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	R20-11

Document Change History

Date	Release	Changed by	Change Description
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

Document Change History			
Date	**Release**	**Changed by**	**Change Description**
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_EthSetGlobalTime() added - sequence diagrams corrected • 'const' added to input arguments passed by pointer
2014-10-31	4.2.1	AUTOSAR Release Management	• Initial Release

Disclaimer

Table of Contents

1 Introduction and functional overview

The EthTSyn module handles the Time Synchronization Protocol on Ethernet as specified in [12].

In addition to what is specified in [12] 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 [6]), 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.

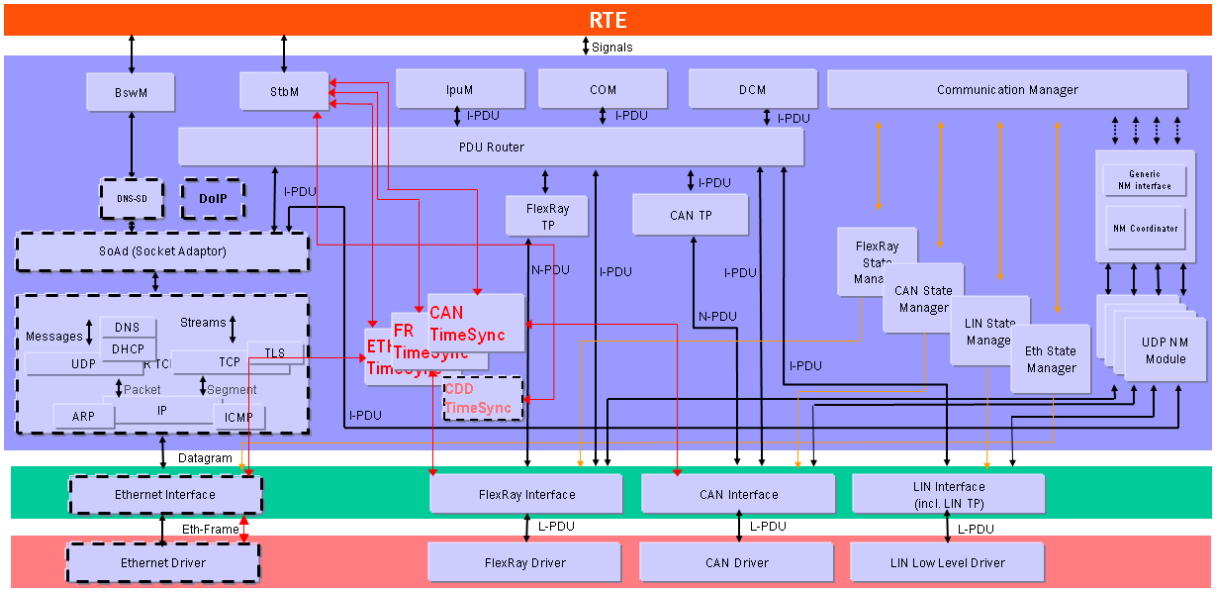


Figure 1: Timesync modules in the AUTOSAR Layered Architecture

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 [5].

Abbreviation / Acronym:	Description
(G)TD	(Global) Time Domain
(G)TM	(Global)Time Master
<Bus>TSyn	A bus specific Time Synchronization module
AVB	Audio Video Bridging
BMCA	Best Master Clock Algorithm
CID	Company ID (IEEE)
CRC	Cyclic Redundancy Checksum
Debounce Time	Minimum gap between 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)
GM(C)	Grand Master (Clock)
OFS	Offset synchronization
Pdelay	Propagation / path delay as given in IEEE 802.1AS
Pdelay_Req	Propagation / path delay request message
Pdelay_Resp	Propagation / path delay response message
Pdelay_Resp_Follow_Up	Propagation / path delay Follow-Up message
PDU	Protocol Data Unit
PTP	Precision Time Protocol
StbM	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] AUTOSAR Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf

[2] General Requirements on Basic Software Modules
AUTOSAR_SRS_BSWGeneral.pdf

[3] Requirements on Time Synchronization
AUTOSAR_RS_TimeSynchronization.pdf

[4] Requirements on Ethernet Support in AUTOSAR
AUTOSAR_SRS_Ethernet.pdf

[5] General Specification of Basic Software Modules
AUTOSAR_SWS_BSWGeneral.pdf

[6] Specification of Synchronized Time-Base Manager
AUTOSAR_SWS_SynchronizedTimeBaseManager.pdf

[7] Specification of the Ethernet Interface
AUTOSAR_SWS_EthernetInterface.pdf

[8] Specification of Default Error Tracer
AUTOSAR_SWS_DefaultErrorTracer.pdf

[9] Specification of Basic Software Mode Manager
AUTOSAR_SWS_BSWModeManager.pdf

[10] AUTOSAR Specification of CRC Routines
AUTOSAR_SWS_CRCLibrary.pdf

[11] Specification of ECU Configuration
AUTOSAR_TPS_ECUConfiguration.pdf

[12] Specification of Time Synchronization Protocol
AUTOSAR_PRS_TimeSynchronizationProtocol.pdf

3.2 Related standards and norms

[13] IEEE Standard 802.1AS™- 30 of March 2011
http://standards.ieee.org/getieee802/download/802.1AS-2011.pdf

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

3.3 Related specification

AUTOSAR provides

- a General Specification on Basic Software (SWS BSW General [5]) which is also valid for EthTSyn and
- a Time Synchronization Protocol Specification (PRS Time Synchronization Protocol [12]) which is also valid for EthTSyn.

Thus,

- the SWS BSW General [5] and
- the PRS Time Synchronization Protocol [12]

shall be considered additionally and as required specification for EthTSyn.

4 Constraints and assumptions

4.1 Limitations

- No support of BMCA protocol, like specified in [13].
- No support of `Announce` and `Signaling` messages, like specified in [13].
- 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 [6]) 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.
- Because of (□), EthTSyn will not maintain the Ethernet HW clock, but may use it as a source for the Virtual Local Time.
- While IEEE 802.1AS states, that IEEE 802.1AS message 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.

4.2 Accuracy

Time Master and Time Slave shall work with a Time Base reference clock accuracy as defined in“[13], ANNEX B.1.2 Time measurement granularity”.

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) and the Default Error Tracer (DET).

- StbM – Get and set the current time value
- EthIf – Receiving and transmitting messages
- BswM – Coordination of network access
- DET – Reporting of development errors

