Document Title	Specification of Time Synchronization over Ethernet
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	676

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R23-11

Document Change History

Date	Release	Changed by	Description
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

Contents

1 Introduction and functional overview

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

In addition to what is specified in [1, PRS Time Synchronization 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.

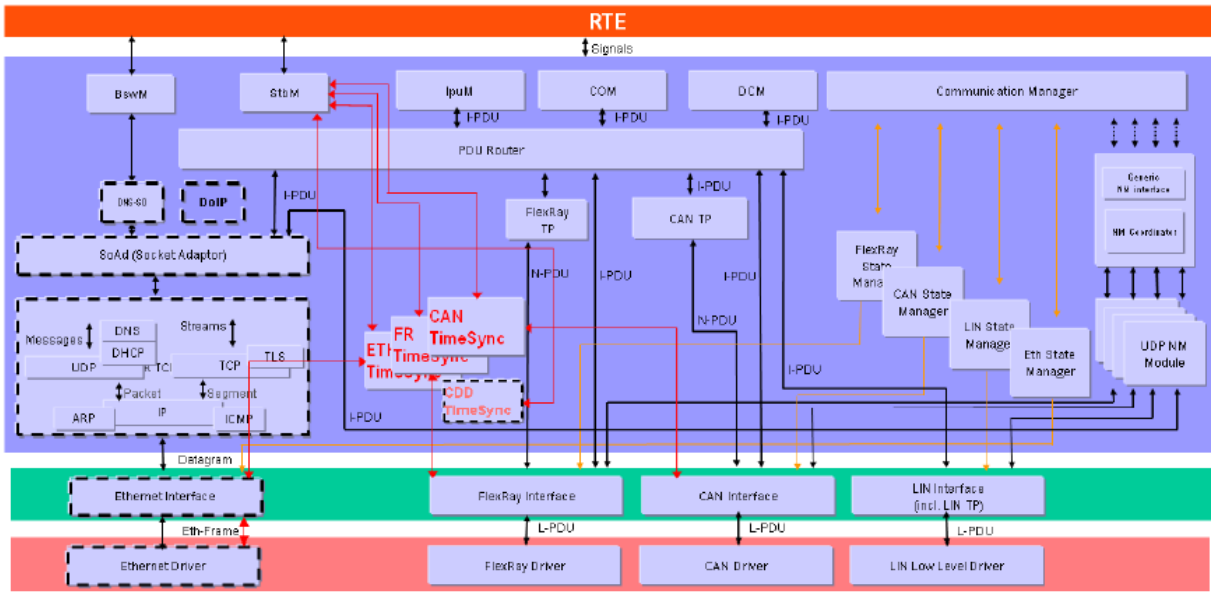

Figure 1.1: Timesync modules in the AUTOSAR Layered Architecture

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.

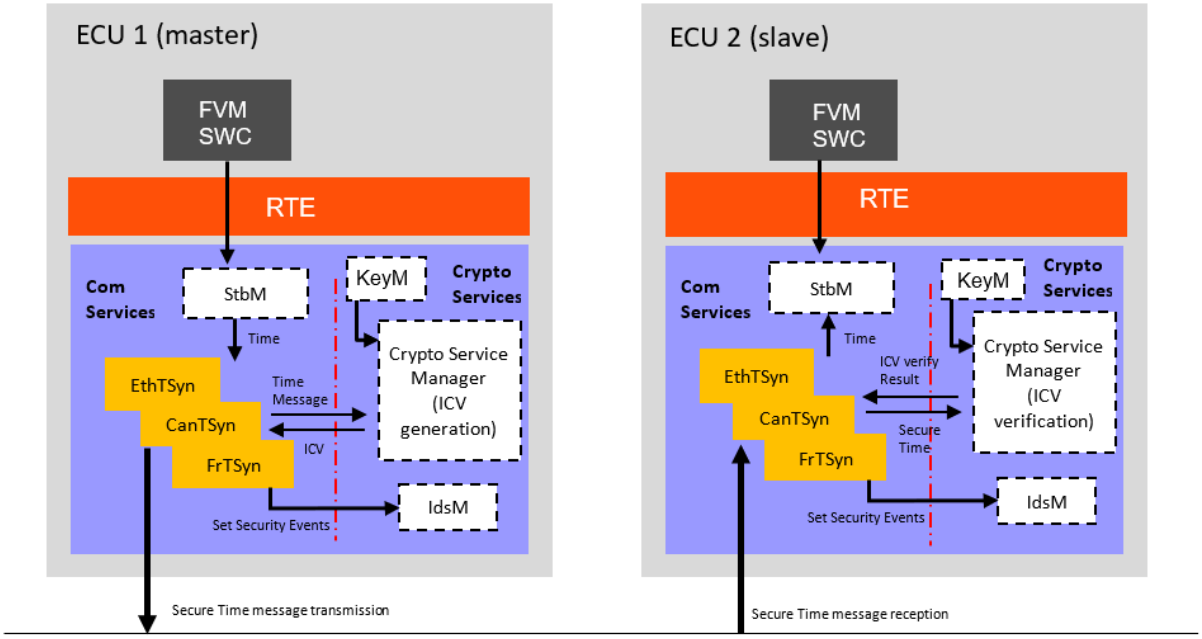

Figure 1.2: Timesync modules interface with security modules in the AUTOSAR Layered Architecture

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
MAC [context - Ethernet protocol]	Media Access Control
MAC [context - security]	Message Authentication Code
meanPropagationDelay	meanPropagationDelay as defined by IEEE 802.1 AS
neighborRateRatio	neighborRateRatio as defined by IEEE 802.1 AS
OFS	Offset synchronization
Pdelay	Propagation / path delay as given in IEEE 802.1AS
Pdelay_Req	Propagation / path delay request message
Pdelay_Resp	Propagation / path delay response message
Pdelay_Resp_Follow_Up	Propagation / path delay Follow-Up message
PDU	Protocol Data Unit
PTP	Precision Time Protocol
rateRatio	rateRatio as defined by IEEE 802.1 AS
StbM	Synchronized Time-Base Manager
Timesync	Time Synchronization
Sync	Time synchronization message (Sync)
TG	Time Gateway
TLV	Type, Length, Value field (acc. to IEEE 802.1AS)
TS	Time Slave
TSD	Time Sub-domain
VLAN	Virtual Local Area Network

3 Related documentation

3.1 Input documents

[1] Time Synchronization Protocol Specification
AUTOSAR_FO_PRS_TimeSyncProtocol

[2] Specification of Synchronized Time-Base Manager
AUTOSAR_CP_SWS_SynchronizedTimeBaseManager

[3] General Specification of Basic Software Modules
AUTOSAR_CP_SWS_BSWGeneral

[4] IEEE Standard 802.1AS-2011

[5] Explanation of Time Sensitive Network features
AUTOSAR_FO_EXP_TimeSensitiveNetworkFeatures

[6] Requirements on Time Synchronization
AUTOSAR_FO_RS_TimeSync

[7] General Requirements on Basic Software Modules
AUTOSAR_CP_SRS_BSWGeneral

[8] Specification of Crypto Service Manager
AUTOSAR_CP_SWS_CryptoServiceManager

[9] Specification of CRC Routines
AUTOSAR_CP_SWS_CRCLibrary

[10] 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 Protocol] which is also valid for EthTSyn.

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

4 Constraints and assumptions

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 802.1 AS] states, that IEEE 802.1AS messages shall not have a VLAN tag nor a priority tag, EthTSyn would allow Time Synchronization on VLANs under the condition, that the switch HW supports forwarding of reserved multicast address using the range of 01:80:C2:00:00:00 .. 0F.
- "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.
- While multidrop topology is used, pDelay measurement are not supported and shall be set to static value.
- No support of securing the messages of PDelay protocol.

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 [5, EXP Time Sensitive Network Features], chapter "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 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
- 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

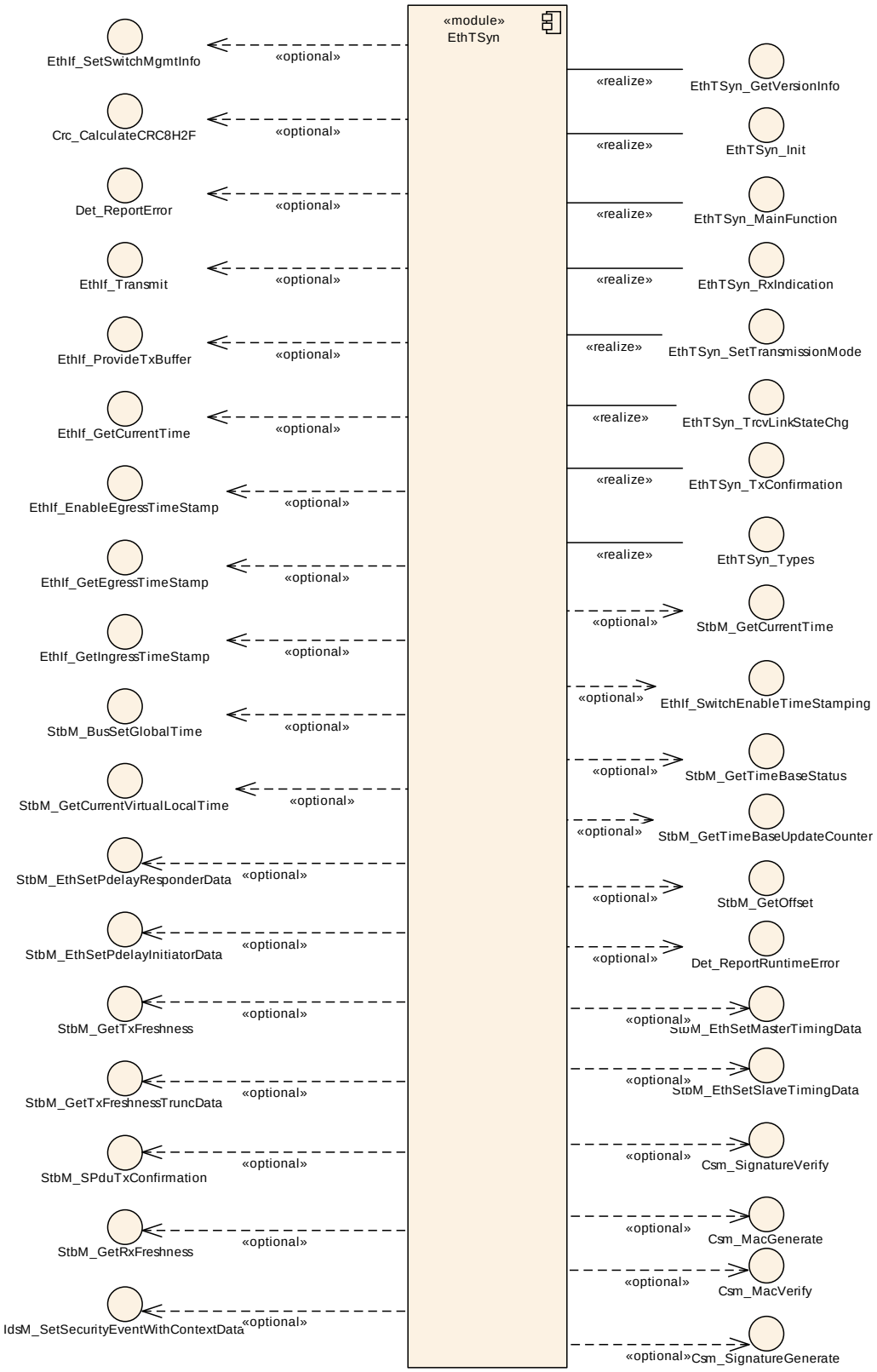

Figure 5.1: Module dependencies of the EthTSyn module

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 [6, RS TimeSync] and [7, 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_Ids_00810]	Basic SW security events	[SWS_EthTSyn_00231] [SWS_EthTSyn_00261] [SWS_EthTSyn_00262]
[RS_TS_00002]	The Implementation of Time Synchronization shall maintain its own Time Base independently of the acting role.	[SWS_EthTSyn_00210]
[RS_TS_00034]	The Implementation of Time Synchronization shall provide measurement data to the application	[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]	The Timesync over Ethernet module shall 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_00136] [SWS_EthTSyn_00137] [SWS_EthTSyn_00139] [SWS_EthTSyn_00187] [SWS_EthTSyn_00202] [SWS_EthTSyn_00211] [SWS_EthTSyn_00400] [SWS_EthTSyn_00401]
[RS_TS_20048]	The Timesync over Ethernet module shall support IEEE 802.1AS as well as 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_00036] [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_00148] [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]
[RS_TS_20051]	The Timesync over Ethernet module shall detect and handle errors in synchronization protocol / communication	[SWS_EthTSyn_00019] [SWS_EthTSyn_00020] [SWS_EthTSyn_00021] [SWS_EthTSyn_00022] [SWS_EthTSyn_00029] [SWS_EthTSyn_00129] [SWS_EthTSyn_00145] [SWS_EthTSyn_00146]
[RS_TS_20052]	The configuration of the Time Synchronization over Ethernet module shall allow the module to work as a Time Master	[SWS_EthTSyn_00051]

△

Requirement	Description	Satisfied by
[RS_TS_20053]	The configuration of the Time Synchronization over Ethernet module shall allow the module to work as a Time Slave	[SWS_EthTSyn_00051]
[RS_TS_20054]	The Implementation of the Time Synchronization shall evaluate and propagate Time Gateway relevant information	[SWS_EthTSyn_00051]
[RS_TS_20058]	The Timesync over Ethernet module shall provide the precision of Synchronized Time Bases	[SWS_EthTSyn_00150]
[RS_TS_20059]	The Timesync over Ethernet module shall access all communication ports belonging to Time Synchronization	[SWS_EthTSyn_00031] [SWS_EthTSyn_00047]
[RS_TS_20061]	The Timesync over Ethernet module shall support means to protect the Time Synchronization protocol	[SWS_EthTSyn_00080] [SWS_EthTSyn_00086] [SWS_EthTSyn_00087] [SWS_EthTSyn_00096] [SWS_EthTSyn_00111]
[RS_TS_20062]	The Timesync over Ethernet module shall support user specific data within the time measurement and synchronization protocol	[SWS_EthTSyn_00080] [SWS_EthTSyn_00086] [SWS_EthTSyn_00087] [SWS_EthTSyn_00230]
[RS_TS_20063]	The Timesync over Ethernet module shall use the Time Synchronization protocol for Synchronized Time Bases to transmit and receive Offset Time Bases	[SWS_EthTSyn_00198] [SWS_EthTSyn_00199]
[RS_TS_20066]	The Timesync over Ethernet module shall support measuring the peer-to-peer delay using the IEEE 802.1AS peer-to-peer delay mechanism.	[SWS_EthTSyn_00200] [SWS_EthTSyn_00201] [SWS_EthTSyn_00224] [SWS_EthTSyn_00225]
[RS_TS_20069]	The TimeSync over Ethernet module shall provide read / write access to bus protocol specific parameters	[SWS_EthTSyn_00226] [SWS_EthTSyn_00227]
[RS_TS_20072]	The Timesync over Ethernet module shall support means to secure the Time Synchronization protocol	[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_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]

▽

△

Requirement	Description	Satisfied by
[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]	A static status variable denoting if a BSW module is initialized shall be initialized with value 0 before any APIs of the BSW module is called	[SWS_EthTSyn_00030]

Table 6.1: RequirementsTracing

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 Protocol].

7.1.2 VLAN Support

[SWS_EthTSyn_00148] ⌈If the parameter `EthTSynFramePrio` exists, the `EthTSynGlobalTimeEthIfRef` shall refer to a Virtual Ethernet Controller representing a VLAN.⌋*(RS_TS_20048)*

[SWS_EthTSyn_00162] ⌈Time Slave and Time Master shall use the`EthTSynFramePrio` value as priority parameter when calling `EthIf_ProvideTxBuffer`.⌋*(RS_TS_20048)*

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

7.2 Initialization

The Global Time Synchronization over Ethernet is initialized via `EthTSyn_Init`. Except for `EthTSyn_GetVersionInfo` and `EthTSyn_Init`, the API functions of the EthTSyn module may only be called when the module has been properly initialized.

[SWS_EthTSyn_00006] ⌈A call to `EthTSyn_Init` initializes all internal variables and sets the EthTSyn module to the initialized state.⌋*(SRS_BSW_00101)*

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

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 Figure 9.4, Figure 9.5, and Figure 9.6).
- The conversion can happen linearly, i.e., no rate correction terms need to be determined and applied.

[SWS_EthTSyn_00210] ⌈EthTSyn shall discard a timestamp derived from the Ethernet Controller HW (e.g., via `EthIf_GetCurrentTimeTuple`, `EthIf_GetIngressTimeStamp` or `EthIf_GetEgressTimeStamp`), if the quality of the timestamp (refer to `Eth_TimeStampQualType`) is indicated as `ETH_INVALID` or `ETH_UNCERTAIN`.⌋*(RS_TS_00002)*

7.4 Debounce Time

[SWS_EthTSyn_00130] ⌈If `EthTSynGlobalTimeDebounceTime` is set to 0, Eth TSyn shall ignore any debouncing.⌋*(RS_TS_20047)*

[SWS_EthTSyn_00131] ⌈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.
⌋*(RS_TS_20047)*

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

[SWS_EthTSyn_00132] ⌈`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.
⌋*(RS_TS_20047)*

[SWS_EthTSyn_00133] ⌈A new Timesync PDU shall only be sent, if the corresponding `debounceCounter` has reached 0.⌋*(RS_TS_20047)*

[SWS_EthTSyn_00187] ⌈Each port of a `EthTSynGlobalTimeDomain` shall have its own `debounceCounter`.⌋*(RS_TS_20047)*

7.5 Pdelay Protocol for Latency Calculation

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

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

7.5.1 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] ⌈If Master and Time Slave transmit Pdelay_Req for latency calculation with the cycle (refer to PRS_TS_00011 in [1, PRS Time Synchronization Protocol]), the following sequence shall be applied:

1. Get a free transmission buffer via `EthIf_ProvideTxBuffer`
2. Activate the time stamping via `EthIf_EnableEgressTimeStamp` if `EthTSynHardwareTimestampSupport` is set to TRUE
3. Trigger transmit request via `EthIf_Transmit`.

⌋*(RS_TS_20048, RS_TS_20066)*

[SWS_EthTSyn_00201] ⌈If Time Master and Time Slave transmit Pdelay_Resp for latency calculation (refer to PRS_TS_00012 in [1, PRS Time Synchronization Protocol]) the following sequence shall be applied:

1. Get a free transmission buffer via `EthIf_ProvideTxBuffer`
2. Activate the time stamping via `EthIf_EnableEgressTimeStamp` if `EthTSynHardwareTimestampSupport` is set to TRUE
3. Trigger transmit request via `EthIf_Transmit`

⌋*(RS_TS_20048, RS_TS_20066)*

[SWS_EthTSyn_00013] ⌈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.⌋*(RS_TS_20048)*

[SWS_EthTSyn_00123] ⌈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.⌋*(RS_TS_20048)*

[SWS_EthTSyn_00159] ⌈On invocation of EthTSyn_TxConfirmation with parameter Result equal to E_OK the egress timestamp shall be retrieved for t3 from the Eth If 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.⌋*(RS_TS_20048)*

[SWS_EthTSyn_00122] ⌈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.⌋*(RS_TS_20048)*

[SWS_EthTSyn_00225] ⌈The Time Master shall set responseOriginTimestamp (for the Pdelay_Resp_Follow_Up message) to t3.⌋*(RS_TS_20066)*

[SWS_EthTSyn_00014] ⌈If EthTSynGlobalTimePdelayRespEnable is set to TRUE, Time Master and Time Slave shall transmit Pdelay_Resp_Follow_Up with the transmission timestamp of that messages as defined in [SWS_EthTSyn_00159] as well as defined in [1, PRS Time Synchronization Protocol] chapter "Propagation delay measurement" considering debounceCounter which represents a time offset between Pdelay_Resp and Pdelay_Resp_Follow_Up.

