EthTSyn_00096] [ECUC_EthTSyn_00097] [ECUC_EthTSyn_00098] [ECUC_EthTSyn_00099] [ECUC_EthTSyn_00100] [ECUC_EthTSyn_00101] [ECUC_EthTSyn_00103] [ECUC_EthTSyn_00104] [ECUC_EthTSyn_00105] [ECUC_EthTSyn_00106] [ECUC_EthTSyn_00107] [ECUC_EthTSyn_00108] [ECUC_EthTSyn_00109] [ECUC_EthTSyn_00110] [ECUC_EthTSyn_00112] [ECUC_EthTSyn_00113] [ECUC_EthTSyn_00114] [ECUC_EthTSyn_00115] [ECUC_EthTSyn_00116] [ECUC_EthTSyn_00117] [ECUC_EthTSyn_00119] [ECUC_EthTSyn_00120] [ECUC_EthTSyn_00121] [ECUC_EthTSyn_00122] [ECUC_EthTSyn_00123] [ECUC_EthTSyn_00124] [ECUC_EthTSyn_00125] [ECUC_EthTSyn_00126] [ECUC_EthTSyn_00127] [SWS_EthTSyn_00014] [SWS_EthTSyn_00029] [SWS_EthTSyn_00031] [SWS_EthTSyn_00032] [SWS_EthTSyn_00047] [SWS_EthTSyn_00104] [SWS_EthTSyn_00127] [SWS_EthTSyn_00162] [SWS_EthTSyn_00172] [SWS_EthTSyn_00187] [SWS_EthTSyn_00200] [SWS_EthTSyn_00201] [SWS_EthTSyn_00202] [SWS_EthTSyn_00203] [SWS_EthTSyn_00210] [SWS_EthTSyn_00213] [SWS_EthTSyn_00214] [SWS_EthTSyn_00215] [SWS_EthTSyn_00216] [SWS_EthTSyn_00217] [SWS_EthTSyn_00227] [SWS_EthTSyn_00228] [SWS_EthTSyn_00229] [SWS_EthTSyn_00231] [SWS_EthTSyn_00232] [SWS_EthTSyn_00236] [SWS_EthTSyn_00243] [SWS_EthTSyn_00250] [SWS_EthTSyn_00256] [SWS_EthTSyn_00261] [SWS_EthTSyn_00263] [SWS_EthTSyn_00264] [SWS_EthTSyn_00265] [SWS_EthTSyn_00266] [SWS_EthTSyn_00330] [SWS_EthTSyn_00331] [SWS_EthTSyn_00412] [SWS_EthTSyn_00413] [SWS_EthTSyn_00414] [SWS_EthTSyn_00417] [SWS_EthTSyn_91001] [SWS_EthTSyn_91002] [SWS_EthTSyn_92000] [SWS_EthTSyn_92001] [SWS_EthTSyn_92002] [SWS_EthTSyn_92003]

B.2.3 Deleted Specification Items in R25-11

[ECUC_EthTSyn_00067] [ECUC_EthTSyn_00069] [ECUC_EthTSyn_00118]