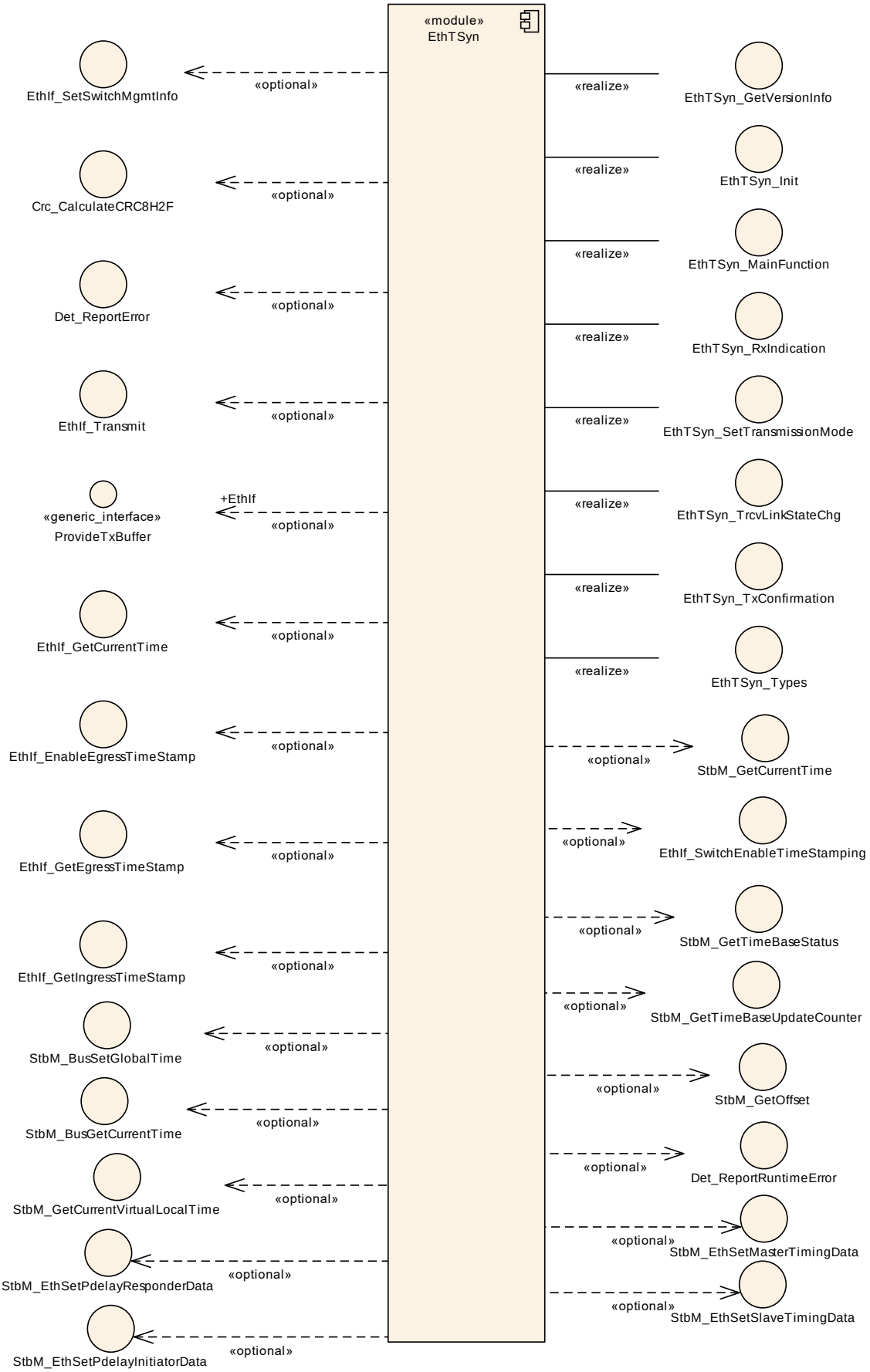

«module»
EthTSyn
EthIf_SetSwitchMgmtInfo
«optional»
Crc_CalculateCRC8H2F
«optional»
Det_ReportError
«optional»
EthIf_Transmit
«optional»
«generic_interface»
ProvideTxBuffer
+EthIf
«optional»
EthIf_GetCurrentTime
«optional»
EthIf_EnableEgressTimeStamp
«optional»
EthIf_GetEgressTimeStamp
«optional»
EthIf_GetIngressTimeStamp
«optional»
StbM_BusSetGlobalTime
«optional»
StbM_BusGetCurrentTime
«optional»
StbM_GetCurrentVirtualLocalTime
«optional»
StbM_EthSetPdelayResponderData
«optional»
StbM_EthSetPdelayInitiatorData
«optional»
«realize»
EthTSyn_GetVersionInfo
«realize»
EthTSyn_Init
«realize»
EthTSyn_MainFunction
«realize»
EthTSyn_RxIndication
«realize»
EthTSyn_SetTransmissionMode
«realize»
EthTSyn_TrcvLinkStateChg
«realize»
EthTSyn_TxConfirmation
«realize»
EthTSyn_Types
«optional»
StbM_GetCurrentTime
«optional»
EthIf_SwitchEnableTimeStamping
«optional»
StbM_GetTimeBaseStatus
«optional»
StbM_GetTimeBaseUpdateCounter
«optional»
StbM_GetOffset
«optional»
Det_ReportRuntimeError
«optional»
StbM_EthSetMasterTimingData
«optional»
StbM_EthSetSlaveTimingData

Figure 2: 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 [5].

6 Requirements traceability

Requirement	Description	Satisfied by
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, SWS_EthTSyn_00224, SWS_EthTSyn_00225
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
RS_TS_20048	The Timesync over Ethernet module shall support IEEE 802.1AS as well as AUTOSAR extensions	SWS_EthTSyn_00010, SWS_EthTSyn_00013, SWS_EthTSyn_00014, SWS_EthTSyn_00017, SWS_EthTSyn_00019, SWS_EthTSyn_00020, SWS_EthTSyn_00021, SWS_EthTSyn_00022, SWS_EthTSyn_00024, 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_00148, SWS_EthTSyn_00159, SWS_EthTSyn_00160, SWS_EthTSyn_00161, SWS_EthTSyn_00162, SWS_EthTSyn_00179, 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
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
RS_TS_20053	The configuration of the	SWS_EthTSyn_00051

	Time Synchronization over Ethernet module shall allow the module to work as a Time Slave	
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_00087, SWS_EthTSyn_00096, SWS_EthTSyn_00111, SWS_EthTSyn_00153
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_00087, SWS_EthTSyn_00153
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 a static (pre)configuration of IEEE 802.1AS Pdelay	SWS_EthTSyn_00200, SWS_EthTSyn_00201
RS_TS_20069	The TimeSync over Ethernet module shall provide read / write access to bus protocol specific parameters	SWS_EthTSyn_00226, SWS_EthTSyn_00227
SRS_BSW_00101	The Basic Software Module shall be able to initialize variables and hardware in a separate initialization function	SWS_EthTSyn_00006, SWS_EthTSyn_00008
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
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
SRS_BSW_00385	List possible error notifications	SWS_EthTSyn_00030, SWS_EthTSyn_00144

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 5.1 General in [12].

7.1.2 VLAN Support

[SWS_EthTSyn_00148][
If the parameter `EthTSynFramePrio` (**ECUC_EthTSyn_00034 :**) exists, the `EthTSynGlobalTimeEthIfRef` (**ECUC_EthTSyn_00065 :**) 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` (**ECUC_EthTSyn_00034 :**) value as priority parameter when calling `EthIf_ProvideTxBuffer()`.
](RS_TS_20048)

Refer to chapter 5.2 VLAN Support in [12] 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 [13].

[SWS_EthTSyn_00008]⌈
When `EthTSyn_Init()` is called in initialized state, the EthTSyn module shall re-initialize its internal variables.
⌋(SRS_BSW_00101)

[SWS_EthTSyn_00010]⌈
When `EthTSyn_Init()` is called in initialized state, the EthTSyn module shall set each port-specific `Pdelay` value to 0.
⌋(RS_TS_20048)

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 6 and Figure 7).
- 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_GetCurrentTime()`,`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` (**ECUC_EthTSyn_00048 :**) is set to 0, EthTSyn shall ignore any debouncing.
⌋(RS_TS_20047)

[SWS_EthTSyn_00131]⌈

If EthTSynGlobalTimeDebounceTime (**ECUC_EthTSyn_00048 :**) 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][
If EthTSynGlobalTimeDebounceTime (**ECUC_EthTSyn_00048 :**) 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)

[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 5.6.1 “Pdelay Protocol for Latency Calculation” in [12].

[SWS_EthTSyn_00200][
If Master and Time Slave transmit Pdelay_Req for latency calculation with the cycle (refer to PRS_TS_00011 in [12]), 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 (**ECUC_EthTSyn_00018 :**) 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 [12]) 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` (**ECUC_EthTSyn_00018 :**) 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 according to Figure 6, if `EthTSynHardwareTimestampSupport` **(ECUC_EthTSyn_00018 :)** is set to `TRUE`.

If the StbM does not use the Ethernet controller as source for the Virtual Local Time (refer to parameter `StbMLocalTimeHardware`, **ECUC_StbM_00053**, in [4]), 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 according to Figure 6, if `EthTSynHardwareTimestampSupport` **(ECUC_EthTSyn_00018 :)** 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 EthIf via `EthIf_GetEgressTimeStamp()` on egress of the `Pdelay_Resp` message according to Figure 6, if `EthTSynHardwareTimestampSupport` **(ECUC_EthTSyn_00018 :)** is set to `TRUE`.