For that, the following sequence shall be applied:

1. Get a free transmission buffer via EthIf_ProvideTxBuffer
2. Trigger transmit request with the transmission timestamp of [SWS_EthTSyn_00159] via EthIf_Transmit.

⌋*(RS_TS_20048)*

7.5.2 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 Figure 9.5, Figure 9.6.

[SWS_EthTSyn_00160] ⌈On invocation of `EthTSyn_RxIndication` the ingress timestamp t2 shall be retrieved from the EthIf via `EthIf_GetIngressTimeStamp` on ingress 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 ingress time stamp to the Virtual Local Time as used in the StbM.⌋*(RS_TS_20048)*

[SWS_EthTSyn_00124] ⌈On invocation of `EthTSyn_RxIndication` the ingress timestamp shall be retrieved for t2 from the StbM via `StbM_GetCurrentVirtual-LocalTime` on ingress of Pdelay_Req message, if `EthTSynHardwareTimestamp-Support` is set to FALSE.⌋*(RS_TS_20048)*

[SWS_EthTSyn_00224] ⌈The Time Master shall set `requestReceiptTimestamp` (to be used in the Pdelay_Resp message) to t2.⌋*(RS_TS_20066)*

[SWS_EthTSyn_00049] ⌈On invocation of `EthTSyn_RxIndication` the ingress time stamp shall be retrieved for t4 from the EthIf via E`EthIf_GetIngressTimeS-tamp` on ingress of the Pdelay_Resp message, if `EthTSynHardwareTimestamp-Support` 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 ingress time stamp to the Virtual Local Time as used in the StbM.⌋*(RS_TS_20048)*

[SWS_EthTSyn_00161] ⌈On invocation of `EthTSyn_RxIndication` the ingress time stamp shall be retrieved for t4 on ingress of the Pdelay_Resp message from the StbM via `StbM_GetCurrentVirtualLocalTime`, if `EthTSynHardwareTimes-tampSupport` is set to FALSE.⌋*(RS_TS_20048)*

[SWS_EthTSyn_00263]{DRAFT} ⌈If

- configuration parameter `EthTSynRateRatioEnable` is set to `TRUE`
- 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].⌋*(RS_TS_20048)*

[SWS_EthTSyn_00264]{DRAFT} ⌈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].⌋*(RS_TS_20048)*

7.6 Message Format

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

7.6.1 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 Protocol] .

7.6.2 Sync and Follow_Up acc. to AUTOSAR

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

7.6.2.1 Follow_Up Message Header [AUTOSAR]

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

7.6.2.2 AUTOSAR and OEM Sub-TLV's

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

7.6.2.2.1 AUTOSAR Sub-TLV: Time Secured

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

7.6.2.2.2 AUTOSAR Sub-TLV: Status Secured / Not Secured

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

7.6.2.2.3 AUTOSAR Sub-TLV: UserData Secured / Not Secured

[SWS_EthTSyn_00080] ⌈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.StbM_UserDataLength` and vice versa.⌋*(RS_TS_20061, RS_TS_20062)*

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

7.6.2.2.4 AUTOSAR Sub-TLV: OFS Secured / Not Secured

the corresponding AUTOSAR Sub-TLV: OFS shall be mapped to the Follow_Up Message of that Synchronized Time Domain.

[SWS_EthTSyn_00086] ⌈If a Offset Time Domain on Ethernet references a Synchronized Time Domain on Ethernet (refer to parameter `StbMOffsetTimeBase` in the StbM), the corresponding AUTOSAR Sub-TLV: OFS shall be mapped to the Follow_Up Message of that Synchronized Time Domain.⌋*(RS_TS_20061, RS_TS_20062)*

[SWS_EthTSyn_00087] ⌈The User Data of the AUTOSAR Sub-TLV: OFS shall be mapped to the `StbM_UserDataType`, whereas the 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.StbM_UserDataLength` and vice versa.⌋*(RS_TS_20061, RS_TS_20062)*

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

7.6.2.2.5 AUTOSAR Sub-TLV: Time Authenticated

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

7.7 Acting as Time Master

Refer to chapter Acting as Time Master in [1] for additional requirements.

If the EthTSyn is configured as a Time Master for Time Domain, the EthTSyn module checks on each `EthTSyn_MainFunction` call the necessity for a Timesync message transmission for that Time Domain.

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

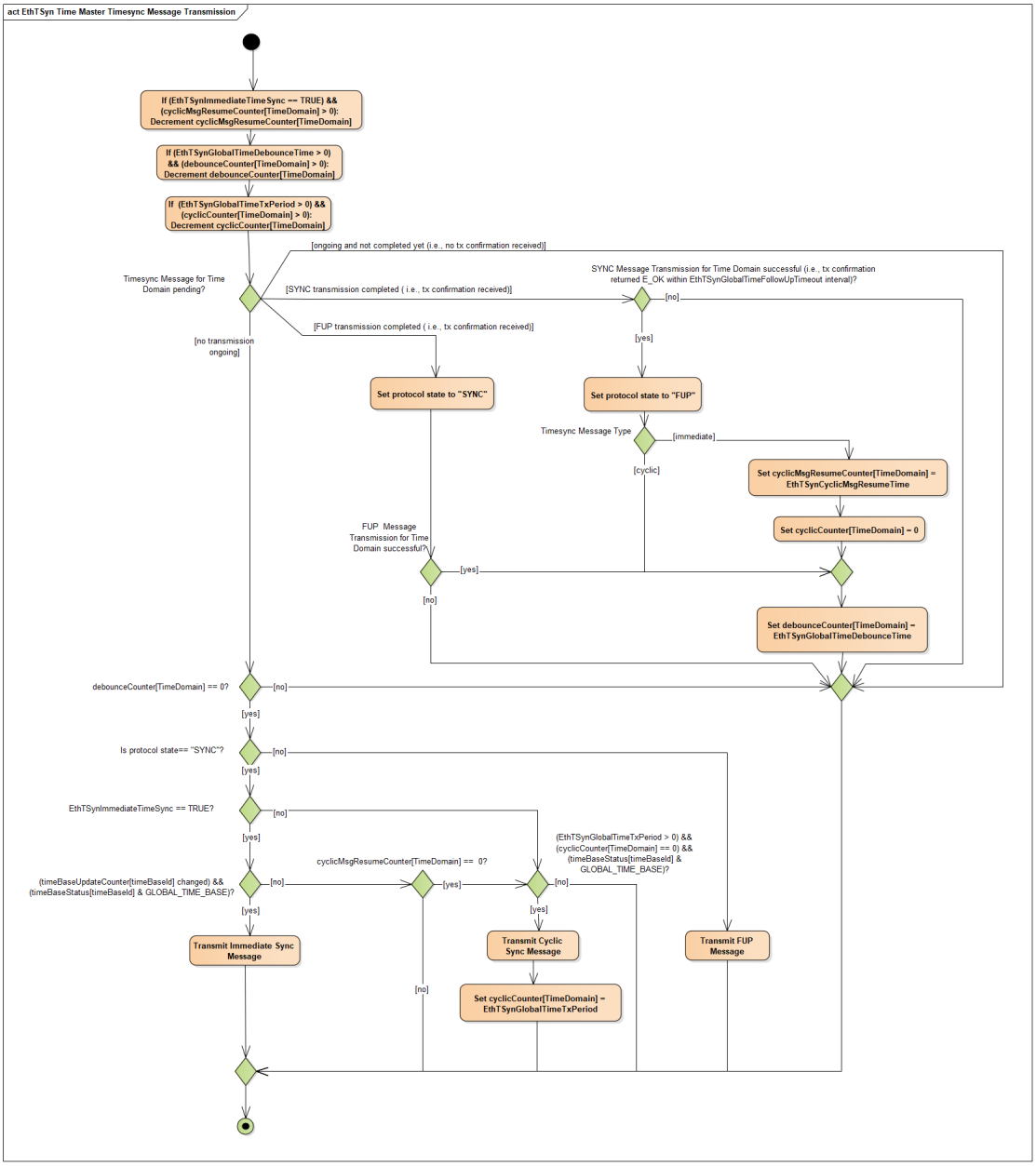

Figure 7.1: Timesync Message Transmission

7.7.1 Message processing

[SWS_EthTSyn_00265]{DRAFT} ⌈When `EthTSynGlobalTimePortRole` 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] for additional requirements.

[SWS_EthTSyn_00202] ⌈If the Time Master transmits a Sync message (refer to [PRS_TS_00016] in [1]), the following sequence shall be applied:

- The Global Time Tuple $[T0; T0_{VLT}]$ shall be retrieved from the StbM via StbM_GetCurrentTime according to EthTSyn Egress Time Stamping.
- Get a free transmission buffer via EthIf_Provide_TxBuffer
- Activate the time stamping via EthIf_EnableEgressTimeStamp if EthTSynHardwareTimestampSupport is set to TRUE
- Trigger transmit request via EthIf_Transmit

⌋*(RS_TS_20047, RS_TS_20048)*

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

Note: EthTSyn Egress Time Stamping is shown in Figure 9.4.

[SWS_EthTSyn_00211] ⌈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.⌋*(RS_TS_20047)*

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] ⌈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.

⌋*(RS_TS_20048)*

Note: EthTSyn_Egress_Time_Stamping is shown in Figure 9.4

[SWS_EthTSyn_00017] ⌈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 the EthIf 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 $T0 - (T3_{VLT} - T2_{VLT}) + (T4_{VLT} - T0_{VLT})$

⌋(*RS_TS_20048*)

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] ⌈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 $T0 + (T2_{VLT} - T0_{VLT})$.⌋(*RS_TS_20048*)

[SWS_EthTSyn_00189] ⌈If EthTSynHardwareTimestampSupport is set to FALSE the preciseOriginTimestamp shall be calculated as $T0 + (T4_{VLT} - T0_{VLT})$.⌋(*RS_TS_20048*)

[SWS_EthTSyn_00204] ⌈The Time Master shall consider the debounceCounter, which represents a time offset between Sync and Follow_Up message, before transmitting the Follow_Up message.⌋(*RS_TS_20048*)

[SWS_EthTSyn_00226] ⌈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

⌋(*RS_TS_20069*)

[SWS_EthTSyn_00203] ⌈If the Time Master transmits a Follow_Up message (refer to [PRS_TS_00018] in [1]), the following sequence shall be applied:

- Get a free transmission buffer via EthIf_Provide_TxBuffer

- Trigger transmit request with the transmission timestamp of [SWS_EthTSyn_00017] via `EthIf_Transmit`

⌋(*RS_TS_20048*)

7.7.1.1 Runtime Error detection

[SWS_EthTSyn_00145] ⌈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.⌋(*RS_TS_20051*)

7.7.1.2 Frame Debouncing

Refer to chapter Frame Debouncing in [1].

7.7.1.3 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] ⌈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.⌋(*RS_TS_20047*)

[SWS_EthTSyn_00135] ⌈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.⌋(*RS_TS_20047*)

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

The debounceCounter as described in section 7.4 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_00136]{OBSOLETE} ⌈If EthTSynImmediateTimeSync is set to TRUE, EthTSynCyclicMsgResumeTime shall be considered.⌋*(RS_TS_20047)*

[SWS_EthTSyn_00137] ⌈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.⌋*(RS_TS_20047)*

[SWS_EthTSyn_00400]{DRAFT} ⌈While for a Time Domain:

- cyclicMsgResumeCounter is greater than 0

EthTSyn shall discard cyclic Timesync message transmission requests for that Time Domain.⌋*(RS_TS_20047)*

[SWS_EthTSyn_00401] ⌈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.⌋*(RS_TS_20047)*

[SWS_EthTSyn_00139] ⌈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.⌋ *(RS_TS_20047)*

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`.

7.7.1.4 Secure Time Synchronization

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

[SWS_EthTSyn_00246]{DRAFT} ⌈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`.⌋*(RS_TS_20072)*
Note:(refer)

[SWS_EthTSyn_00247]{DRAFT} ⌈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`.⌋ *(RS_TS_20072)*

[SWS_EthTSyn_00248]{DRAFT} ⌈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.⌋*(RS_TS_20072)*

[SWS_EthTSyn_00249]{DRAFT} ⌈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.⌋*(RS_TS_20072)*

[SWS_EthTSyn_00250]{DRAFT} ⌈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]),
- call IdsM_SetSecurityEventWithContextData with the parameters EventId := SEV_TSYN_ETH_FRESHNESS_NOT_AVAILABLE (refer [SWS_EthTSyn_00261])

⌋(*RS_TS_20072*)

Note: Refer to chapter ICV Generation 7.7.3.5

Note: Refer to the chapter in [8] for the configuration details of CSM job used for ICV generation.

[SWS_EthTSyn_00251]{DRAFT} ⌈If EthTSynIcvGenerationBase for the Time Domain is configured to ICV_MAC, the Time Master shall call Csm_MacGenerate to generate the ICV value.⌋(*RS_TS_20072*)

[SWS_EthTSyn_00252]{DRAFT} ⌈If EthTSynIcvGenerationBase for the Time Domain is configured to ICV_SIGNATURE, the Time Master shall call Csm_SignatureGenerate to generate the ICV value.⌋(*RS_TS_20072*)

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]{DRAFT} ⌈If the CSM job used to generate ICV is configured in synchronous behaviour, the Time Master shall disable ICV generation timeout monitoring.⌋(*RS_TS_20072*)

[SWS_EthTSyn_00254]{DRAFT} ⌈If Csm_MacGenerate or Csm_SignatureGenerate returns E_OK, the Time Master shall start the EthTSynIcvGenerationTimeout.⌋(*RS_TS_20072*)

[SWS_EthTSyn_00255]{DRAFT} ⌈When the EthTSyn_IcvGenerationIndication callback is called, the Time Master shall stop the running ICV generation timeout timer (EthTSynIcvGenerationTimeout).⌋(*RS_TS_20072*)

[SWS_EthTSyn_00256]{DRAFT} ⌈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,
- call `IdsM_SetSecurityEventWithContextData` with the parameters EventId := `SEV_TSYN_ETH_ICV_GENERATION_FAILED` (refer to [SWS_EthTSyn_00261])

⌋(*RS_TS_20072*)

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

[SWS_EthTSyn_00257]{DRAFT} ⌈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.⌋(*RS_TS_20072*)

[SWS_EthTSyn_00258]{DRAFT} ⌈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.⌋(*RS_TS_20072*)

[SWS_EthTSyn_00402]{DRAFT} ⌈For every transmission of messages that contain the AUTOSAR Sub-TLV: Time Authenticated, EthTSyn shall maintain an authentication build counter (refer `EthTSynTxAuthenticationBuildAttempts`).⌋()

[SWS_EthTSyn_00403]{DRAFT} ⌈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]{DRAFT} ⌈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]{DRAFT} ⌈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]{DRAFT} ⌈If building the authenticated message generation has failed and the authentication build counter has not yet reached the configuration value `EthTSynTxAuthenticationBuildAttempts`, the freshness attempt and ICV calculation shall be retried in the next call of the `EthTSyn_MainFunction`.⌋()

7.7.2 Link State and Transmission Mode

[SWS_EthTSyn_00019] ⌈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.⌋(*RS_TS_20048, RS_TS_20051*)

[SWS_EthTSyn_00020] ⌈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.⌋*(RS_TS_20048, RS_TS_20051)*

[SWS_EthTSyn_00021] ⌈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.⌋*(RS_TS_20048, RS_TS_20051)*

[SWS_EthTSyn_00022] ⌈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.⌋*(RS_TS_20048, RS_TS_20051)*

7.7.3 Message Field Calculation and Assembling

Refer to chapter Message Field Calculation and Assembling in [1] for additional requirements.

7.7.3.1 SGW Calculation

Refer to chapter SGW Calculation in [1].

7.7.3.2 OFS Calculation

[SWS_EthTSyn_00199] ⌈The Time Master shall get the Offset Time Base value from the StbM via `StbM_GetOffset`.⌋*(RS_TS_20063)*

Refer to chapter OFS Calculation in [1] for additional requirements.

7.7.3.3 CRC Calculation

Refer to chapter CRC Calculation in [1] for additional requirements.

[SWS_EthTSyn_00096] ⌈The function `Crc_CalculateCRC8H2F` as defined in [9] shall be used to calculate the CRC if configured.⌋*(RS_TS_20061)*

7.7.3.3.1 AUTOSAR Sub-TLV: Time Secured

Refer to chapter AUTOSAR Sub-TLV: Time Secured in [1].

7.7.3.3.2 AUTOSAR Sub-TLV: Status secured

Refer to chapter AUTOSAR Sub-TLV: Status secured in [1].

7.7.3.3.3 AUTOSAR Sub-TLV: UserData secured

Refer to chapter AUTOSAR Sub-TLV: UserData secured in [1].

7.7.3.3.4 AUTOSAR Sub-TLV: OFS secured

Refer to chapter AUTOSAR Sub-TLV: OFS secured in [1].

7.7.3.4 Sequence Counter (sequenceId) Calculation

Refer to chapter Sequence Counter (sequenceId) Calculation in [1] for additional requirements.

7.7.3.5 ICV Generation

Refer to chapter ICV Generation in [1].

7.7.3.6 Message Assembling

[SWS_EthTSyn_00104]{DRAFT} ⌈Refer to chapter Message Assembling in [1].⌋*(RS_TS_20048, RS_TS_20072)*

7.7.3.7 Dynamic port configuration for Time Master and Time Slave

[SWS_EthTSyn_00414]{DRAFT} ⌈If the parameter `EthTSynGlobalTimePortRole` 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]{DRAFT} ⌈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]{DRAFT} ⌈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 Acting as Time Slave

Refer to chapter Acting as Time Slave in [1] for additional requirements.

7.8.1 Message processing

In addition to the Follow_Up message fields:

- `preciseOriginTimeStamp`
- `correctionField`

(refer to [1] 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 diagrams Figure 9.5, and Figure 9.6.

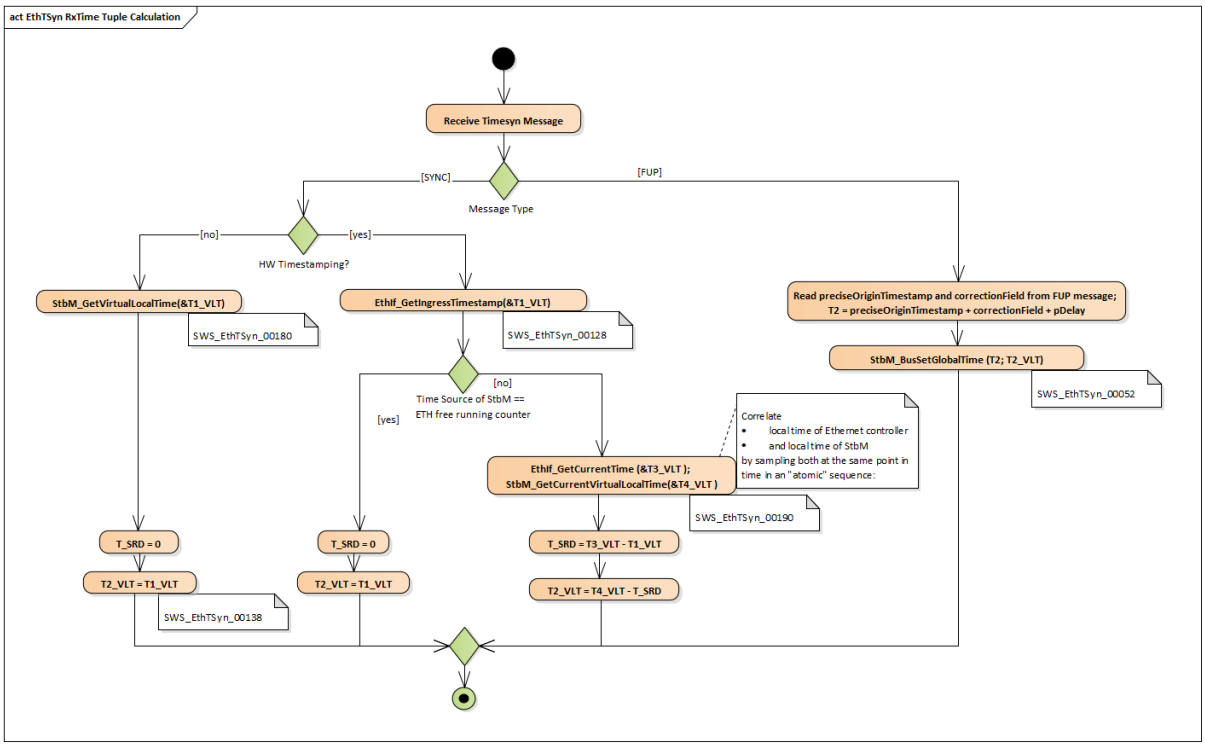

Figure 7.2: Evaluate Timesync message

[SWS_EthTSyn_00412] ⌈If `EthTSynGlobalTimePortRole` 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] ⌈If `EthTSynGlobalTimePortRole` 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] ⌈On invocation of `EthTSyn_RxIndication` and if `EthTSynHardwareTimestampSupport` is set to TRUE, the ingress time stamp for the Sync message, T1VLT, shall be retrieved via `EthIf_GetIngressTimeStamp`.⌋*(RS_TS_20048)*

[SWS_EthTSyn_00138] ⌈On invocation of `EthTSyn_RxIndication` for the Sync message and 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 $T2_{VLT}$ 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

⌋(*RS_TS_20048*)

[SWS_EthTSyn_00180] ⌈On invocation of `EthTSyn_RxIndication` and if `EthTSynHardwareTimestampSupport` 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_GetCurrentVirtualLocalTime` 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.

⌋(*RS_TS_20048*)

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] ⌈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 $T4_{VLT} - T_{SRD}$.

⌋(*RS_TS_20048*)

[SWS_EthTSyn_00052] ⌈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`

⌋*(RS_TS_20048)*

Note: The Pdelay value is not influenced significantly by a RateRatio acc to [4] Note-2 of chapter "computePropTime():".

[SWS_EthTSyn_00266]{DRAFT} ⌈If

- configuration parameter `EthTSynRateRatioEnable` is set to `TRUE`
- 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].⌋*(RS_TS_20048)*

[SWS_EthTSyn_00267]{DRAFT} ⌈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`.

⌋*(RS_TS_20048)*

Note: According to [4, IEEE 802.1 AS] 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]{DRAFT} ⌈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.

⌋(*RS_TS_20048*)

[SWS_EthTSyn_00150] ⌈When calling StbM_BusSetGlobalTime, EthTSyn shall pass the current linkDelay value (refer [SWS_EthTSyn_00264]) by the parameter measureDataPtr->pathDelay to the StbM.⌋(*RS_TS_20058*)

[SWS_EthTSyn_00129] ⌈When providing a new Global Time tuple to the StbM via StbM_BusSetGlobalTime, EthTSyn shall set the SYNC_TO_GATEWAY bit in time BaseStatus (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.⌋(*RS_TS_20051*)

[SWS_EthTSyn_00230] ⌈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.⌋(*RS_TS_20062*)

[SWS_EthTSyn_00227] ⌈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⌋(*RS_TS_20069*)

7.8.1.1 Runtime Error detection

[SWS_EthTSyn_00146] ⌈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.⌋(*RS_TS_20051*)

7.8.1.2 Frame Debouncing

Refer to chapter Frame Debouncing in PRS-TimeSyncProtocol [1] for additional requirements.

[SWS_EthTSyn_00232]{DRAFT} ⌈. During the EthTSynGlobalTimeRxDebounceTime, if the sequence is reset, then the Time Slave shall call IdsM_SetSecurityEventWithContextData with the parameters EventId := SEV_TSYN_ETH_MSG_SEQUENCE_ERROR (refer to [SWS_EthTSyn_00261])⌋*(RS_TS_20072)*

7.8.1.3 Secure Time Synchronization

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

[SWS_EthTSyn_00233]{DRAFT} ⌈When the FV is referenced (refer EthTSynIcvVerificationFvIdRef), FVL is greater than 0 and 'ICV with FV' bit is set in ICV_Flags of received Follow_Up message, 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

⌋*(RS_TS_20072)*

[SWS_EthTSyn_00234]{DRAFT} ⌈If StbM_GetRxFreshness returns E_OK, the Time Slave shall use the FV in ICV verification.⌋*(RS_TS_20072)*

[SWS_EthTSyn_00236]{DRAFT} ⌈If StbM_GetRxFreshness returns non-recoverable error code (e.g., E_NOT_OK) or FVL == 0 and ICV with FV bit is set in ICV_Flags of received Follow_Up message, 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]),
- call IdsM_SetSecurityEventWithContextData with the parameters EventId := SEV_TSYN_ETH_FRESHNESS_NOT_AVAILABLE (refer to [SWS_EthTSyn_00261])

⌋(*RS_TS_20072*)

Refer to the chapter in [8] for the configuration details of CSM job used for ICV verification.

[SWS_EthTSyn_00237]{DRAFT} ⌈If `EthTSynIcvVerificationBase` for the Time Domain is configured to `ICV_MAC`, the Time Slave shall call `Csm_MacVerify` to verify the `ICV` value.⌋(*RS_TS_20072*)

[SWS_EthTSyn_00238]{DRAFT} ⌈If `EthTSynIcvVerificationBase` for the Time Domain is configured to `ICV_SIGNATURE`, the Time Slave shall call `Csm_SignatureVerify` to verify the `ICV` value.⌋(*RS_TS_20072*)

Note: 7.8.2.5

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]{DRAFT} ⌈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.⌋(*RS_TS_20072*)

[SWS_EthTSyn_00240]{DRAFT} ⌈If `Csm_MacVerify` or `Csm_SignatureVerify` returns `E_OK`, the Time Slave shall start the `EthTSynIcvVerificationTimeout`.⌋(*RS_TS_20072*)

[SWS_EthTSyn_00241]{DRAFT} ⌈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.⌋(*RS_TS_20072*)

[SWS_EthTSyn_00242]{DRAFT} ⌈The `EthTSynIcvVerificationTimeout` shall be stopped with the notification of the `EthTSyn_IcvVerificationIndication` callback.⌋(*RS_TS_20072*)

[SWS_EthTSyn_00243]{DRAFT} ⌈If one of the following conditions is true:

- the authentication build counter has reached the configuration value `EthTSynRxAuthenticationBuildAttempts`,
- 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,
- call `IdsM_SetSecurityEventWithContextData` with the parameters EventId := `SEV_TSYN_ETH_ICV_VERIFICATION_FAILED` (refer to [SWS_EthTSyn_00261])

⌋*(RS_TS_20072)*

Note: 7.8.2.5

[SWS_EthTSyn_00407]{DRAFT} ⌈For every reception of messages that require ICV verification, EthTSyn shall maintain an authentication build counter (refer `EthTSynRxAuthenticationBuildAttempts`).⌋*()*

[SWS_EthTSyn_00408]{DRAFT} ⌈Upon the initial processing of messages that require ICV verification, the authentication build counter shall be set to 0.⌋*()*

[SWS_EthTSyn_00409]{DRAFT} ⌈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]{DRAFT} ⌈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]{DRAFT} ⌈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]{DRAFT} ⌈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.⌋*(RS_TS_20072)*

[SWS_EthTSyn_00245]{DRAFT} ⌈When the `EthTSyn_IcvVerificationIndication` callback is called and ICV verification result is unsuccessful, the Time Slave shall discard the `Follow_Up` message.⌋*(RS_TS_20072)*

7.8.2 Message Field Validation and Disassembling

Additional content to this chapter can be found in [1] in chapter Message Field Validation and Disassembling.

7.8.2.1 SGW Calculation

Refer to chapter SGW Calculation in [1].

7.8.2.2 OFS Calculation

[SWS_EthTSyn_00198] ⌈The Time Slave shall forward the new Offset Time to the StbM via `StbM_BusSetGlobalTime` (as calculated according to [PRS_TS_00110]), if successfully validated.⌋*(RS_TS_20063)*

Refer to chapter OFS Calculation in [1] for additional requirements.

7.8.2.3 CRC Validation

[SWS_EthTSyn_00111] ⌈The function `Crc_CalculateCRC8H2F` as defined in [9] shall be used to calculate the CRC if configured.⌋*(RS_TS_20061)*

Refer to chapter CRC Calculation in [1] for additional requirements.

7.8.2.3.1 AUTOSAR Sub-TLV: Time Secured

Refer to chapter AUTOSAR Sub-TLV: Time Secured in [1].

7.8.2.3.2 AUTOSAR Sub-TLV: Status secured

Refer to chapter AUTOSAR Sub-TLV: Status secured in [1].

7.8.2.3.3 AUTOSAR Sub-TLV: UserData secured

Refer to chapter AUTOSAR Sub-TLV: UserData secured in [1].

7.8.2.3.4 AUTOSAR Sub-TLV: OFS secured

Refer to chapter AUTOSAR Sub-TLV: OFS secured in [1].

7.8.2.4 Sequence Counter (sequenceId) Validation

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

7.8.2.5 ICV Verification

Refer to chapter ICV Verification in PRS-TimeSyncProtocol [1].

7.8.2.6 Message Disassembling

Refer to chapter Message Disassembling in [1].

7.9 Time Recording

7.9.1 Time Validation

[SWS_EthTSyn_00212] ⌈The `EthTSyn` shall support Time Validation, if `EthTSynTimeValidationSupport` set to `TRUE`.⌋*(RS_TS_00034)*

[SWS_EthTSyn_00213] ⌈If

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

EthTSyn shall do time recording for Time Validation for that Time Domain⌋*(RS_TS_00034)*

[SWS_EthTSyn_00214] ⌈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`)

⌋*(RS_TS_20048)*

Note: `EthTSyn TimesyncSequence` is shown in Figure 9.2

[SWS_EthTSyn_00215] ⌈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.⌋(*RS_TS_20048*)

[SWS_EthTSyn_00216] ⌈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.⌋(*RS_TS_00034*)

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] ⌈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.⌋(*RS_TS_00034*)

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]).

7.9.1.1 Recording of Pdelay Measurement

[SWS_EthTSyn_00218] ⌈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).⌋*(RS_TS_00034)*

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] ⌈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).⌋*(RS_TS_00034)*

Note: EthTSyn PdelaySequence is shown in Figure 9.3

[SWS_EthTSyn_00220] ⌈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 [$T_{refPDResponder}$; $T_{VLT\_refPDResponder}$] (refer to [SWS_EthTSyn_00218])

to StbM_EthSetPdelayResponderData upon invocation by the parameter measure DataPtr.⌋*(RS_TS_00034)*

[SWS_EthTSyn_00223] ⌈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 [$T_{refPDInitiator}$; $T_{VLT\_refPDInitiator}$] before sending the pDelay_Req message (refer to EthTSyn PdelaySequence).⌋*(RS_TS_00034)*

Note: The Time Tuple Tuple [$T_{refPDInitiator}$; $T_{VLT\_refPDInitiator}$] 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] ⌈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).⌋*(RS_TS_00034)*

Note: EthTSyn PdelaySequence is shown in Figure 9.3

[SWS_EthTSyn_00222] ⌈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 [$T_{refPDInitiator}$; $T_{VLT\_refPDInitiator}$] (refer [SWS_EthTSyn_00223])

to StbM_EthSetPdelayInitiatorData upon invocation by the parameter measureDataPtr.⌋*(RS_TS_00034)*

7.10 Security Events

[SWS_EthTSyn_00231]{DRAFT} ⌈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 [10] via the interfaces defined in BSWGeneral [3].⌋*(RS_Ids_00810)*

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 ⌈

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 or OFS and OFNS from the TimeMaster within (CanTSyn GlobalTimeFollowUpTimeout).	76

⌋(*RS_Ids_00810*)

The following table describes the context data which shall be reported for the respective security events:

[SWS_EthTSyn_00262]{DRAFT} **Context data of respective Security events of Eth TSyn** ⌈

Security Event	*Context Data*
SEV_TSYN_ETH_ICV_GENERATION_FAILED	Context Data (1 Byte) - GlobalTimeDomainId
SEV_TSYN_ETH_ICV_VERIFICATION_FAILED	Context Data (1 Byte) - GlobalTimeDomainId
SEV_TSYN_ETH_FRESHNESS_NOT_AVAILABLE	Context Data (1 Byte) - GlobalTimeDomainId
SEV_TSYN_ETH_MSG_SEQUENCE_ERROR	Context Data (1 Byte) - GlobalTimeDomainId

⌋(*RS_Ids_00810*)

7.11 Error Classification

Section 7.x "Error Handling" of the document "General Specification of Basic Software Modules" describes the error handling of the Basic Software in detail. Above all, it constitutes a classification scheme consisting of five error types which may occur in BSW modules.

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

[SWS_EthTSyn_00029] ⌈On errors and exceptions, the `EthTSyn` module shall not modify its current module state but shall simply report the error event.⌋(*RS_TS_20051, SRS_BSW_00323*)

7.11.1 Development Errors

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

[SWS_EthTSyn_00030] Definiton of development errors in module EthTSyn ⌈

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
API called with invalid parameter	ETHTSYN_E_PARAM	0x24

⌋(*SRS_BSW_00337*, *SRS_BSW_00385*, *SRS_BSW_00323*, *SRS_BSW_00406*)

7.11.2 Runtime Errors

[SWS_EthTSyn_00144] Definiton of runtime errors in module EthTSyn ⌈

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

⌋(*SRS_BSW_00385*)

7.11.3 Transient Faults

No Transient Faults defined.

7.11.4 Production Errors

No Production Errors defined.

7.11.5 Extended Production Errors

No Extended Production Errors defined.

8 API specification

8.1 API

8.1.1 Imported types

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

[SWS_EthTSyn_00031] Definition of imported datatypes of module EthTSyn ⌈

Module	*Header File*	*Imported Type*
ComStack_Types	ComStack_Types.h	BufReq_ReturnType
Csm	Rte_Csm_Type.h	Crypto_OperationModeType
	Rte_Csm_Type.h	Crypto_ResultType
	Rte_Csm_Type.h	Crypto_VerifyResultType
Eth	Eth.h	Eth_RateDeviationStatusType (draft)
	Eth.h	Eth_RateDeviationType (draft)
	Eth_GeneralTypes.h	Eth_BufIdxType
	Eth_GeneralTypes.h	Eth_DataType
	Eth_GeneralTypes.h	Eth_FrameType
	Eth_GeneralTypes.h	Eth_TimeStampQualType (obsolete)
	Eth_GeneralTypes.h	Eth_TimeStampType (obsolete)
EthSwt	Eth_GeneralTypes.h	EthSwt_MgmtInfoType
EthTrcv	Eth_GeneralTypes.h	EthTrcv_LinkStateType
IdsM	IdsM_Types.h	IdsM_SecurityEventIdType
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
	Rte_StbM_Type.h	StbM_UserDataType
	StbM.h	StbM_MeasurementType
	StbM.h	StbM_VirtualLocalTimeType
Std	Std_Types.h	Std_ReturnType
	Std_Types.h	Std_VersionInfoType

⌋(*RS_TS_20048*, *RS_TS_20059*)

8.1.2 Type definitions

8.1.2.1 EthTSyn_ConfigType

[SWS_EthTSyn_00032] Definition of datatype EthTSyn_ConfigType ⌈

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	

⌋(*RS_TS_20048*)

8.1.2.2 EthTSyn_TransmissionModeType

[SWS_EthTSyn_00033] Definition of datatype EthTSyn_TransmissionModeType ⌈

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		

⌋(*RS_TS_20048*)

8.1.3 Function definitions

8.1.3.1 EthTSyn_Init

[SWS_EthTSyn_00035] Definition of API function EthTSyn_Init ⌈

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	

⌋*(RS_TS_20048)* See section 7.1.1 for details.

8.1.3.2 EthTSyn_GetVersionInfo

[SWS_EthTSyn_00036] Definition of API function EthTSyn_GetVersionInfo ⌈

Service Name	EthTSyn_GetVersionInfo	

Syntax	`void EthTSyn_GetVersionInfo (`
	`  Std_VersionInfoType* versioninfo`
	`)`	

Service ID [hex]	0x02	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	versioninfo	Pointer to where to store the version information of this module.
Return value	None	
Description	Returns the version information of this module.	
Available via	EthTSyn.h	

⌋*(RS_TS_20048)*

8.1.3.3 EthTSyn_SetTransmissionMode

[SWS_EthTSyn_00039] Definition of API function EthTSyn_SetTransmissionMode ⌈

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

⌋*(RS_TS_20048)*

[SWS_EthTSyn_00172] ⌈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)

⌋*(SRS_BSW_00323, SRS_BSW_00337, SRS_BSW_00386)*

8.1.3.4 EthTSyn_SetProtocolParam

[SWS_EthTSyn_00330] Definition of API function EthTSyn_SetProtocolParam ⌈

Service Name	EthTSyn_SetProtocolParam	
Syntax	`Std_ReturnType EthTSyn_SetProtocolParam ( StbM_SynchronizedTimeBaseType timeBaseId, const StbM_ProtocolParamType* protocolParam )`	
Service ID [hex]	0xa	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
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] ⌈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)

⌋(*SRS_BSW_00323*, *SRS_BSW_00337*, *SRS_BSW_00386*)

8.1.3.5 EthTSyn_GetProtocolParam

[SWS_EthTSyn_00331] Definition of API function EthTSyn_GetProtocolParam ⌈

Service Name	EthTSyn_GetProtocolParam	
Syntax	`Std_ReturnType EthTSyn_GetProtocolParam ( StbM_SynchronizedTimeBaseType timeBaseId, StbM_ProtocolParamType* protocolParam )`	
Service ID [hex]	0xb	
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] ⌈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)

⌋(*SRS_BSW_00323*, *SRS_BSW_00337*, *SRS_BSW_00386*)

8.1.4 Call-back notifications

This is a list of functions provided for other modules.