If the StbM does not use the Ethernet controller as source for the Virtual Local Time (refer to parameter `StbMLocalTimeHardware`, **ECUC_StbM_00053**, in [4]), 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 according to Figure 6 if `EthTSynHardwareTimestampSupport` **(ECUC_EthTSyn_00018 :)** 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_00034)

[SWS_EthTSyn_00014]⌈
If `EthTSynGlobalTimePdelayRespEnable` (**ECUC_EthTSyn_00069 :**) 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 [13] chapter 11.1.2 “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)

[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 according to Figure 7, if `EthTSynHardwareTimestampSupport` **(ECUC_EthTSyn_00018 :)** is set to `TRUE`.

If the StbM does not use the Ethernet controller as source for the Virtual Local Time (refer to parameter `StbMLocalTimeHardware`, **ECUC_StbM_00053**, in [4]), 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_GetCurrentVirtualLocalTime()` on ingress of `Pdelay_Req` message according to Figure 7, if `EthTSynHardwareTimestampSupport` **(ECUC_EthTSyn_00018 :)** 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_00034)

[SWS_EthTSyn_00049]⌈

On invocation of `EthTSyn_RxIndication()` the ingress time stamp shall be retrieved for **t4** from the EthIf via `EthIf_GetIngressTimeStamp()` on ingress of the Pdelay_Resp message according to Figure 7, if `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) is set to `TRUE`.

If the StbM does not use the Ethernet controller as source for the Virtual Local Time (refer to parameter `StbMLocalTimeHardware`, **ECUC_StbM_00053**, in [4]), 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()` according to Figure 5, if `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) is set to `FALSE`.
](RS_TS_20048)

7.6 Message Format

Refer to chapter 5.3 Message format in [12] for additional requirements.

7.6.1 `Sync` and `Follow_Up` acc. to IEEE 802.1AS

Refer to chapter 5.3.1.1 `Sync` and `Follow_Up` acc. to IEEE 802.1AS in [12].

7.6.2 `Sync` and `Follow_Up` acc. to AUTOSAR

Refer to chapter 5.3.1.2 `Sync` and `Follow_Up` acc. to AUTOSAR in [12].

7.6.2.1 `Follow_Up` Message Header [AUTOSAR]

Refer to chapter 5.3.1.3 `Follow_Up` Message Header [AUTOSAR] in [12].

7.6.2.2 AUTOSAR TLV *Sub-TLV's*

Refer to chapter 5.3.1.5 AUTOSAR *TLV* Sub-TLVs in [12].

7.6.2.2.1 AUTOSAR TLV *Sub-TLV*: Time Secured

Refer to chapter 5.3.1.6 AUTOSAR *TLV Sub-TLV*: Time Secured in [12].

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

Refer to chapter 5.3.1.7 AUTOSAR *TLV Sub-TLV*: Status Secured in [12].

7.6.2.2.3 AUTOSAR TLV *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.userDataLength` and vice versa.
](RS_TS_20061, RS_TS_20062)

[SWS_EthTSyn_00153][
If `StbM_UserDataType.userDataLength` is set to 0 the complete AUTOSAR *Sub-TLV*: UserData shall be excluded from the message.
](RS_TS_20061, RS_TS_20062)

Refer to chapter 5.3.1.8 AUTOSAR *TLV Sub-TLV*: UserData Secured / Not Secured in [12] for additional requirements.

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

[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 TLV 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.userDataLength` and vice versa.
](RS_TS_20061, RS_TS_20062)

Refer to chapter 5.3.1.9 AUTOSAR *TLV Sub-TLV*: OFS Secured / Not Secured in [12] for additional requirements.

7.7 Acting as Time Master

Refer to chapter 5.6.2 Acting as Time Master in [12] for additional requirements.

7.7.1 Message processing

Refer to chapter 5.6.3.1 Message Processing in [12] for additional requirements.

[SWS_EthTSyn_00202][
If the Time Master transmits a `Sync` message (refer to [PRS_TS_00016] in [12]), the following sequence shall be applied:

1. The Global Time Tuple [T0;$T0_{VLT}$] shall be retrieved from the StbM via `StbM_BusGetCurrentTime()` according to Figure 6
2. Get a free transmission buffer via `EthIf_ProvideTxBuffer()`
3. Activate the time stamping via `EthIf_EnableEgressTimeStamp()` if `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) is set to `TRUE`
4. Trigger transmit request via `EthIf_Transmit()`

](RS_TS_20047, RS_TS_20048)

Note: The `timeBaseStatus` can be read from StbM by `StbM_GetTimeBaseStatus()`, `StbM_BusGetCurrentTime()` or `StbM_GetCurrentTime()`.

[SWS_EthTSyn_00211][
The Time Master shall start cyclic transmission of Sync messages in the earliest possible `EthTSyn_MainFunction()` call once the protocol requirement [PRS_TS_00016] is fulfilled.
](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 Figure 6, if `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) is set to `TRUE`.
](RS_TS_20048)

[SWS_EthTSyn_00017][
If `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) 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 `EthIf_GetCurrentTime()` 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` (**ECUC_EthTSyn_00018 :**) 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 $T0 – (T0_{VLT} – T2_{VLT})$.
](RS_TS_20048)

[SWS_EthTSyn_00189][
If `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) is set to `FALSE` T0 shall be used as value for the `preciseOriginTimestamp`.
](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 [12]), 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_00017]** via `EthIf_Transmit()`

](RS_TS_20048)

7.7.1.1 Runtime Error detection

[SWS_EthTSyn_00145][
If `EthTSynMasterSlaveConflictDetection` (**ECUC_EthTSyn_00075 :**) 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(ETHTSYN_E_TMCONFLICT)` and discard the received `Sync` message.
](RS_TS_20051)

7.7.1.2 Frame Debouncing

Refer to chapter 5.6.2.1.1 Frame Debouncing in [12].

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.

[SWS_EthTSyn_00134][
If `EthTSynImmediateTimeSync` (**ECUC_EthTSyn_00046 :**) 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` (**ECUC_EthTSyn_00046 :**) 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(),StbM_BusGetCurrentTime()` or `StbM_GetCurrentTime()`.

The `debounceCounter` as described in 7.4 has always to be considered.

[SWS_EthTSyn_00136][

If EthTSynImmediateTimeSync (**ECUC_EthTSyn_00046 :**) is set to TRUE, EthTSynCyclicMsgResumeTime (**ECUC_EthTSyn_00047 :**) shall be considered.
](RS_TS_20047)

[SWS_EthTSyn_00137][
EthTSynCyclicMsgResumeTime (**ECUC_EthTSyn_00047 :**) represents the timeout value of a cyclicMsgResumeCounter that shall be started when a Sync has been sent immediately, asynchronous to the cyclic transmission. The cyclicMsgResumeCounter shall be decremented on each invocation of EthTSyn_MainFunction() if no Timesync PDU is transmitted asynchronously.
](RS_TS_20047)

[SWS_EthTSyn_00139][
If the cyclicMsgResumeCounter has reached a value equal or less than 0, EthTSyn shall resume cyclic Timesync message transmission by sending a Sync.
](RS_TS_20047)

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 5.6.2.2 Message Field Calculation and Assembling in [12] for additional requirements.

7.7.3.1 SGW Calculation

Refer to chapter 5.6.2.2.1 SGW Calculation in [12].

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 5.6.2.2.2 OFS Calculation in [12] for additional requirements.

7.7.3.3 CRC Calculation

Refer to chapter 5.6.2.2.3 CRC Calculation in [12] for additional requirements.

[SWS_EthTSyn_00096][
The function `Crc_CalculateCRC8H2F()` as defined in [10] shall be used to calculate the *CRC* if configured.
](RS_TS_20061)

7.7.3.3.1 AUTOSAR TLV *Sub-TLV*: Time Secured

Refer to chapter 5.6.2.2.3.1 AUTOSAR TLV Sub-TLV: Time Secured in [12].

7.7.3.3.2 AUTOSAR TLV *Sub-TLV*: Status secured

Refer to chapter 5.6.2.2.3.2 AUTOSAR TLV Sub-TLV: Status secured in [12].

7.7.3.3.3 AUTOSAR TLV *Sub-TLV*: UserData secured

Refer to chapter 5.6.2.2.3.3 AUTOSAR TLV Sub-TLV: UserData secured in [12].