8.1.4.1 EthTSyn_RxIndication

[SWS_EthTSyn_00040] Definition of API function EthTSyn_RxIndication ⌈

Service Name	EthTSyn_RxIndication	
Syntax	`void EthTSyn_RxIndication ( uint8 CtrlIdx, Eth_FrameType FrameType, boolean IsBroadcast, const uint8* PhysAddrPtr, const uint8* DataPtr, uint16 LenByte )`	
Service ID [hex]	0x06	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	CtrlIdx	Index of the Ethernet controller
	FrameType	frame type of received Ethernet frame
	IsBroadcast	Parameter to indicate a broadcast frame which can be ignored as gPTP works over Multicast domain
	PhysAddrPtr	pointer to Physical source address (MAC address in network byte order) of received Ethernet frame
	DataPtr	Pointer to payload of the received Ethernet frame (i.e. Ethernet header is not provided).
	LenByte	Length of received data.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	By this API service the EthTSyn gets an indication and the data of a received frame.	
Available via	EthTSyn.h	

⌋(*RS_TS_20048*)

[SWS_EthTSyn_00041] ⌈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:

- CtrlIdx is invalid (ETHTSYN_E_CTRL_IDX)
- DataPtr or PhysAddrPtr is invalid (ETHTSYN_E_PARAM_POINTER)

⌋(*SRS_BSW_00337*, *SRS_BSW_00323*)

8.1.4.2 EthTSyn_TxConfirmation

[SWS_EthTSyn_00042] Definition of API function EthTSyn_TxConfirmation ⌈

Service Name	EthTSyn_TxConfirmation	
Syntax	`void EthTSyn_TxConfirmation ( uint8 CtrlIdx, Eth_BufIdxType BufIdx, Std_ReturnType Result )`	
Service ID [hex]	0x07	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant Dont care	
Parameters (in)	CtrlIdx	Index of the Ethernet controller within the context of the Ethernet Interface
	BufIdx	Index of the buffer resource
	Result	E_OK: The transmission was successful, E_NOT_OK: The transmission failed.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Confirms the transmission of an Ethernet frame	
Available via	EthTSyn.h	

⌋*(RS_TS_20048)*

[SWS_EthTSyn_00175] ⌈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:

- CtrlIdx is invalid (ETHTSYN_E_CTRL_IDX)

⌋*(SRS_BSW_00323, SRS_BSW_00337)*

[SWS_EthTSyn_00176] ⌈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.⌋ *(SRS_BSW_00323, SRS_BSW_00337)*

8.1.4.3 EthTSyn_TrcvLinkStateChg

[SWS_EthTSyn_00043] Definition of callback function EthTSyn_TrcvLinkStateChg ⌈

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	

⌋(*RS_TS_20048*)

[SWS_EthTSyn_00174] ⌈The function EthTSyn_TrcvLinkStateChg() 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)

⌋(*SRS_BSW_00323*, *SRS_BSW_00337*)

8.1.4.4 EthTSyn_IcvGenerationIndication

[SWS_EthTSyn_91001]{DRAFT} **Definition of API function EthTSyn_IcvGenerationIndication** ⌈

Service Name	EthTSyn_IcvGenerationIndication (draft)	
Syntax	`void EthTSyn_IcvGenerationIndication ( uint32 jobId, Crypto_ResultType result )`	
Service ID [hex]	0xc	
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. **Tags:** atp.Status=draft
Available via	EthTSyn.h

⌋(*RS_TS_20072*)

[SWS_EthTSyn_00259]{DRAFT} ⌈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)

⌋(*SRS_BSW_00323*, *SRS_BSW_00337*)

8.1.4.5 EthTSyn_IcvVerificationIndication

[SWS_EthTSyn_91002]{DRAFT} **Definition of API function EthTSyn_IcvVerificationIndication** ⌈

Service Name	EthTSyn_IcvVerificationIndication (draft)	
Syntax	`void EthTSyn_IcvVerificationIndication ( uint32 jobId, Crypto_ResultType result )`	
Service ID [hex]	0xd	
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. **Tags:** atp.Status=draft	
Available via	EthTSyn.h	

⌋(*RS_TS_20072*)

[SWS_EthTSyn_00260]{DRAFT} ⌈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)

⌋(*SRS_BSW_00323*, *SRS_BSW_00337*)

8.1.5 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.

8.1.5.1 EthTSyn_MainFunction

[SWS_EthTSyn_00044] Definition of scheduled function EthTSyn_MainFunction ⌈

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

⌋(*RS_TS_20048*)

[SWS_EthTSyn_00045] ⌈The frequency of invocations of `EthTSyn_MainFunction` is determined by the configuration parameter EthTSynMainFunctionPeriod (ECUC_EthTSyn_00012 :).⌋(*RS_TS_20048*)

8.1.6 Expected Interfaces

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

8.1.6.1 Mandatory Interfaces

There are no mandatory interfaces defined.

8.1.6.2 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 in module EthTSyn ⌈

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 MAC by comparing if the signature is 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_GetCurrentTime (obsolete)	EthIf.h	Returns a time value out of the HW registers according to the capability of the HW. Is the HW resolution is lower than the Eth_TimeStampType resolution resp. range, the remaining bits will be filled with 0. Important Note: EthIf_GetCurrentTime may be called within an exclusive area. **Tags:** atp.Status=obsolete
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_GetIngressTimeStamp	EthIf.h	Reads back the ingress time stamp on a dedicated message object. It must be called within the Rx Indication() function.
EthIf_ProvideTxBuffer	EthIf.h	Provides access to a transmit buffer of the specified Ethernet controller.
EthIf_SetSwitchMgmtInfo	EthIf.h	Provides additional management information along to an Ethernet frame that requires special treatment within the Switch. It has to be called between EthIf_ ProvideTxBuffer() and EthIf_Transmit() of the related frame.
EthIf_SwitchEnableTimeStamping	EthIf.h	Activates egress time stamping on a dedicated message object, addressed by CtrlIdx and BufIdx.
EthIf_Transmit	EthIf.h	Triggers transmission of a previously filled transmit buffer
IdsM_SetSecurityEventWithContext Data	IdsM.h	This API is the application interface to report security events with context data to the IdsM.
StbM_BusSetGlobalTime	StbM.h	Allows the Time Base Provider Modules to forward the Rx Time Tuple to the StbM.
StbM_EthSetMasterTimingData (draft)	StbM_EthTSyn.h	Provides Ethernet Timesyn module specific data for a Time Master to the StbM. **Tags:** atp.Status=draft

▽

△

API Function	*Header File*	*Description*
StbM_EthSetPdelayInitiatorData (draft)	StbM_EthTSyn.h	– **Tags:** atp.Status=draft
StbM_EthSetPdelayResponderData (draft)	StbM_EthTSyn.h	– **Tags:** atp.Status=draft
StbM_EthSetSlaveTimingData (draft)	StbM_EthTSyn.h	Allows the EthTSyn Module to forward Ethernet specific details to the StbM. **Tags:** atp.Status=draft
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_GetOffset	StbM.h	Allows the Timesync Modules to get the current Offset Time and User Data.
StbM_GetRxFreshness (draft)	StbM.h	This interface is used by the StbM to query the current freshness value. **Tags:** atp.Status=draft
StbM_GetTimeBaseStatus	StbM.h	Returns detailed status information for a Synchronized (or Pure Local) Time Base and, if called for an Offset Time Base, for the Offset Time Base and the underlying Synchronized 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 (draft)	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. **Tags:** atp.Status=draft
StbM_GetTxFreshnessTruncData (draft)	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. **Tags:** atp.Status=draft
StbM_SPduTxConfirmation (draft)	StbM.h	This interface is used by the StbM to indicate that the Secured Time Synchronization Message has been initiated for transmission. **Tags:** atp.Status=draft

⌋(*RS_TS_20048*, *RS_TS_20059*)

9 Sequence diagrams

Note: Please consider, that all sequence diagrams are use case specific (Ethernet controller w/o Switch).

9.1 EthIf_EnableEgressTimeStamp

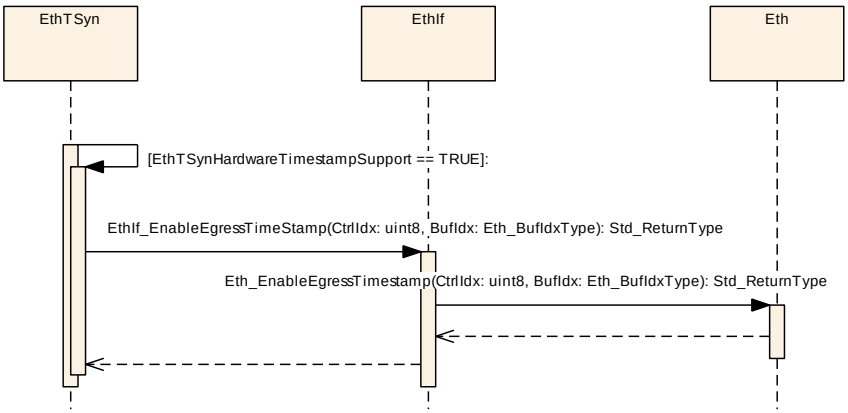

Figure 9.1: EnableEgressTimeStamp

9.2 Time Synchronization Sequence

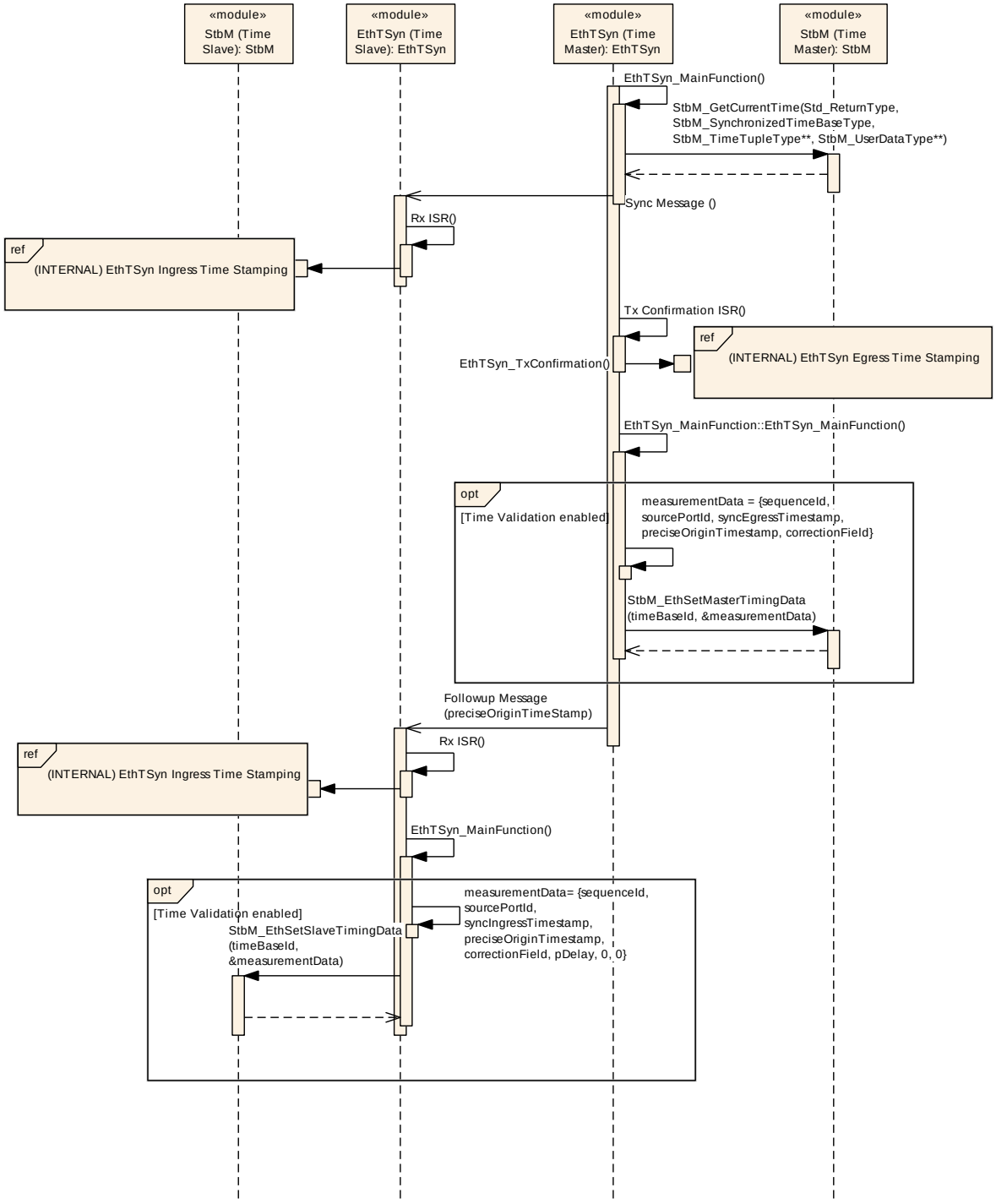

Figure 9.2: : Time Synchronization Sequence

9.3 Pdelay Measurement Sequence

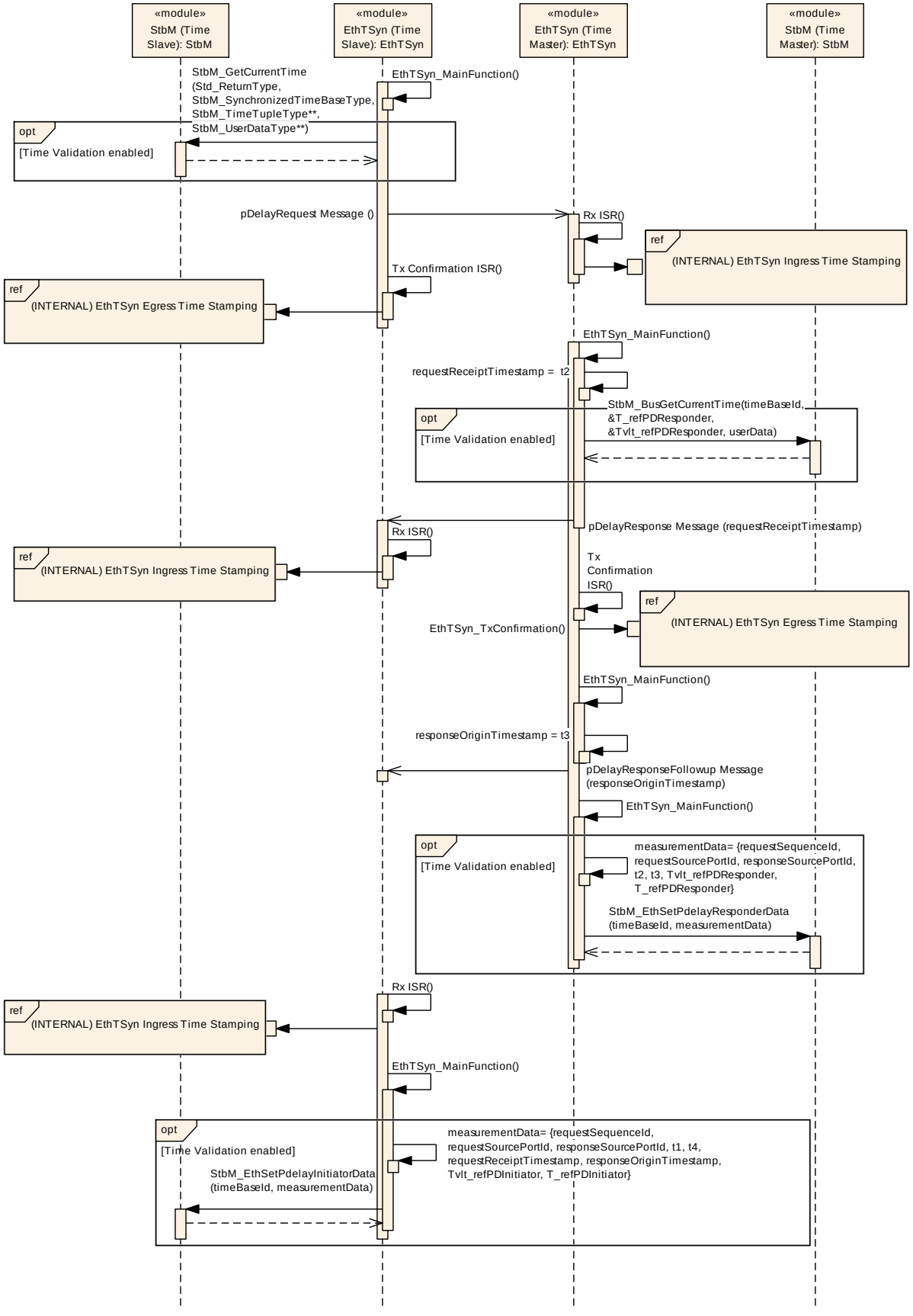

Figure 9.3: : Pdelay Sequence

9.4 EthTSyn Egress Timestamping

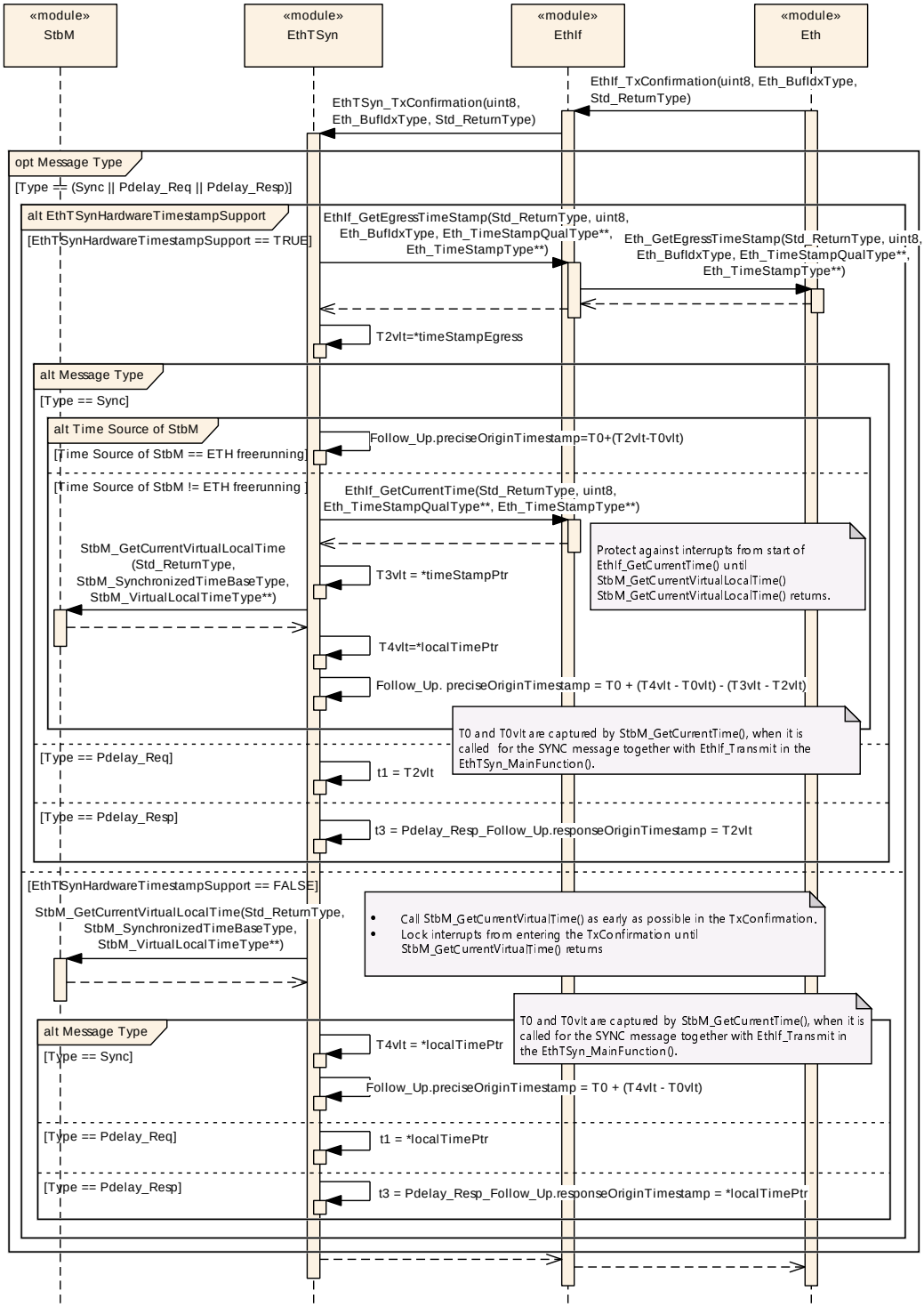

Figure 9.4: EthTSynEgressTimestamping

9.5 EthTSyn Ingress Timestamping

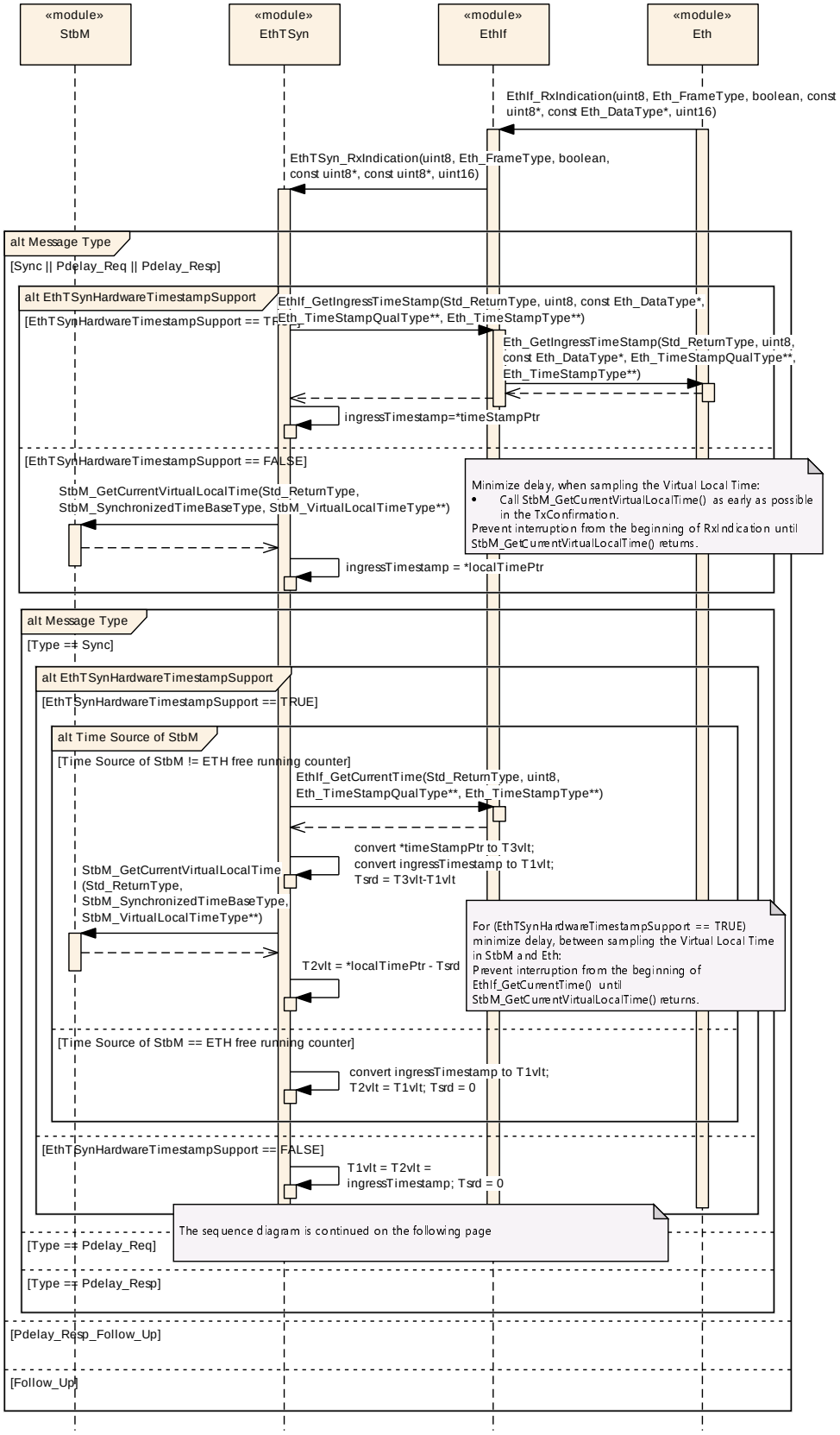

Figure 9.5: EthTSyn Ingress Timestamping. Continued on the next page

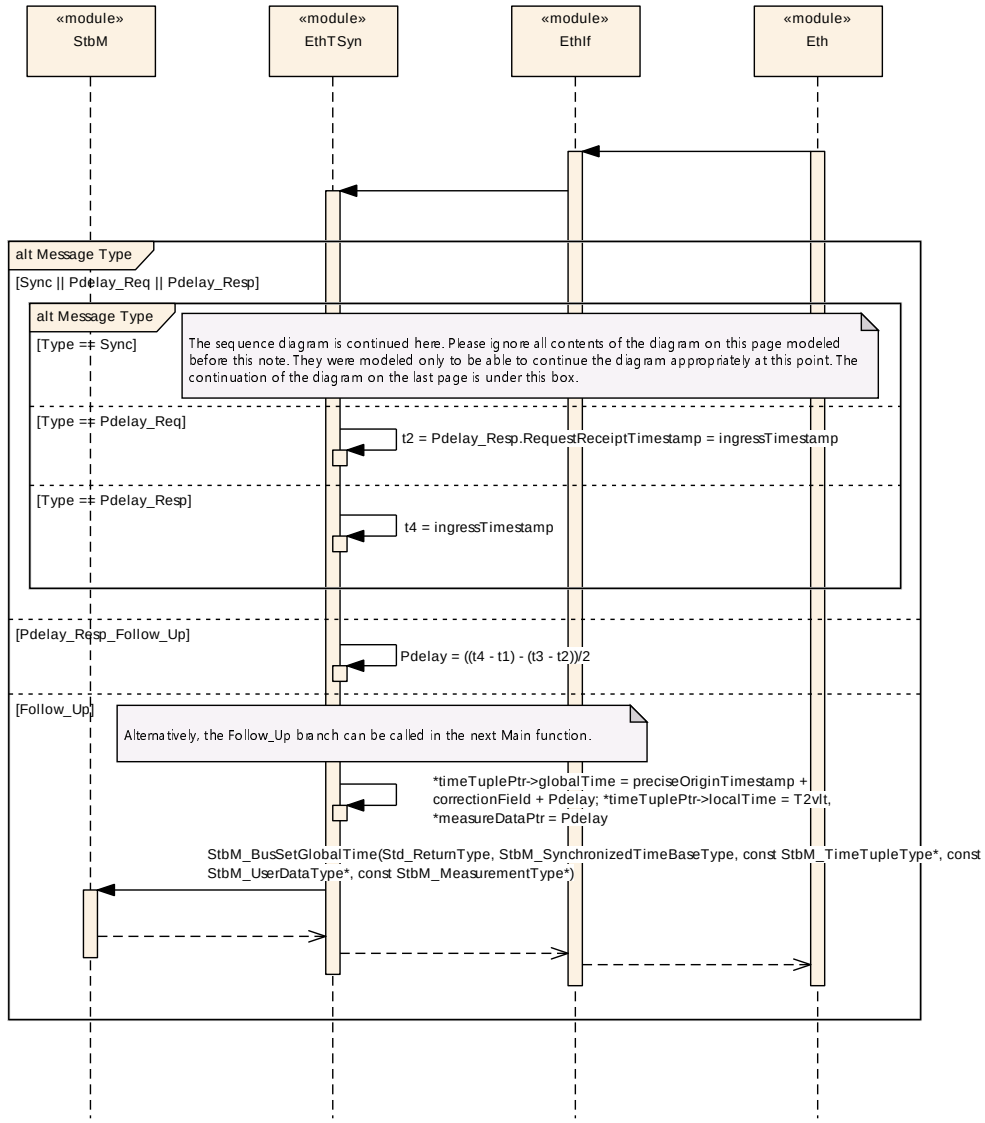

Figure 9.6: EthTSyn Ingress Timestamping

9.6 Time measurement with Switches

9.6.1 Time Aware Bridge with GTM as Management CPU - Tx

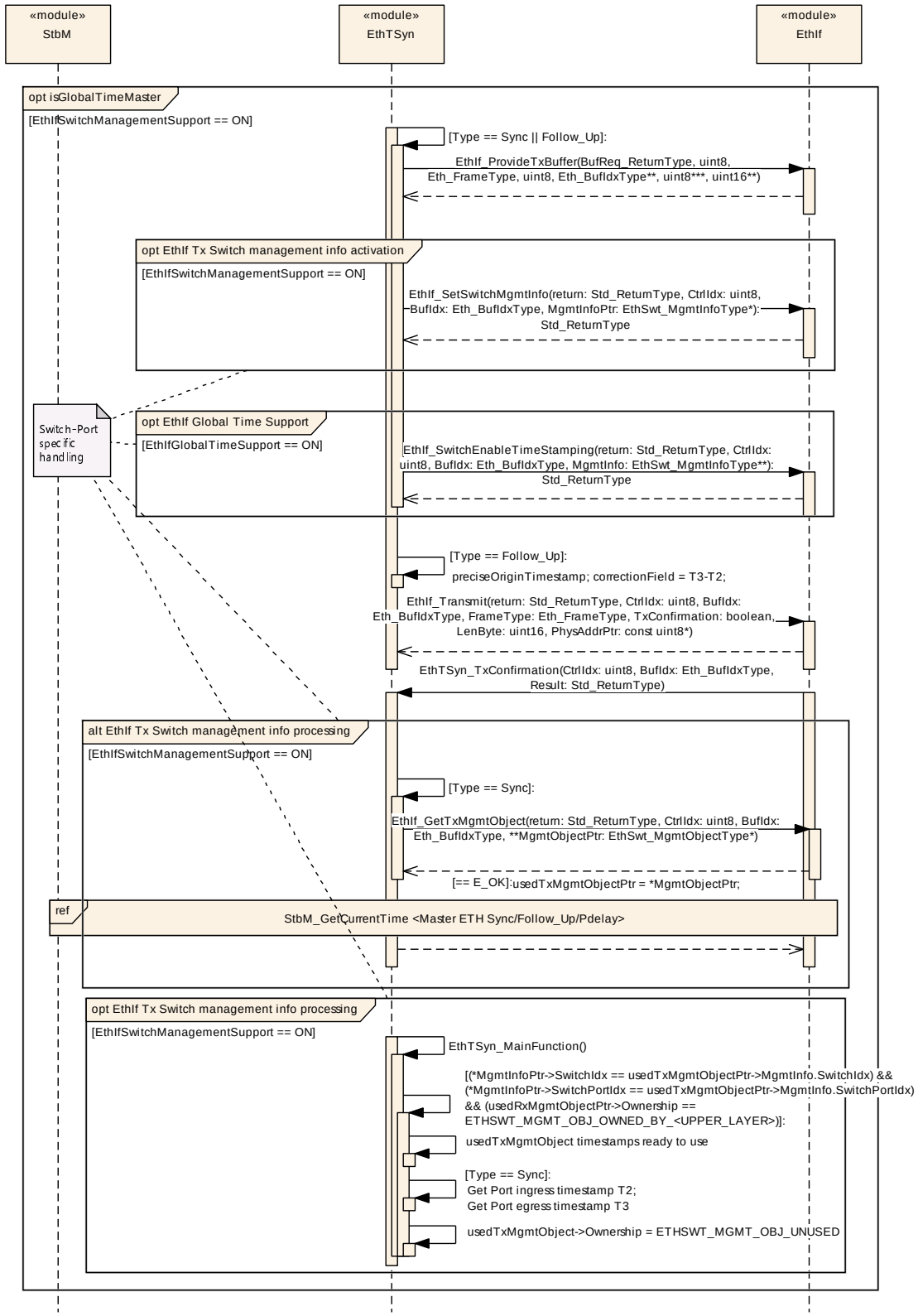

Figure 9.7: Time Aware Bridge with GTM as Management CPU Sync_Up Follow_Up Tx

9.6.2 Time Aware Bridge without GTM as Management CPU - Tx

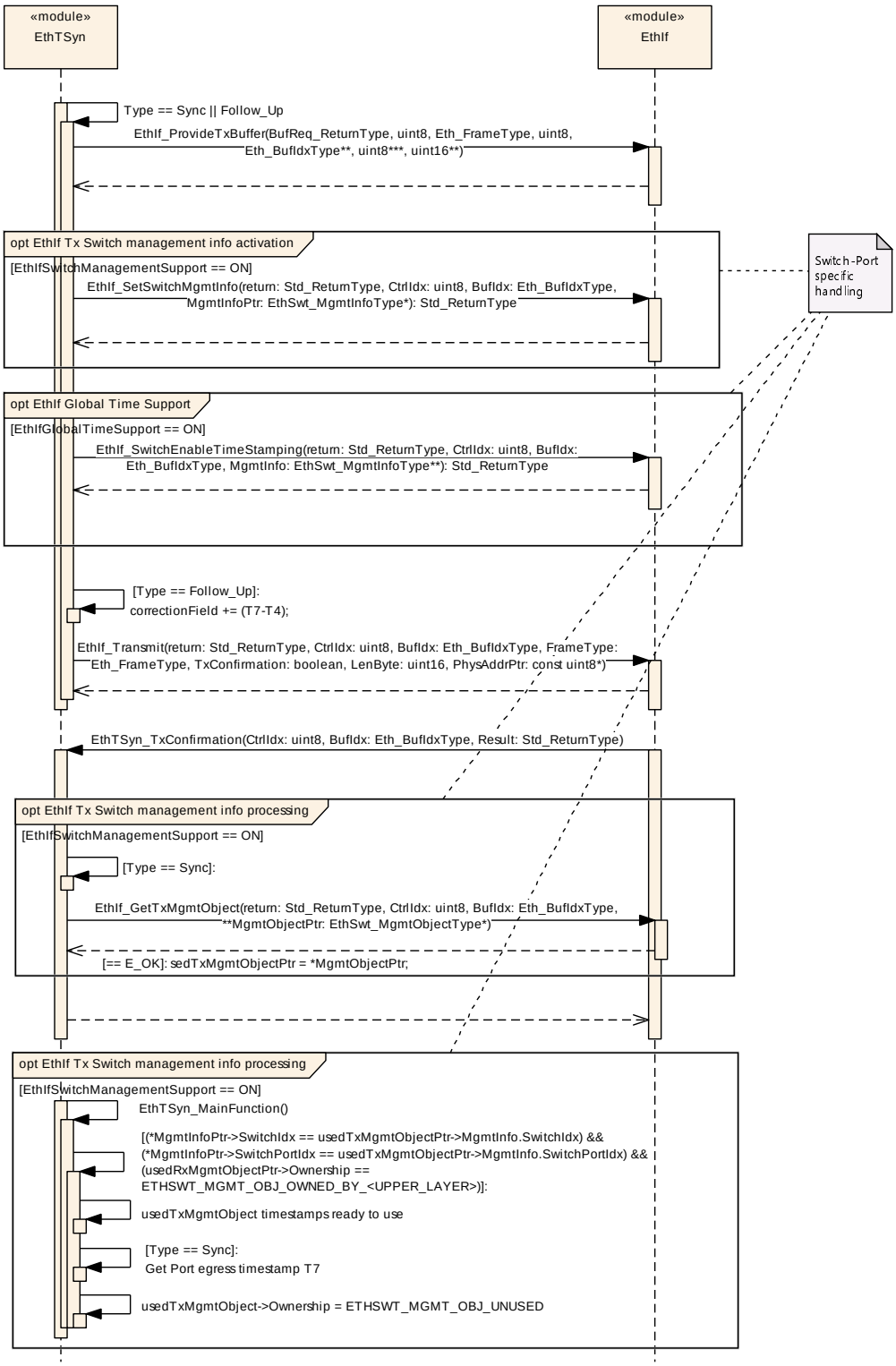

Figure 9.8: EthTSyn_SwitchWithoutGTM_Sync_Follow_Up_Tx

9.6.3 Time Aware Bridge without GTM as Management CPU - Rx

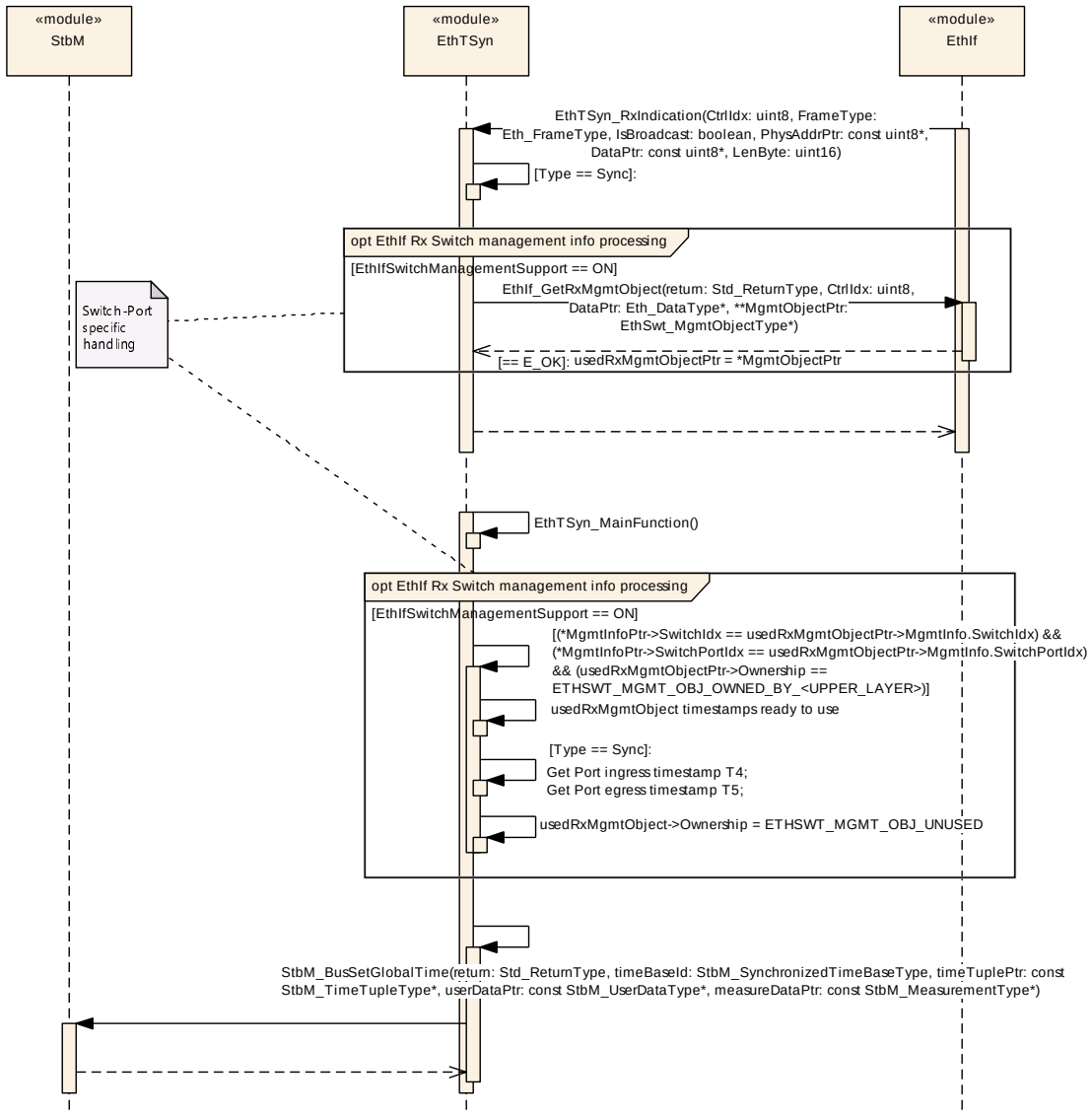

Figure 9.9: EthTSyn_SwitchWithoutGTM_Sync_Follow_Up_Rx

9.7 Secure Time Synchronization Sequence

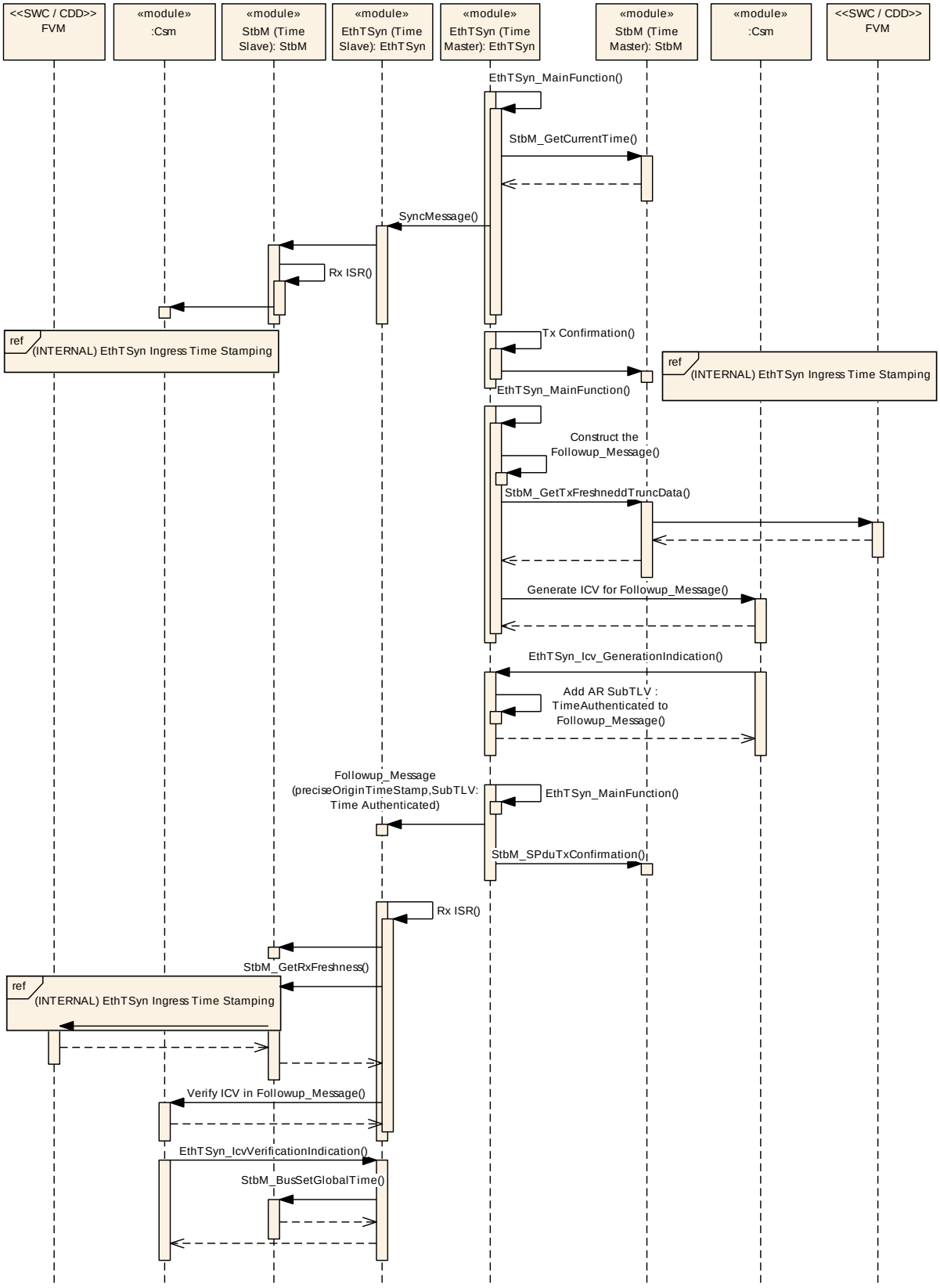

Figure 9.10: Secure Time Synchronization Sequence

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 10.4 specifies published information of the module EthTSyn.

10.1 How to read this chapter

For details refer to the chapter 10.1 "Introduction to configuration specification" in [3].

[SWS_EthTSyn_00051] ⌈The EthTSyn module shall support the configuration for Time Master, Time Slave and Time Gateway.⌋ *(RS_TS_20052, RS_TS_20053, RS_TS_20054)*

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

SWS Item	[ECUC_EthTSyn_00001]
Module Name	EthTSyn
Description	Configuration of the Synchronized Time-base Manager (StbM) module with respect to global time handling on Ethernet.
Post-Build Variant Support	true
Supported Config Variants	VARIANT-PRE-COMPILE

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthTSynGeneral	1	This container holds the general parameters of the Ethernet-specific Synchronized Time-base Manager
EthTSynGlobalTimeDomain	1..*	This represents the existence of a global time domain on Ethernet. The EthTSyn module can administrate several global time domains at the same time that in itself form a hierarchy of domains and sub-domains. If the EthTSyn exists it is assumed that at least one global time domain exists.

EthSyn is shown in the Figure Figure 5.1

10.2.2 EthTSynGeneral

SWS Item	[ECUC_EthTSyn_00003]
Container Name	EthTSynGeneral
Parent Container	EthTSyn
Description	This container holds the general parameters of the Ethernet-specific Synchronized Time-base Manager
Configuration Parameters	

SWS Item	[ECUC_EthTSyn_00058]		
Parameter Name	EthTSynDestPhyAddr		
Parent Container	EthTSynGeneral		
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	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00002]
Parameter Name	EthTSynDevErrorDetect
Parent Container	EthTSynGeneral
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		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00089]**		
Parameter Name	EthTSynEnableSecurityEventReporting		
Parent Container	EthTSynGeneral		
Description	Switches the reporting of security events to the IdsM: - true: reporting is enabled. - false: reporting is disabled. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00060]**		
Parameter Name	EthTSynGlobalTimeRxToUplinkSwitchResidenceTime		
Parent Container	EthTSynGeneral		
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		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00061]**		
Parameter Name	EthTSynGlobalTimeUplinkToTxSwitchResidenceTime		
Parent Container	EthTSynGeneral		

▽

△

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		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00018]**		
Parameter Name	EthTSynHardwareTimestampSupport		
Parent Container	EthTSynGeneral		
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	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00012]**		
Parameter Name	EthTSynMainFunctionPeriod		
Parent Container	EthTSynGeneral		
Description	Schedule period of the main function EthTSyn_MainFunction. Unit: seconds.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	]0 .. INF[		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00075]		
Parameter Name	EthTSynMasterSlaveConflictDetection		
Parent Container	EthTSynGeneral		
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		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00029]		
Parameter Name	EthTSynMessageCompliance		
Parent Container	EthTSynGeneral		
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	–		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00059]		
Parameter Name	EthTSynSwitchMgmtRxMessageBufferCount		
Parent Container	EthTSynGeneral		
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		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00081]		
Parameter Name	EthTSynTimeValidationSupport		
Parent Container	EthTSynGeneral		
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		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00015]		
Parameter Name	EthTSynVersionInfoApi		
Parent Container	EthTSynGeneral		
Description	Activate/Deactivate the version information API (EthTSyn_GetVersionInfo). True: version information API activated False: version information API deactivated.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00062]		
Parameter Name	EthTSynEthIfFrameType		
Parent Container	EthTSynGeneral		
Description	The chosen frame owner determines which frames (in respect to ethertype) are received.		
Multiplicity	1		
Type	Reference to EthIfFrameOwnerConfig		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
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. **Tags:** atp.Status=draft

10.2.3 EthTSynSecurityEventRefs

SWS Item	**[ECUC_EthTSyn_00090]**		
Container Name	EthTSynSecurityEventRefs		
Parent Container	EthTSynGeneral		
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. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

SWS Item	**[ECUC_EthTSyn_00093]**		
Parameter Name	SEV_TSYN_ETH_FRESHNESS_NOT_AVAILABLE		
Parent Container	EthTSynSecurityEventRefs		
Description	FV not available from FVM. Context data provides the respective domain ID. **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	Symbolic name reference to IdsMEvent		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00091]**		
Parameter Name	SEV_TSYN_ETH_ICV_GENERATION_FAILED		
Parent Container	EthTSynSecurityEventRefs		
Description	ICV generation for Follow_Up message failed. Context data provides the respective domain ID **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	Symbolic name reference to IdsMEvent		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants

▽

△

	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00092]**		
Parameter Name	SEV_TSYN_ETH_ICV_VERIFICATION_FAILED		
Parent Container	EthTSynSecurityEventRefs		