7.7.3.3.4 AUTOSAR TLV *Sub-TLV*: OFS secured

Refer to chapter 5.6.2.2.3.4 AUTOSAR TLV Sub-TLV: OFS secured in [12].

7.7.3.4 Sequence Counter Calculation

Refer to chapter 5.6.2.2.4: Sequence Counter (`sequenceId`) Calculation in [12] for additional requirements.

7.7.3.5 Message Assembling

[SWS_EthTSyn_00104][
For each transmission of a Time Synchronization message, the EthTSyn module shall assemble the message as follows:

1. If `Sync`: Calculate Message Header
2. If `Follow_Up`: Calculate `Follow_Up.preciseOriginTimestamp` and Message Header inclusive `correctionField`
3. If `Follow_Up`: Calculate IEEE *TLV*
4. If `Follow_Up`: Calculate AUTOSAR *TLV* (configuration dependent)
    a. Calculate *CRC* (configuration dependent)
5. Copy all data to the appropriate position within the related message

](RS_TS_20048)

7.8 Acting as Time Slave

Refer to chapter 5.6.3 Acting as Time Slave in [12] for additional requirements.

7.8.1 Message processing

Additional content to this chapter can be found in [12] in chapter 5.6.3.1 Message Processing.

[SWS_EthTSyn_00128][
On invocation of `EthTSyn_RxIndication` the ingress time stamp shall be retrieved for `Sync` via `EthIf_GetIngressTimeStamp()` from the EthIf and converted to the Virtual Local Time $T1_{VLT}$ according to Figure 7, if `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) is set to `TRUE`.
](RS_TS_20048)

[SWS_EthTSyn_00180][
On invocation of `EthTSyn_RxIndication` and if `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) is set to `FALSE` the following sequence shall be applied:

1. Immediately establish a protection against interruptions and run the next step directly afterwards:
2. Retrieve the reference time $T1_{VLT}$ for the `Sync` message via `StbM_GetCurrentVirtualLocalTime()` from the StbM
3. The protection against interruptions may be removed now

](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_00024]⌈
On invocation of `EthTSyn_RxIndication()` or in the following `EthTSyn_MainFunction()` call, a reference time shall be retrieved on reception of the `Follow_Up` message via `EthIf_GetCurrentTime()` from the EthIf and converted to the Virtual Local Time $T2_{VLT}$, if `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) is set to `TRUE` and if the StbM does use the Ethernet hardware counter as Virtual Local Time Source for the Time Base.
The `Sync` reception delay shall be calculated as $T2_{VLT} – T1_{VLT}$.
⌋(RS_TS_20048)

[SWS_EthTSyn_00190]⌈
On invocation of `EthTSyn_RxIndication()` or in the following `EthTSyn_MainFunction()` call, a reference time shall be retrieved on reception of the `Follow_Up` message if `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) 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:

1. Protect the following two steps against interruptions:
2. the current time of the Ethernet hardware counter shall be retrieved via `EthIf_GetCurrentTime()` 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 $T2_{VLT}$ via `StbM_GetCurrentVirtualLocalTime()`
4. the sync reception delay shall be calculated as $T3_{VLT} – T1_{VLT}$

⌋(RS_TS_20048)

[SWS_EthTSyn_00179]⌈
On invocation of `EthTSyn_RxIndication()` or in the following `EthTSyn_MainFunction()` call, the reference time $T2_{VLT}$ shall be retrieved on reception of the `Follow_Up` message via `StbM_GetCurrentVirtualLocalTime()` from the StbM, if `EthTSynHardwareTimestampSupport` (**ECUC_EthTSyn_00018 :**) is set to `FALSE`.
The `Sync` reception delay shall be calculated as $T2_{VLT} – T1_{VLT}$.
⌋(RS_TS_20048)

[SWS_EthTSyn_00052]⌈
For a valid `Follow_Up` message a new Time Tuple [T2;$T2_{VLT}$] shall be calculated and forwarded to the StbM module via `StbM_BusSetGlobalTime()`, according to Figure 7, where T2 is the sum of:

- `preciseOriginTimestamp`,
- `correctionField`,

- Pdelay and
- the Sync reception delay.

](RS_TS_20048)

Note: The Pdelay value is not influenced significantly by a RateRatio acc to [13] Note-2 of chapter 11.2.15.2.4 “computePropTime():”.

[SWS_EthTSyn_00150][
On an invocation of StbM_BusSetGlobalTime() the current Pdelay value shall be passed by the parameter measureDataPtr->PathDelay.
](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 timeBaseStatus (structure member, which is referenced by the parameter timeStampPtr), according to the *SGW* value (refer to [PRS_TS_00156]). The remaining status bits shall be set to 0.
](RS_TS_20051)

[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 (**ECUC_EthTSyn_00075 :**) 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(ETHTSYN_E_TSCONFLICT) and discard the received Sync frame.
](RS_TS_20051)

7.8.2 Message Field Validation and Disassembling

Additional content to this chapter can be found in [12] in chapter 5.6.3.2 Message Field Validation and Disassembling.

7.8.2.1 SGW Calculation

Refer to chapter 5.6.3.2.1 SGW Calculation in [12].

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 5.6.3.2.2 OFS Calculation in [12] for additional requirements.

7.8.2.3 CRC Validation

[SWS_EthTSyn_00111][
The function `Crc_CalculateCRC8H2F()` as defined in [10] shall be used to calculate the *CRC* if configured.
](RS_TS_20061)

Refer to chapter 5.6.3.2.3 CRC Calculation in [12] for additional requirements.

7.8.2.3.1 AUTOSAR TLV *Sub-TLV*: Time Secured

Refer to chapter 5.6.3.2.3.1 AUTOSAR TLV Sub-TLV: Time Secured in [12].

7.8.2.3.2 AUTOSAR TLV *Sub-TLV*: Status secured

Refer to chapter 5.6.3.2.3.2 AUTOSAR TLV Sub-TLV: Status secured in [12].

7.8.2.3.3 AUTOSAR TLV *Sub-TLV*: UserData secured

Refer to chapter 5.6.3.2.3.3 AUTOSAR TLV Sub-TLV: UserData secured in [12].

7.8.2.3.4 AUTOSAR TLV *Sub-TLV*: OFS secured

Refer to chapter 5.6.3.2.3.4 AUTOSAR TLV Sub-TLV: OFS secured in [12].

7.8.2.4 Sequence Counter Calculation

Refer to chapter 5.6.3.2.4: Sequence Counter (`sequenceId`) Calculation in [12] for additional requirements.

7.8.2.5 Message Disassembling

Refer to chapter 5.6.3.2.5 Message Disassembling in [12].

7.9 Time Recording

7.9.1 Time Validation

[SWS_EthTSyn_00212]⌈
The EthTSyn shall support Time Validation, if `EthTSynTimeValidationSupport` (**ECUC_EthTSyn_00081**) 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