Description	ICV verification for Follow_Up message failed. Context data provides the respective domain ID. **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	Symbolic name reference to IdsMEvent		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00114]**		
Parameter Name	SEV_TSYN_ETH_SYNC_FUP_SEQUENCE_ERROR		
Parent Container	EthTSynSecurityEventRefs		
Description	Failed to receive correct sequence of SYNC and Follow_Up from the TimeMaster within (EthTSynGlobalTimeFollowUpTimeout). **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	Symbolic name reference to IdsMEvent		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.2.4 EthTSynGlobalTimeDomain

SWS Item	[ECUC_EthTSyn_00004]
Container Name	EthTSynGlobalTimeDomain
Parent Container	EthTSyn
Description	This represents the existence of a global time domain on Ethernet. The EthTSyn module can administrate several global time domains at the same time that in itself form a hierarchy of domains and sub-domains. If the EthTSyn exists it is assumed that at least one global time domain exists.
Configuration Parameters	

SWS Item	[ECUC_EthTSyn_00034]		
Parameter Name	EthTSynFramePrio		
Parent Container	EthTSynGlobalTimeDomain		
Description	This optional parameter, if present, indicates the priority of outgoing EthTSyn messages, if sent via VLAN (used for the 3-bit PCP field of the VLAN tag). If this optional parameter is not present, frames are sent without a priority and VLAN field.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 7		
Default value	–		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00048]		
Parameter Name	EthTSynGlobalTimeDebounceTime		
Parent Container	EthTSynGlobalTimeDomain		
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 Variant Value	true		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00005]		
Parameter Name	EthTSynGlobalTimeDomainId		
Parent Container	EthTSynGlobalTimeDomain		
Description	The global time domain ID.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 127		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00094]		
Parameter Name	EthTSynGlobalTimeRxDebounceTime		
Parent Container	EthTSynGlobalTimeDomain		
Description	This represents the configuration of a RX debounce time for Sync and Follow_Up. Unit: seconds **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0 .. 4]		
Default value	–		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00119]		
Parameter Name	EthTSynClkUnitRef		
Parent Container	EthTSynGlobalTimeDomain		
Description	Reference to a HW clock unit in the Ethernet controller. **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	Symbolic name reference to EthIfClkUnit		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants

△

	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

SWS Item	**[ECUC_EthTSyn_00065]**		
Parameter Name	EthTSynGlobalTimeEthIfRef		
Parent Container	EthTSynGlobalTimeDomain		
Description	This represents the reference to the Ethernet interface taken to fetch the global time information.		
Multiplicity	0..*		
Type	Symbolic name reference to EthIfController		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00013]**		
Parameter Name	EthTSynSynchronizedTimeBaseRef		
Parent Container	EthTSynGlobalTimeDomain		
Description	Mandatory reference to the required synchronized time-base.		
Multiplicity	1		
Type	Symbolic name reference to StbMSynchronizedTimeBase		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
EthTSynGlobalTimeFollowUpDataIDList	0..1	The DataIDList for Follow_Up message ensures the identification of data elements due to CRC calculation and message authentication process.
EthTSynPortConfig	0..*	Configuration of the EthTSyn-Ports within the TimeDomain.
EthTSynPortRole	0..1	Specifying the Role of the EthTSyn-Port (Master or Slave).

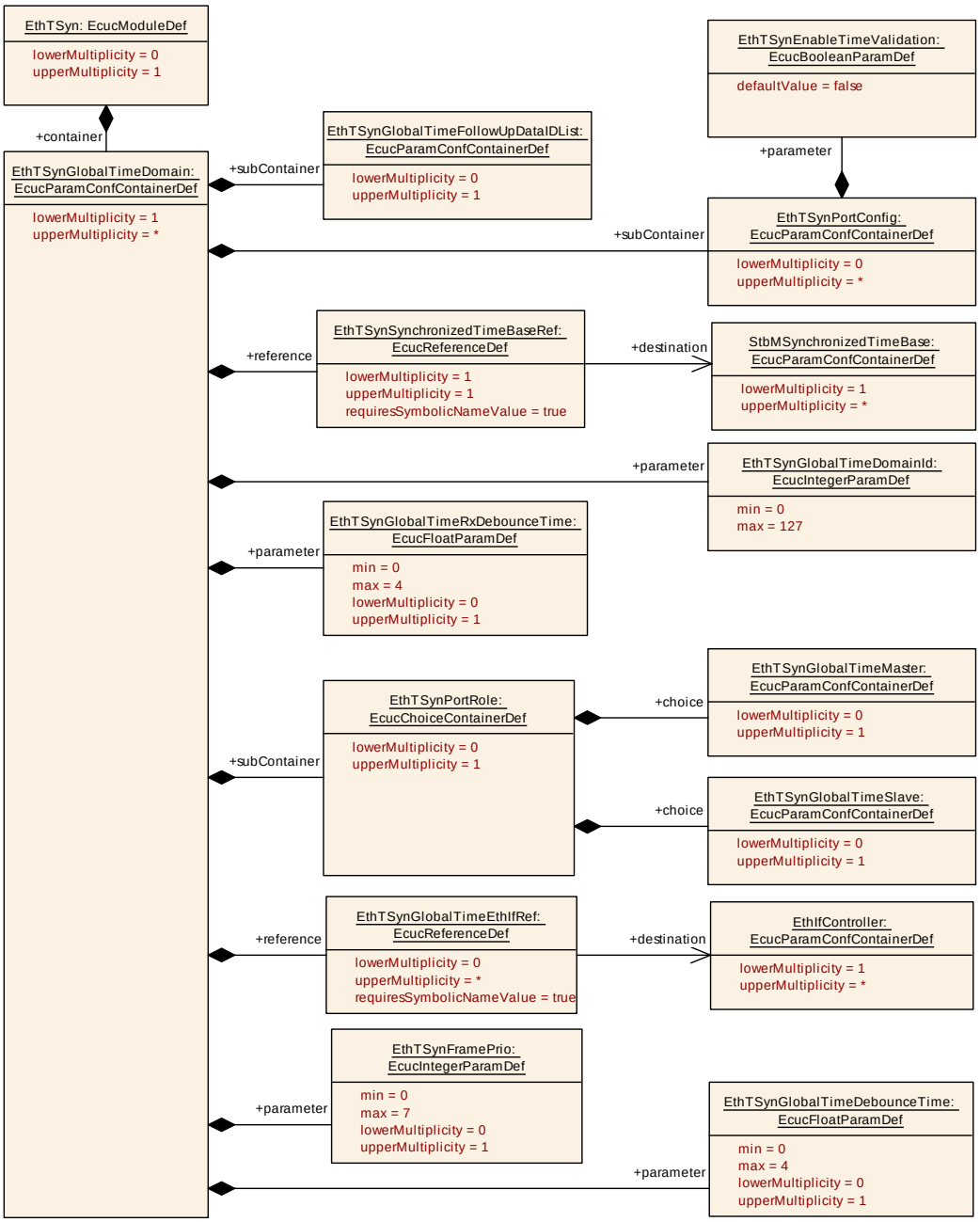

Figure 10.1: EthTSynGlobalTimeDomain

10.2.5 EthTSynGlobalTimeFollowUpDataIDList

SWS Item	[ECUC_EthTSyn_00030]
Container Name	EthTSynGlobalTimeFollowUpDataIDList
Parent Container	EthTSynGlobalTimeDomain
Description	The DataIDList for Follow_Up message ensures the identification of data elements due to CRC calculation and message authentication process.

▽

△

Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

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

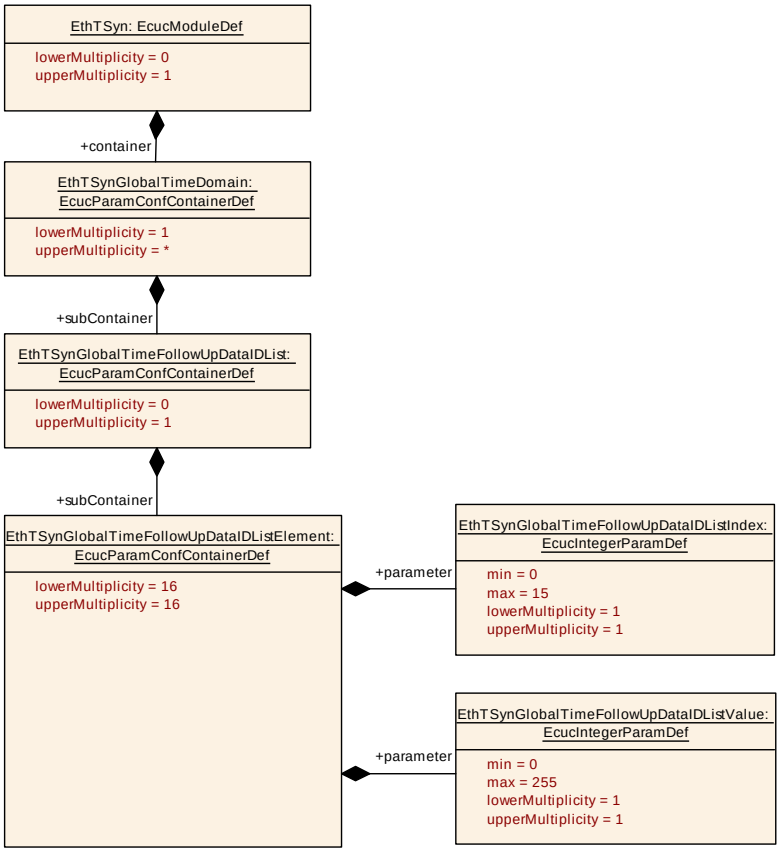

Figure 10.2: EthTSyn_GlobalTimeFollowUpDataIDList

10.2.6 EthTSynGlobalTimeFollowUpDataIDListElement

SWS Item	**[ECUC_EthTSyn_00031]**
Container Name	EthTSynGlobalTimeFollowUpDataIDListElement
Parent Container	EthTSynGlobalTimeFollowUpDataIDList

▽

△

Description	Element of the DataIDList for Follow_Up message ensures the identification of data elements due to CRC calculation and message authentication process.
Configuration Parameters	

SWS Item	**[ECUC_EthTSyn_00032]**		
Parameter Name	EthTSynGlobalTimeFollowUpDataIDListIndex		
Parent Container	EthTSynGlobalTimeFollowUpDataIDListElement		
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 Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00033]**		
Parameter Name	EthTSynGlobalTimeFollowUpDataIDListValue		
Parent Container	EthTSynGlobalTimeFollowUpDataIDListElement		
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 Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.2.7 EthTSynPortConfig

SWS Item	**[ECUC_EthTSyn_00063]**		
Container Name	EthTSynPortConfig		
Parent Container	EthTSynGlobalTimeDomain		
Description	Configuration of the EthTSyn-Ports within the TimeDomain.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants

▽

△

	Link time	–	
	Post-build time	–	
Configuration Parameters			

SWS Item	**[ECUC_EthTSyn_00082]**		
Parameter Name	EthTSynEnableTimeValidation		
Parent Container	EthTSynPortConfig		
Description	Enables/disables time recording for time validation for a specific Time Domain.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00116]**		
Parameter Name	EthTSynGlobalTimePortRole		
Parent Container	EthTSynPortConfig		
Description	Parameter to set the port behavior to Time Slave, Time Master or Dynamic (Time Slave or Time Master at runtime).		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	DYNAMIC	–	
	TIME_MASTER	–	
	TIME_SLAVE	–	
Default value	DYNAMIC		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00010]**		
Parameter Name	EthTSynGlobalTimeTxPeriod		
Parent Container	EthTSynPortConfig		
Description	This represents configuration of the TX period. Unit: seconds		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0 .. INF[		
Default value	–		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	

▽

△

Scope / Dependency	scope: local

SWS Item	**[ECUC_EthTSyn_00066]**		
Parameter Name	EthTSynSwitchManagementEthSwitchPortRef		
Parent Container	EthTSynPortConfig		
Description	In an AVB-Bridge config, this reference is used to assign the EthTSyn-Port to an Ethernet Switch-Port.		
Multiplicity	0..1		
Type	Symbolic name reference to EthSwtPort		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
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. **Tags:** atp.Status=draft
EthTSynPdelayConfig	1	Configuration of cyclic propagation delay measurement.

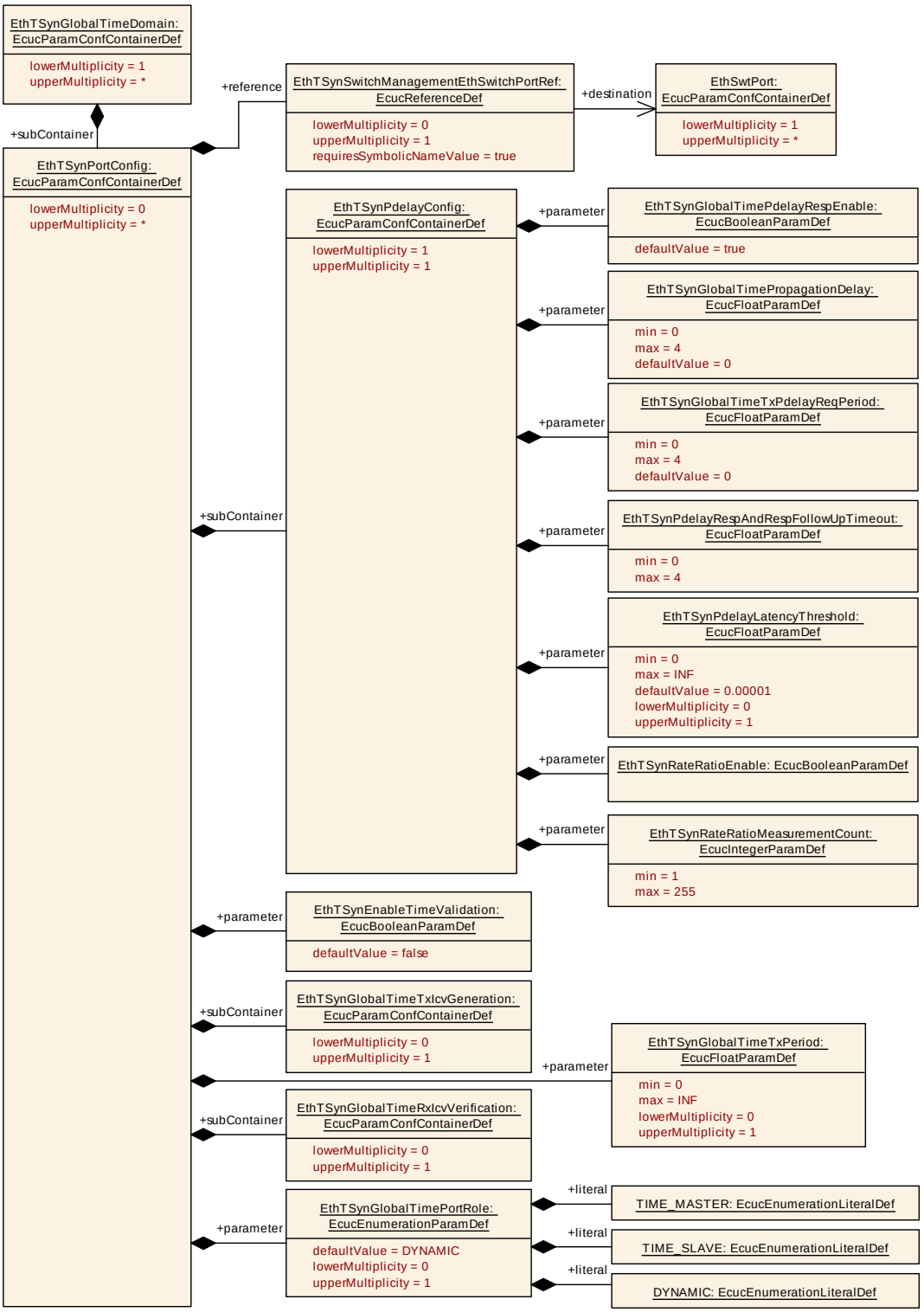

Figure 10.3: EthTSyn_PortConfig

10.2.8 EthTSynGlobalTimeTxIcvGeneration

SWS Item	**[ECUC_EthTSyn_00096]**		
Container Name	EthTSynGlobalTimeTxIcvGeneration		
Parent Container	EthTSynPortConfig		
Description	This container collects configuration that shall be used for ICV generation. **Tags:** atp.Status=draft		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

SWS Item	**[ECUC_EthTSyn_00098]**		
Parameter Name	EthTSynIcvGenerationBase		
Parent Container	EthTSynGlobalTimeTxIcvGeneration		
Description	Symmetric or asymmetric cryptography selection for the ICV generation **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	ICV_MAC	Symmetric cryptography selection for the ICV generation.	
	ICV_SIGNATURE	Asymmetric cryptography selection for the ICV generation.	
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00101]**		
Parameter Name	EthTSynIcvGenerationTimeout		
Parent Container	EthTSynGlobalTimeTxIcvGeneration		
Description	Timeout of ICV generation (respective CSM job completion in asynchronous behaviour). A value of 0 disables the ICV timeout monitoring. Unit: Seconds **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. INF[		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00099]		
Parameter Name	EthTSynIcvTxLength		
Parent Container	EthTSynGlobalTimeTxIcvGeneration		
Description	Length of ICV to be transmitted within Follow_Up Message on the bus (in bytes). **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 1061		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00113]		
Parameter Name	EthTSynTxAuthenticationBuildAttempts		
Parent Container	EthTSynGlobalTimeTxIcvGeneration		
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. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	0		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00097]		
Parameter Name	EthTSynIcvGenerationFvIdRef		
Parent Container	EthTSynGlobalTimeTxIcvGeneration		
Description	This represents the reference to the FV taken to generate the ICV generation. **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	Symbolic name reference to StbMFreshnessValue		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	

△

Scope / Dependency	scope: local

SWS Item	**[ECUC_EthTSyn_00100]**		
Parameter Name	EthTSynIcvGenerationJobRef		
Parent Container	EthTSynGlobalTimeTxIcvGeneration		
Description	This represents the reference to the CSM job to fetch the CSM job ID. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	Symbolic name reference to CsmJob		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.2.9 EthTSynGlobalTimeRxIcvVerification

SWS Item	**[ECUC_EthTSyn_00104]**		
Container Name	EthTSynGlobalTimeRxIcvVerification		
Parent Container	EthTSynPortConfig		
Description	This container collects configuration that shall be used for ICV verification.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

SWS Item	**[ECUC_EthTSyn_00107]**		
Parameter Name	EthTSynIcvRxLength		
Parent Container	EthTSynGlobalTimeRxIcvVerification		
Description	Length of ICV to be transmitted within Follow_Up Message on the bus (in bytes). **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 1061		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00110]		
Parameter Name	EthTSynIcvVerificationAttempts		
Parent Container	EthTSynGlobalTimeRxIcvVerification		
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. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	0		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00106]		
Parameter Name	EthTSynIcvVerificationBase		
Parent Container	EthTSynGlobalTimeRxIcvVerification		
Description	Symmetric or asymmetric cryptography selection for the ICV generation **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	ICV_MAC	Symmetric cryptography selection for the ICV generation. **Tags:** atp.Status=draft	
	ICV_SIGNATURE	Asymmetric cryptography selection for the ICV generation. **Tags:** atp.Status=draft	
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00109]		
Parameter Name	EthTSynIcvVerificationTimeout		
Parent Container	EthTSynGlobalTimeRxIcvVerification		
Description	Timeout of ICV verification (respective CSM job completion in asynchronous behavior). A value of 0 disables the ICV timeout monitoring. Unit: Seconds **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. INF[		
Default value	–		
Post-Build Variant Value	false		

▽

△

Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00112]**		
Parameter Name	EthTSynRxAuthenticationBuildAttempts		
Parent Container	EthTSynGlobalTimeRxIcvVerification		
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. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	0		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00105]**		
Parameter Name	EthTSynIcvVerificationFvIdRef		
Parent Container	EthTSynGlobalTimeRxIcvVerification		
Description	This represents the reference to the FV taken to verify the ICV. **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	Symbolic name reference to StbMFreshnessValue		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00108]**
Parameter Name	EthTSynIcvVerificationJobRef
Parent Container	EthTSynGlobalTimeRxIcvVerification
Description	This represents the reference to the CSM job to fetch the CSM job ID. **Tags:** atp.Status=draft
Multiplicity	1
Type	Symbolic name reference to CsmJob

▽

△

Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.2.10 EthTSynPortRole

SWS Item	**[ECUC_EthTSyn_00067]**		
Choice Container Name	EthTSynPortRole		
Parent Container	EthTSynGlobalTimeDomain		
Description	Specifying the Role of the EthTSyn-Port (Master or Slave).		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	

Container Choices		
Container Name	**Multiplicity**	**Scope / Dependency**
EthTSynGlobalTimeMaster	0..1	Configuration of a (global) time master. Each time domain is required to have exactly one global time master, but may have multiple ports acting as time (sub-) master (see Time Gateway) to relay global time from the global time master to the time slaves. The global time master may or may not exist on the configured ECU. The exact role of the port is derived implicitly.
EthTSynGlobalTimeSlave	0..1	Configuration of a time slave. Each global time domain is required to have at least one time slave. The configured ECU may or may not represent a time slave.

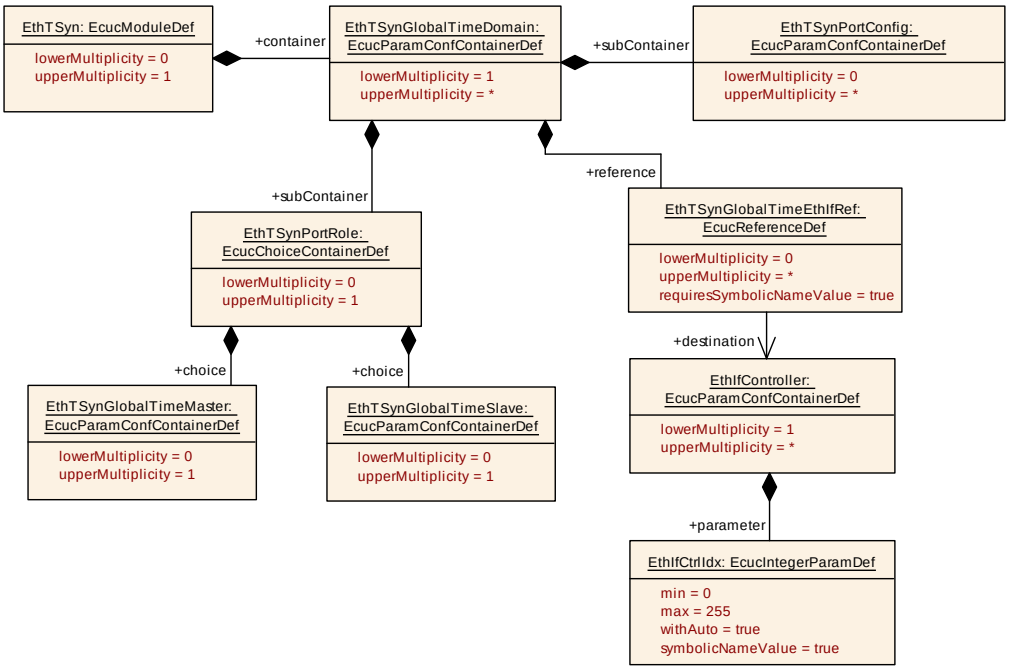

Figure 10.4: EthTSyn_GlobalTimePdu

10.2.11 EthTSynPdelayConfig

SWS Item	**[ECUC_EthTSyn_00068]**		
Container Name	EthTSynPdelayConfig		
Parent Container	EthTSynPortConfig		
Description	Configuration of cyclic propagation delay measurement.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

SWS Item	**[ECUC_EthTSyn_00069]**
Parameter Name	EthTSynGlobalTimePdelayRespEnable
Parent Container	EthTSynPdelayConfig
Description	This parameter allows disabling Pdelay_Resp / Pdelay_Resp_Follow_Up transmission, if no Pdelay_Req messages are expected. FALSE: No Pdelay requests expected. Pdelay_Resp / Pdelay_Resp_Follow_Up transmission is disabled. TRUE: Pdelay requests expected. Pdelay_Resp / Pdelay_Resp_Follow_Up transmission is enabled.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	true
Post-Build Variant Value	true

▽

△

Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00070]**		
Parameter Name	EthTSynGlobalTimePropagationDelay		
Parent Container	EthTSynPdelayConfig		
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 Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00071]**		
Parameter Name	EthTSynGlobalTimeTxPdelayReqPeriod		
Parent Container	EthTSynPdelayConfig		
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 Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00076]**
Parameter Name	EthTSynPdelayLatencyThreshold
Parent Container	EthTSynPdelayConfig
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 Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00074]**		
Parameter Name	EthTSynPdelayRespAndRespFollowUpTimeout		
Parent Container	EthTSynPdelayConfig		
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 Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00118]**		
Parameter Name	EthTSynRateRatioEnable		
Parent Container	EthTSynPdelayConfig		
Description	Enables/disables neighbor rate ratio calculation according to IEEE 802.1AS. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

SWS Item	**[ECUC_EthTSyn_00117]**
Parameter Name	EthTSynRateRatioMeasurementCount
Parent Container	EthTSynPdelayConfig
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 Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.2.12 EthTSynGlobalTimeMaster

SWS Item	**[ECUC_EthTSyn_00008]**		
Container Name	EthTSynGlobalTimeMaster		
Parent Container	EthTSynPortRole		
Description	Configuration of a (global) time master. Each time domain is required to have exactly one global time master, but may have multiple ports acting as time (sub-) master (see Time Gateway) to relay global time from the global time master to the time slaves. The global time master may or may not exist on the configured ECU. The exact role of the port is derived implicitly.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

SWS Item	**[ECUC_EthTSyn_00047]**		
Parameter Name	EthTSynCyclicMsgResumeTime		
Parent Container	EthTSynGlobalTimeMaster		
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 Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00039]**
Parameter Name	EthTSynGlobalTimeTxCrcSecured
Parent Container	EthTSynGlobalTimeMaster

▽

△

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 Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00115]**		
Parameter Name	EthTSynHoldOverTime		
Parent Container	EthTSynGlobalTimeMaster		
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 Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00046]**		
Parameter Name	EthTSynImmediateTimeSync		
Parent Container	EthTSynGlobalTimeMaster		
Description	Enables/Disables the cyclic polling of StbM_GetTimeBaseUpdateCounter() within Eth TSyn_MainFunction().		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00095]		
Parameter Name	EthTSynTLVFollowUpICVSubTLV		
Parent Container	EthTSynGlobalTimeMaster		
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. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00038]		
Parameter Name	EthTSynTxSubTLVOFS		
Parent Container	EthTSynGlobalTimeMaster		
Description	Definition of whether (true) or not (false) a Sub-TLV:OFS Secured or Sub-TLV:OFS Not Secured shall be sent in the AUTOSAR TLV of a Follow_Up message.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00036]		
Parameter Name	EthTSynTxSubTLVStatus		
Parent Container	EthTSynGlobalTimeMaster		
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 Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00035]
Parameter Name	EthTSynTxSubTLVTime
Parent Container	EthTSynGlobalTimeMaster

△

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 Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00037]**		
Parameter Name	EthTSynTxSubTLVUserData		
Parent Container	EthTSynGlobalTimeMaster		
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 Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

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

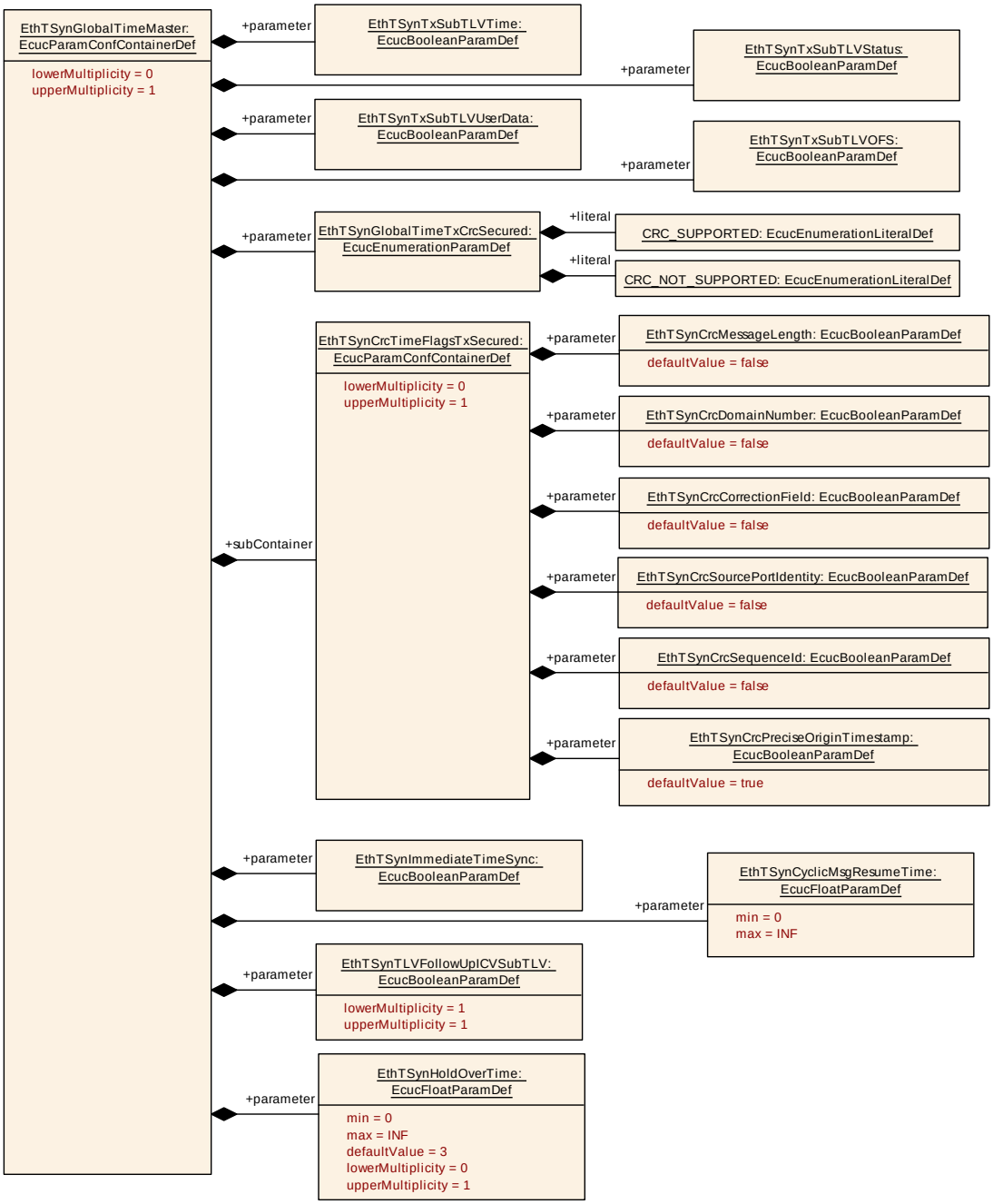

Figure 10.5: EthTSyn_GlobalTimeMaster

10.2.13 EthTSynCrcTimeFlagsTxSecured

SWS Item	[ECUC_EthTSyn_00057]
Container Name	EthTSynCrcTimeFlagsTxSecured
Parent Container	EthTSynGlobalTimeMaster