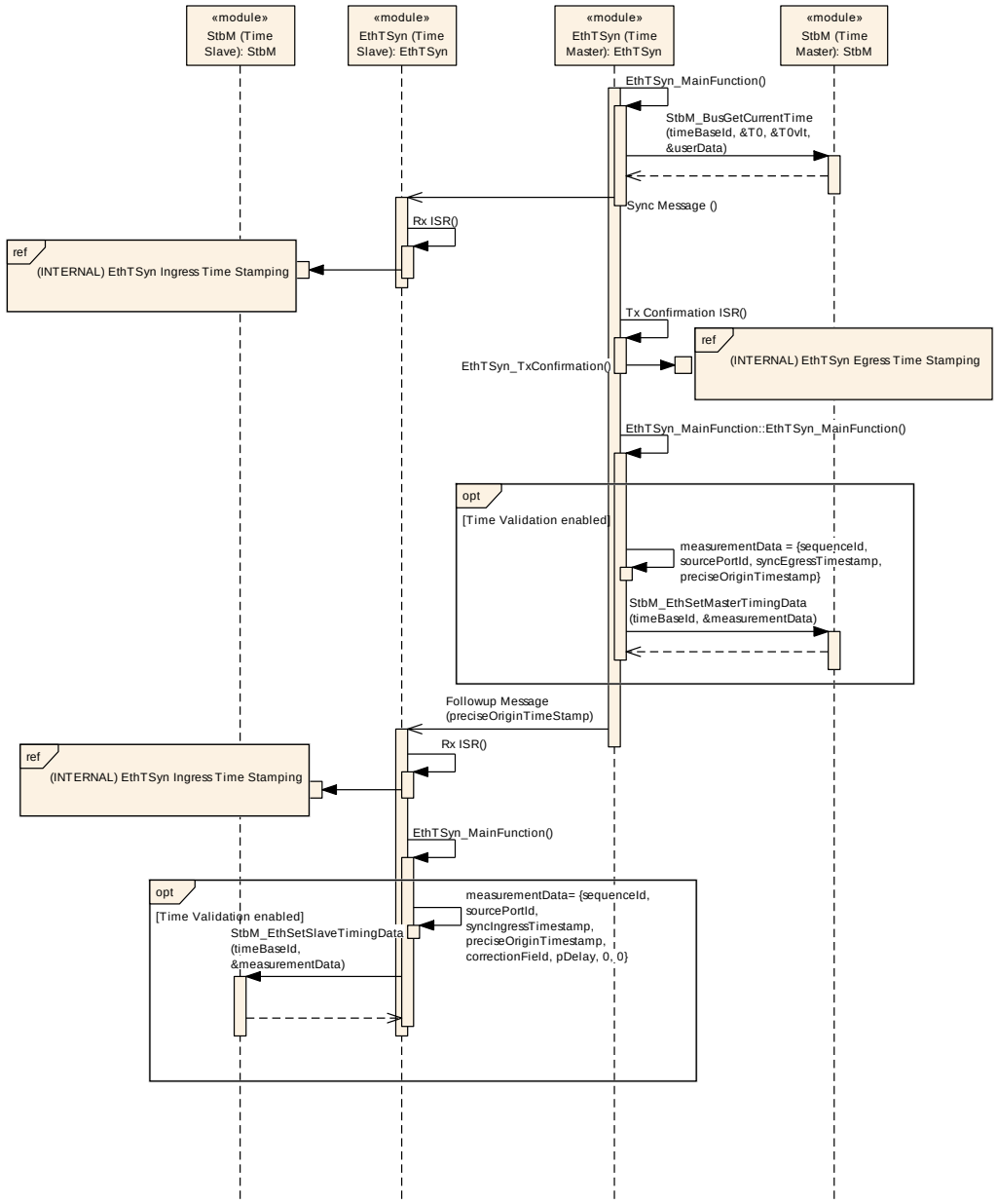


Figure 4).
](RS_TS_20048)

[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 `FollowUp` message and (refer to **[SWS_EthTSyn_00188]**)
- the `correctionField` as copied to the `FollowUp` 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 `Follow_Up` message (refer to

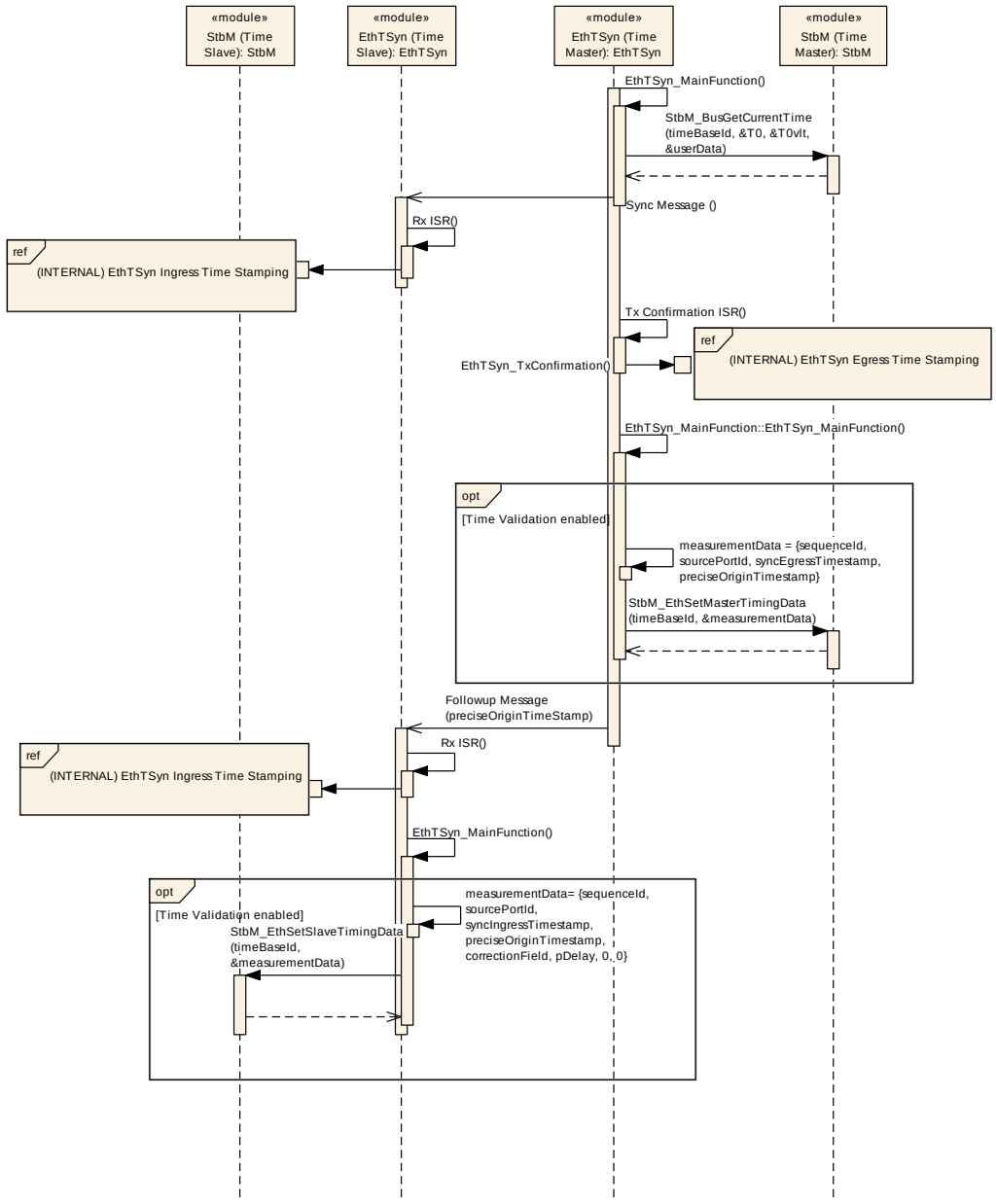


Figure 4).

`StbM_EthSetSlaveTimingData()` shall be called after `StbM_BusSetGlobalTime()`.

⌋(RS_TS_00034)

Note: `StbM_BusSetGlobalTime()` shall be called first, because it updates the Synclocal Time Tuple (refer to [6]), 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 `FollowUp` message
- the `correctionField` received in the `FollowUp` 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 [6]).

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_BusGetCurrentTime()` to retrieve a Time Tuple $[T_{\_refPDResponder}; T_{VLT\_refPDResponder}]$ before sending the `pDelay_Resp` message (refer to

Figure 5).

⌋(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.

[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

Figure 5).

⌋(RS_TS_00034)

[SWS_EthTSyn_00220]{DRAFT}⌈
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 `measureDataPtr`.

⌋(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_BusGetCurrentTime()` to retrieve a Time Tuple [$T_{\_refPdInitiator}$; $T_{VLT\_refPDInitiator}$] before sending the `pDelay_Req` message (refer to Figure 5).

⌋(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.

[SWS_EthTSyn_00221]{DRAFT}⌈
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

Figure 5).

⌋(RS_TS_00034)

[SWS_EthTSyn_00222]{DRAFT}⌈

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 Time measurement with Switches

Refer to chapter 5.7 Time Measurement with Switches in [12].

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 (**ECUC_EthTSyn_00002 :**)).

[SWS_EthTSyn_00030]⌈

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)

7.11.2 Runtime Errors

[SWS_EthTSyn_00144]⌈

Type of error	*Related error code*	*Error value*
Time Master conflict	ETHTSYN_E_TMCONFLICT	0x01
Time Slave conflict	ETHTSYN_E_TSCONFLICT	0x02

⌋(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][

Module	*Header File*	*Imported Type*
ComStack_Types	ComStack_Types.h	BufReq_ReturnType
Eth	Eth_GeneralTypes.h	Eth_BufIdxType
	Eth_GeneralTypes.h	Eth_DataType
	Eth_GeneralTypes.h	Eth_FrameType
	Eth_GeneralTypes.h	Eth_TimeStampQualType
	Eth_GeneralTypes.h	Eth_TimeStampType
EthSwt	Eth_GeneralTypes.h	EthSwt_MgmtInfoType
EthTrcv	Eth_GeneralTypes.h	EthTrcv_LinkStateType
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_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][

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][

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][

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][

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][

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)

8.1.3.4 EthTSyn_SetProtocolParam

[SWS_EthTSyn_91001][

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 ..15 in EthTSyn (Development Error: `ETHTSYN_E_PARAM`)
- `protocolParam` is NULL (Development Error: `ETHTSYN_E_PARAM_POINTER`)

](SRS_BSW_00323, SRS_BSW_00337)

8.1.3.5 EthTSyn_GetProtocolParam

[SWS_EthTSyn_91002][

Service Name	EthTSyn_GetProtocolParam

Syntax	`Std_ReturnType EthTSyn_GetProtocolParam (`
	`  StbM_SynchronizedTimeBaseType timeBaseId,`
	`  StbM_ProtocolParamType* protocolParam`
	`)`

Service ID [hex]	0xb

Sync/Async	Synchronous	
Reentrancy	Non 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 ..15 in EthTSyn (Development Error: `ETHTSYN_E_PARAM`)
- `protocolParam` is NULL (Development Error: `ETHTSYN_E_PARAM_POINTER`)

](SRS_BSW_00323, SRS_BSW_00337)

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][

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
	Is Broadcast	parameter to indicate a broadcast frame
	PhysAddr Ptr	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][

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)	Ctrl Idx	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][

Service Name	EthTSyn_TrcvLinkStateChg
Syntax	`Std_ReturnType 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	Std_ReturnType	E_OK: successful E_NOT_OK: failed
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)

8.1.5 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][

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][

API Function	*Header File*	*Description*
Crc_Calculate-CRC8H2F	Crc.h	This service makes a CRC8 calculation with the Polynomial 0x2F on Crc_Length
Det_Report-Error	Det.h	Service to report development errors.
Det_Report-RuntimeError	Det.h	Service to report runtime errors. If a callout has been configured then this callout shall be called.
EthIf_Enable-EgressTime-Stamp	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_Get-CurrentTime	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.
EthIf_Get-EgressTime-Stamp	EthIf.h	Reads back the egress time stamp on a dedicated message object. It must be called within the TxConfirmation() function.
EthIf_Get-IngressTime-Stamp	EthIf.h	Reads back the ingress time stamp on a dedicated message object. It must be called within the RxIndication() function.
EthIf_Provide-TxBuffer	EthIf.h	Provides access to a transmit buffer of the specified Ethernet controller.
EthIf_Set-SwitchMgmtInfo	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_Switch-EnableTime-Stamping	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
StbM_BusGet-CurrentTime	StbM.h	Returns the current Time Tuple, status and User Data of the Time Base.
StbM_BusSet-GlobalTime	StbM.h	Allows the Time Base Provider Modules to forward a new Global Time tuple (i.e., the Received Time Tuple) to the StbM.
StbM_EthSet-MasterTiming-Data	StbM_ Eth TSyn.h	Provides Ethernet Timesyn module specific data for a Time Master to the StbM. **Tags:**atp.Status=draft
StbM_EthSet-PdelayInitiator-Data	StbM_ Eth TSyn.h	-- **Tags:**atp.Status=draft
StbM_EthSet-Pdelay-ResponderData	StbM_ Eth TSyn.h	-- **Tags:**atp.Status=draft
StbM_EthSet-SlaveTiming-Data	StbM_ Eth TSyn.h	Allows the EthTSyn Module to forward Ethernet specific details to the StbM. **Tags:**atp.Status=draft
StbM_Get-CurrentTime	StbM.h	Returns a time value (Local Time Base derived from Global Time Base) in standard format. 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_Get-CurrentVirtual-LocalTime	StbM.h	Returns the Virtual Local Time of the referenced Time Base.
StbM_Get-Offset	StbM.h	Allows the Timesync Modules to get the current Offset Time and User Data.
StbM_GetTime-BaseStatus	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_GetTime-BaseUpdate-Counter	StbM.h	Allows the Timesync Modules to detect, whether a Time Base should be transmitted immediately in the subsequent <Bus>TSyn_Main Function() cycle.