▽

△

Description	This container collects definitions which parts of the Follow_Up message elements shall be used for CRC calculation.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

SWS Item	**[ECUC_EthTSyn_00042]**		
Parameter Name	EthTSynCrcCorrectionField		
Parent Container	EthTSynCrcTimeFlagsTxSecured		
Description	correctionField from the Follow_Up Message Header shall be included in CRC calculation.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00041]**		
Parameter Name	EthTSynCrcDomainNumber		
Parent Container	EthTSynCrcTimeFlagsTxSecured		
Description	domainNumber from the Follow_Up Message Header shall be included in CRC calculation.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00040]**		
Parameter Name	EthTSynCrcMessageLength		
Parent Container	EthTSynCrcTimeFlagsTxSecured		
Description	messageLength from the Follow_Up Message Header shall be included in CRC calculation.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	

▽

△

Scope / Dependency	scope: local

SWS Item	**[ECUC_EthTSyn_00045]**		
Parameter Name	EthTSynCrcPreciseOriginTimestamp		
Parent Container	EthTSynCrcTimeFlagsTxSecured		
Description	preciseOriginTimestamp from the Follow_Up Message Field shall be included in CRC calculation.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	true		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00044]**		
Parameter Name	EthTSynCrcSequenceId		
Parent Container	EthTSynCrcTimeFlagsTxSecured		
Description	sequenceId from the Follow_Up Message Header shall be included in CRC calculation.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00043]**		
Parameter Name	EthTSynCrcSourcePortIdentity		
Parent Container	EthTSynCrcTimeFlagsTxSecured		
Description	sourcePortIdentity from the Follow_Up Message Header shall be included in CRC calculation.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

10.2.14 EthTSynGlobalTimeSlave

SWS Item	**[ECUC_EthTSyn_00009]**		
Container Name	EthTSynGlobalTimeSlave		
Parent Container	EthTSynPortRole		
Description	Configuration of a time slave. Each global time domain is required to have at least one time slave. The configured ECU may or may not represent a time slave.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

SWS Item	**[ECUC_EthTSyn_00007]**		
Parameter Name	EthTSynGlobalTimeFollowUpTimeout		
Parent Container	EthTSynGlobalTimeSlave		
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 Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00084]**		
Parameter Name	EthTSynGlobalTimeSequenceCounterHysteresis		
Parent Container	EthTSynGlobalTimeSlave		
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 Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00083]		
Parameter Name	EthTSynGlobalTimeSequenceCounterJumpWidth		
Parent Container	EthTSynGlobalTimeSlave		
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 Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00049]	
Parameter Name	EthTSynRxCrcValidated	
Parent Container	EthTSynGlobalTimeSlave	
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, 0x44, 0x50 or 0x60.
	CRC_OPTIONAL	If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 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, 0x44, 0x50 or 0x60, that contain an incorrect CRC value. EthTSyn rejects Follow_Up messages with Sub-TLVs of Type 0x34, 0x51 or 0x61.
Post-Build Variant Value	true	
Value Configuration Class	**Pre-compile time**	X All Variants
	Link time	–
	Post-build time	–
Scope / Dependency	scope: local	

SWS Item	[ECUC_EthTSyn_00103]
Parameter Name	EthTSynRxIcvVerificationType
Parent Container	EthTSynGlobalTimeSlave
Description	This parameter controls whether or not ICV verification shall be supported. **Tags:** atp.Status=draft
Multiplicity	1
Type	EcucEnumerationParamDef

▽

△

Range	ICV_IGNORED	The Timesync module shall not verify the ICV. **Tags:** atp.Status=draft	
	ICV_NOT_VERIFIED	The Timesync module accepts only Time Synchronization messages, which are not ICV secured. All other Time Synchronization messages are ignored. **Tags:** atp.Status=draft	
	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. **Tags:** atp.Status=draft	
	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. **Tags:** atp.Status=draft	
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00088]**		
Parameter Name	EthTSynRxSubTLVOFS		
Parent Container	EthTSynGlobalTimeSlave		
Description	Definition of whether or not a Sub-TLV:OFS Secured or Sub-TLV:OFS Not Secured shall be present and shall be evaluated when processing a received Follow_Up message.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00086]**
Parameter Name	EthTSynRxSubTLVStatus
Parent Container	EthTSynGlobalTimeSlave
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		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00085]**		
Parameter Name	EthTSynRxSubTLVTime		
Parent Container	EthTSynGlobalTimeSlave		
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		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00087]**		
Parameter Name	EthTSynRxSubTLVUserData		
Parent Container	EthTSynGlobalTimeSlave		
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		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

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

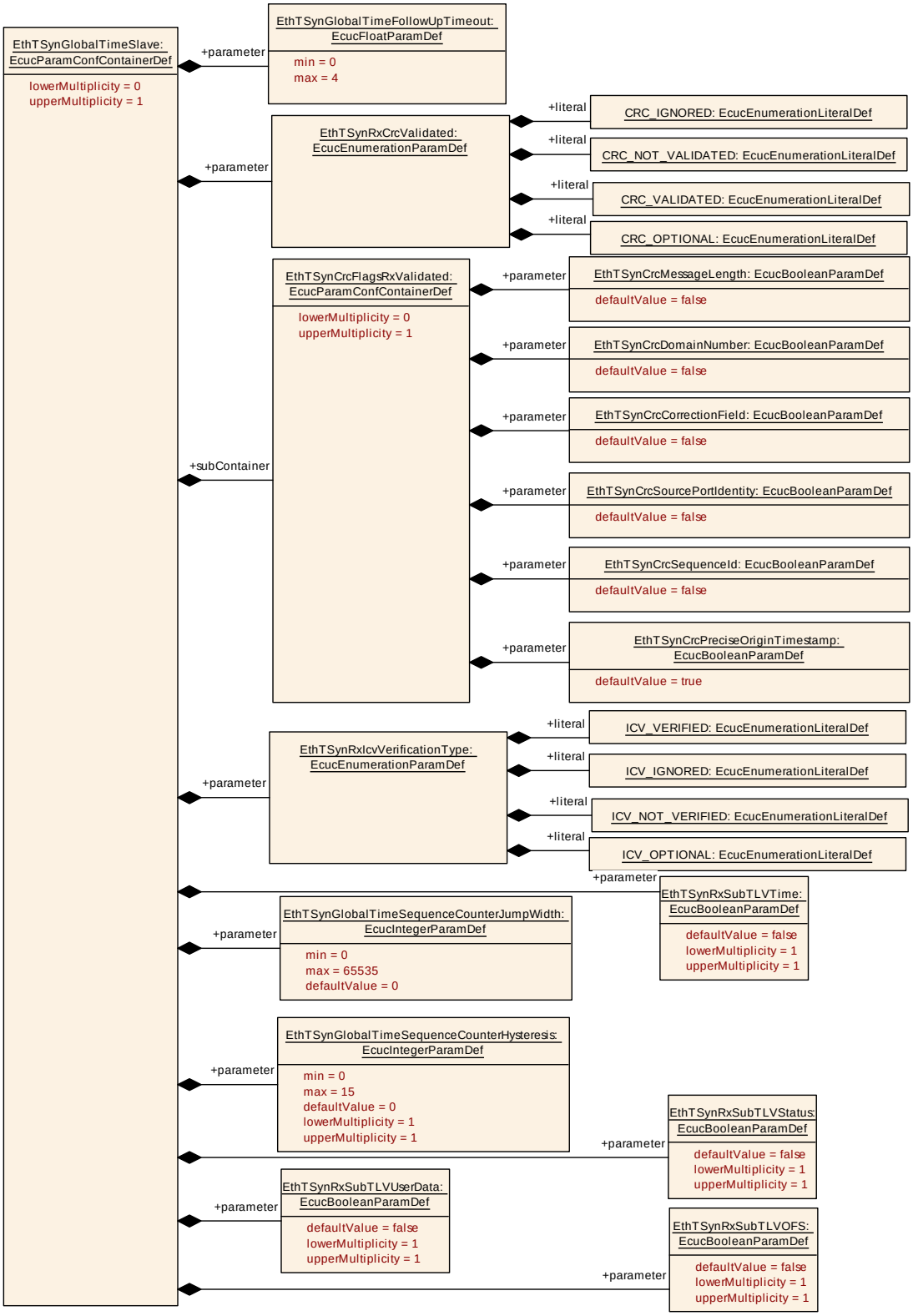

Figure 10.6: EthTSyn_GlobalTimeSlave

10.2.15 EthTSynCrcFlagsRxValidated

SWS Item	[ECUC_EthTSyn_00050]		
Container Name	EthTSynCrcFlagsRxValidated		
Parent Container	EthTSynGlobalTimeSlave		
Description	This container collects definitions which parts of the Follow_Up message elements shall be included in CRC validation.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

SWS Item	[ECUC_EthTSyn_00042]		
Parameter Name	EthTSynCrcCorrectionField		
Parent Container	EthTSynCrcFlagsRxValidated		
Description	correctionField from the Follow_Up Message Header shall be included in CRC calculation.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00041]		
Parameter Name	EthTSynCrcDomainNumber		
Parent Container	EthTSynCrcFlagsRxValidated		
Description	domainNumber from the Follow_Up Message Header shall be included in CRC calculation.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	[ECUC_EthTSyn_00040]
Parameter Name	EthTSynCrcMessageLength
Parent Container	EthTSynCrcFlagsRxValidated
Description	messageLength from the Follow_Up Message Header shall be included in CRC calculation.
Multiplicity	1
Type	EcucBooleanParamDef

△

Default value	false		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00045]**		
Parameter Name	EthTSynCrcPreciseOriginTimestamp		
Parent Container	EthTSynCrcFlagsRxValidated		
Description	preciseOriginTimestamp from the Follow_Up Message Field shall be included in CRC calculation.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	true		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00044]**		
Parameter Name	EthTSynCrcSequenceId		
Parent Container	EthTSynCrcFlagsRxValidated		
Description	sequenceId from the Follow_Up Message Header shall be included in CRC calculation.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

SWS Item	**[ECUC_EthTSyn_00043]**		
Parameter Name	EthTSynCrcSourcePortIdentity		
Parent Container	EthTSynCrcFlagsRxValidated		
Description	sourcePortIdentity from the Follow_Up Message Header shall be included in CRC calculation.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

No Included Containers

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 (EthTSynGlobalTimeDomainId).

[constr_0001] ⌈The EthTSynPortConfig container exists for Synchronized Time Domains (EthTSynGlobalTimeDomain 0 .. 127) only.⌋*()*

[constr_0002] ⌈If the CSM job used to generate ICV is configured in synchronous behaviour, the EthTSynIcvGenerationTimeout shall be set to 0.⌋*()*

[constr_0003] ⌈If the CSM job used to verify ICV is configured in synchronous behavior, the EthTSynIcvVerificationTimeout shall be set to 0.⌋*()*

[constr_0004] ⌈The parameter EthTSynGlobalTimeTxPeriod shall determine the transmission interval of Sync messages. It shall be available if and only if the parameter EthTSynGlobalTimePortRole is set to TIME_MASTER or DYNAMIC.⌋*()*

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

[constr_0005] ⌈The parameter EthTSynGlobalTimePortRole shall not be configured as TIME_SLAVE for two ports under same EthTSynGlobalTimeDomain.⌋*()*

10.4 Published Information

For details refer to the chapter 10.3 "Published Information" in [3].

A Not applicable requirements

[SWS_EthTSyn_NA_00996] ⌈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.⌋*(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_00012, RS_TS_00013, 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_00034, RS_TS_00035, RS_TS_00036, RS_TS_00037, RS_TS_00038, RS_TS_00039, RS_TS_20040, RS_TS_20041, RS_TS_20042, RS_TS_20043, RS_TS_20044, RS_TS_20045, RS_TS_20046, RS_TS_20060, RS_TS_20068, RS_TS_20070, RS_TS_20071, RS_TS_20073, RS_TS_20074)*

[SWS_EthTSyn_NA_00997] ⌈This specification item references requirements, that cannot be traced because they apply only to EcuC elements.⌋*(SRS_BSW_00388, SRS_BSW_00389, SRS_BSW_00390, SRS_BSW_00392, SRS_BSW_00393, SRS_BSW_00394, SRS_BSW_00395, SRS_BSW_00396, SRS_BSW_00401, SRS_BSW_00403, SRS_BSW_00417, SRS_BSW_00478)*

Note: EcuC elements do not support trace links

[SWS_EthTSyn_NA_00998] ⌈This specification item references requirements, that cannot be traced to a specific spec item in the EthTSyn.⌋*(SRS_BSW_00172, SRS_BSW_00310, SRS_BSW_00312, SRS_BSW_00330, SRS_BSW_00331, SRS_BSW_00343, SRS_BSW_00345, SRS_BSW_00369, SRS_BSW_00377, SRS_BSW_00383, SRS_BSW_00384, SRS_BSW_00399, SRS_BSW_00419, SRS_BSW_00448, SRS_BSW_00453, SRS_BSW_00483, SRS_BSW_00484, SRS_BSW_00485, SRS_BSW_00486)*

Note: These requirements are generic in nature (e.g. non-functional requirements) 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] ⌈These requirements are not applicable to EthTSyn.⌋*(SRS_BSW_00005, SRS_BSW_00162, SRS_BSW_00168, SRS_BSW_00170, SRS_BSW_00336, SRS_BSW_00351, SRS_BSW_00357, SRS_BSW_00375, SRS_BSW_00413, SRS_BSW_00416, SRS_BSW_00417, SRS_BSW_00422, SRS_BSW_00425, SRS_BSW_00432, SRS_BSW_00449, SRS_BSW_00454, SRS_BSW_00456, SRS_BSW_00457, SRS_BSW_00458, SRS_BSW_00459, SRS_BSW_00461, SRS_BSW_00462, SRS_BSW_00466, SRS_BSW_00469, SRS_BSW_00470, SRS_BSW_00471, SRS_BSW_00472, SRS_BSW_00473, SRS_BSW_00479, SRS_BSW_00494)*

B Changed Traces History

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

B.1 Change History of this document according to AUTOSAR Release R23-11

B.1.1 Added Specification Items in R23-11

Number	Heading
[SWS_EthTSyn_-00263]	
[SWS_EthTSyn_-00264]	
[SWS_EthTSyn_-00265]	
[SWS_EthTSyn_-00266]	
[SWS_EthTSyn_-00267]	
[SWS_EthTSyn_-00268]	
[SWS_EthTSyn_-00400]	
[SWS_EthTSyn_-00401]	
[SWS_EthTSyn_-00402]	
[SWS_EthTSyn_-00403]	
[SWS_EthTSyn_-00404]	
[SWS_EthTSyn_-00405]	
[SWS_EthTSyn_-00406]	
[SWS_EthTSyn_-00407]	
[SWS_EthTSyn_-00408]	

▽

△

Number	Heading
[SWS_EthTSyn_-00409]	
[SWS_EthTSyn_-00410]	
[SWS_EthTSyn_-00411]	
[SWS_EthTSyn_-00412]	
[SWS_EthTSyn_-00413]	
[SWS_EthTSyn_-00414]	
[SWS_EthTSyn_-00415]	
[SWS_EthTSyn_-00416]	
[SWS_EthTSyn_NA_-00996]	
[SWS_EthTSyn_NA_-00997]	
[SWS_EthTSyn_NA_-00998]	
[SWS_EthTSyn_NA_-00999]	

Table B.1: Added Specification Items in R23-11

B.1.2 Changed Specification Items in R23-11

Number	Heading
[SWS_EthTSyn_-00017]	
[SWS_EthTSyn_-00030]	Definiton of development errors in module EthTSyn
[SWS_EthTSyn_-00031]	Definition of imported datatypes of module EthTSyn
[SWS_EthTSyn_-00043]	Definition of callback function EthTSyn_TrcvLinkStateChg
[SWS_EthTSyn_-00047]	Definition of optional interfaces in module EthTSyn

▽

△

Number	Heading
[SWS_EthTSyn_-00052]	
[SWS_EthTSyn_-00086]	
[SWS_EthTSyn_-00127]	
[SWS_EthTSyn_-00128]	
[SWS_EthTSyn_-00136]	
[SWS_EthTSyn_-00137]	
[SWS_EthTSyn_-00139]	
[SWS_EthTSyn_-00150]	
[SWS_EthTSyn_-00172]	
[SWS_EthTSyn_-00190]	
[SWS_EthTSyn_-00210]	
[SWS_EthTSyn_-00211]	
[SWS_EthTSyn_-00224]	
[SWS_EthTSyn_-00225]	
[SWS_EthTSyn_-00228]	
[SWS_EthTSyn_-00229]	
[SWS_EthTSyn_-00236]	
[SWS_EthTSyn_-00241]	
[SWS_EthTSyn_-00243]	
[SWS_EthTSyn_-00248]	
[SWS_EthTSyn_-00249]	
[SWS_EthTSyn_-00250]	

▽

△

Number	Heading
[SWS_EthTSyn_-00256]	
[SWS_EthTSyn_-00257]	
[SWS_EthTSyn_-00258]	
[SWS_EthTSyn_-00261]	Security events for EthTSyn
[SWS_EthTSyn_-00262]	Context data of respective Security events of EthTSyn

Table B.2: Changed Specification Items in R23-11

B.1.3 Deleted Specification Items in R23-11

Number	Heading
[SWS_EthTSyn_-00235]	

Table B.3: Deleted Specification Items in R23-11

C Changed Constraints History

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

C.1 Change History of this document according to AUTOSAR Release R23-11

C.1.1 Added Constraints in R23-11

Number	Heading
[constr_0001]	
[constr_0002]	
[constr_0003]	
[constr_0004]	
[constr_0005]	

Table C.1: Added Constraints in R23-11

C.1.2 Changed Constraints in R23-11

none

C.1.3 Deleted Constraints in R23-11

Number	Heading
[SWS_EthTSyn_CONSTR_00001]	
[SWS_EthTSyn_CONSTR_00002]	

Table C.2: Deleted Constraints in R23-11