](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

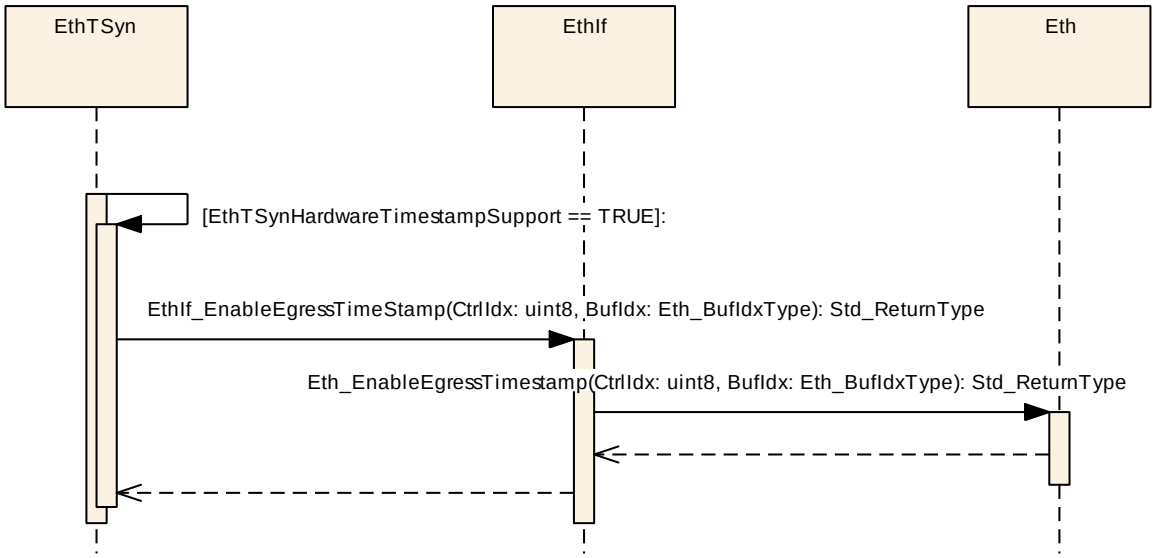


Figure 3: EthIf_EnableEgressTimeStamp

9.2 Time Synchronization Sequence

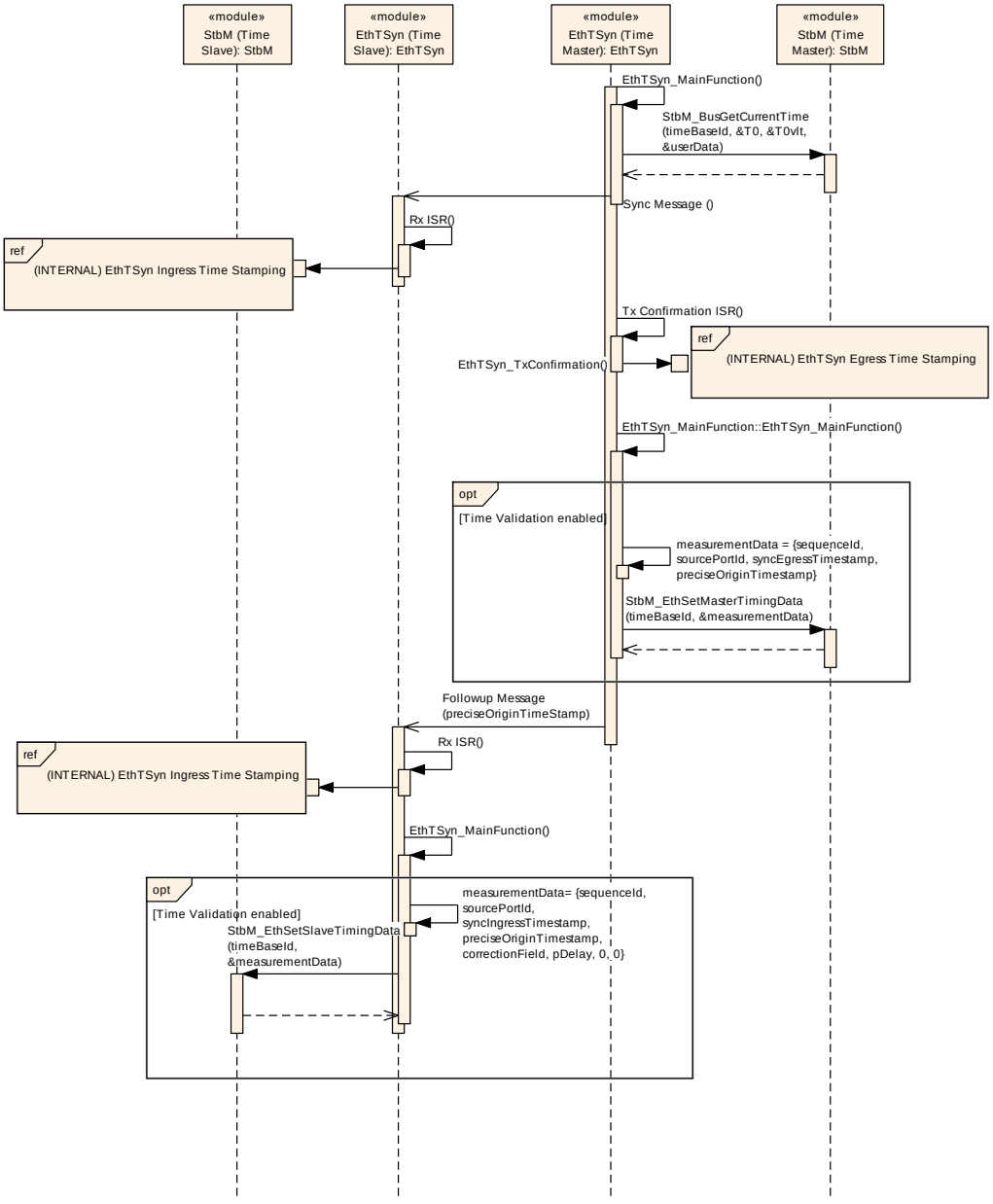


Figure 4 : Time Synchronization Sequence

9.3 Pdelay Measurement Sequence

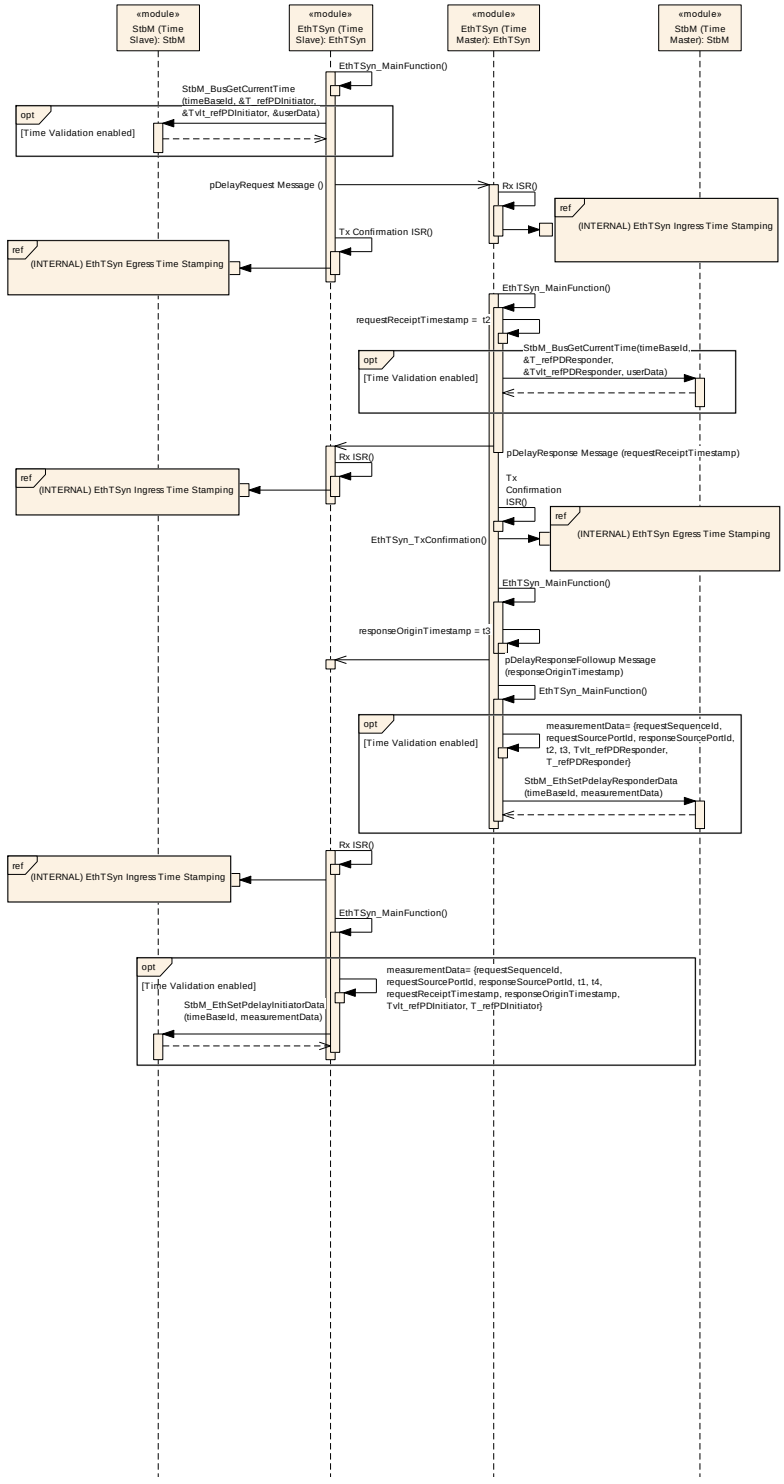


Figure 5 : Pdelay Sequence

9.4 EthTSyn Egress Timestamping

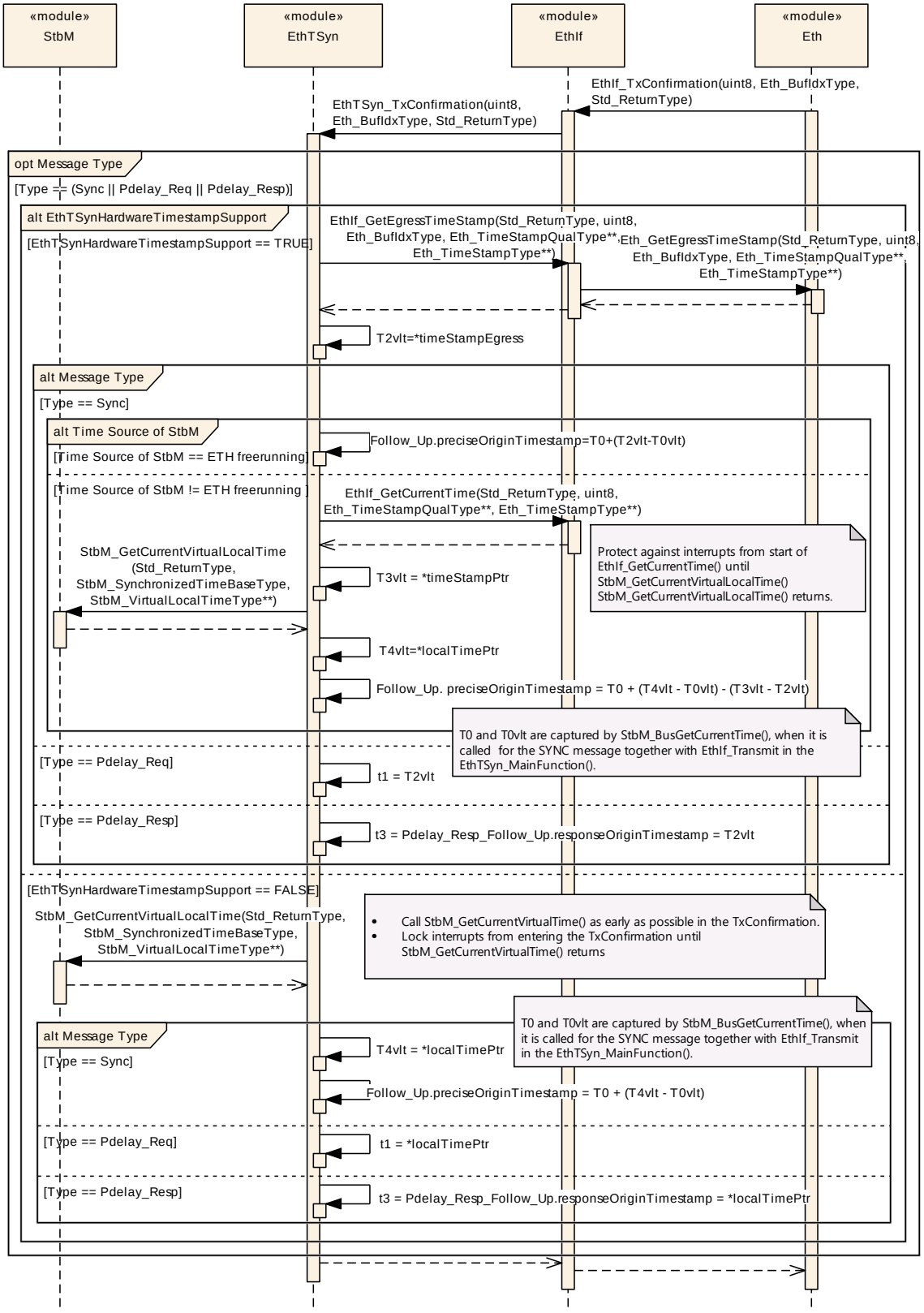


Figure 6: EthTSyn Egress Timestamping

9.5 EthTSyn Ingress Timestamping

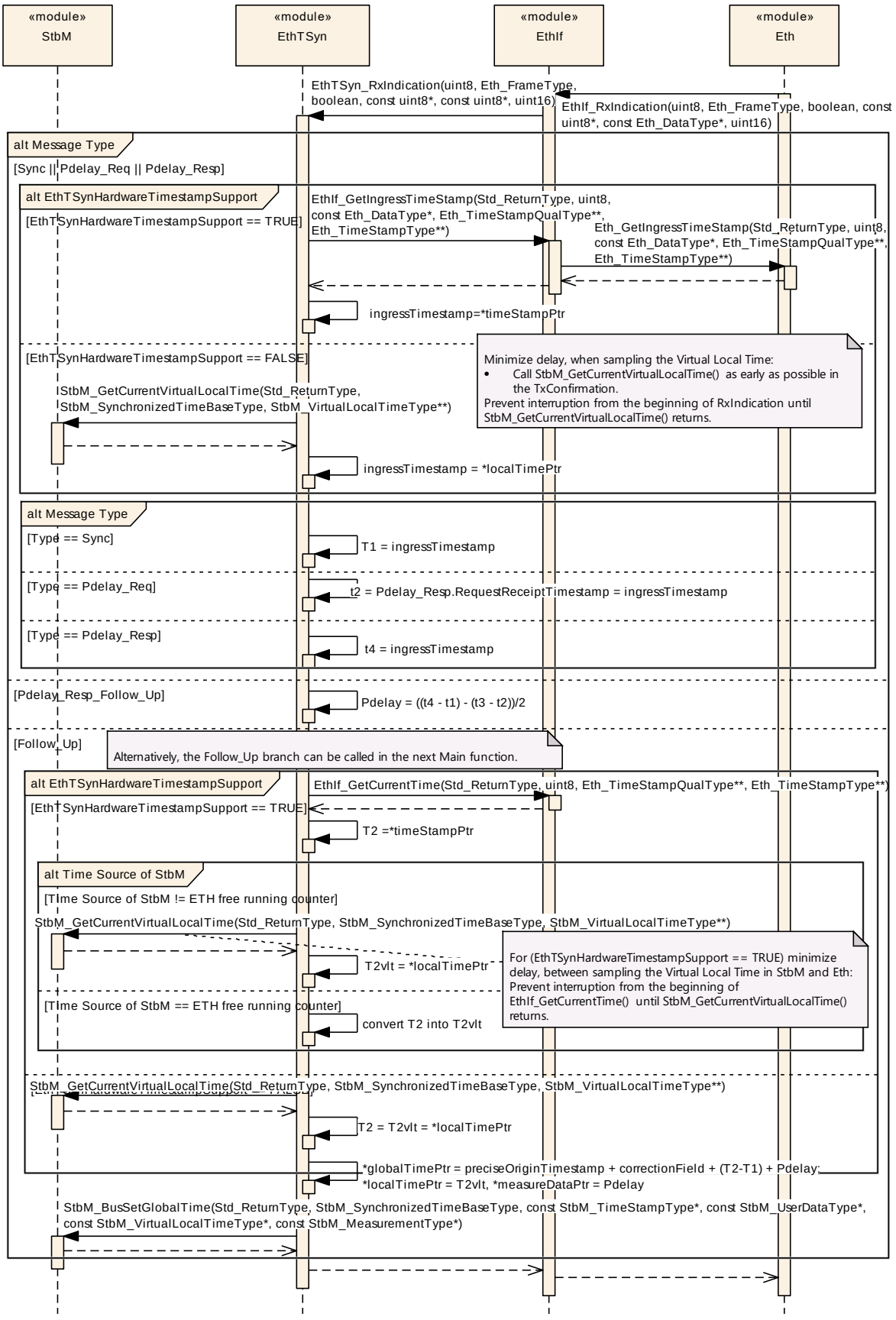

«module»
StbM
«module»
EthTSyn
«module»
EthIf
«module»
Eth
EthTSyn_RxIndication(uint8, Eth_FrameType, boolean, const uint8*, const uint8*, uint16)
EthIf_RxIndication(uint8, Eth_FrameType, boolean, const uint8*, const Eth_DataType*, uint16)
alt Message Type
[Sync || Pdelay_Req || Pdelay_Resp]
alt EthTSynHardwareTimestampSupport
[EthTSynHardwareTimestampSupport == TRUE]
EthIf_GetIngressTimeStamp(Std_ReturnType, uint8, const Eth_DataType*, Eth_TimeStampQualType**, Eth_TimeStampType**)
Eth_GetIngressTimeStamp(Std_ReturnType, uint8, const Eth_DataType*, Eth_TimeStampQualType**, Eth_TimeStampType**)
ingressTimestamp=*timeStampPtr
[EthTSynHardwareTimestampSupport == FALSE]
Minimize delay, when sampling the Virtual Local Time:
Call StbM_GetCurrentVirtualLocalTime() as early as possible in the TxConfirmation.
Prevent interruption from the beginning of RxIndication until StbM_GetCurrentVirtualLocalTime() returns.
StbM_GetCurrentVirtualLocalTime(Std_ReturnType, StbM_SynchronizedTimeBaseType, StbM_VirtualLocalTimeType**)
ingressTimestamp = *localTimePtr
alt Message Type
[Type == Sync]
T1 = ingressTimestamp
[Type == Pdelay_Req]
t2 = Pdelay_Resp.RequestReceiptTimestamp = ingressTimestamp
[Type == Pdelay_Resp]
t4 = ingressTimestamp
[Pdelay_Resp_Follow_Up]
Pdelay = ((t4 - t1) - (t3 - t2))/2
[Follow_Up]
Alternatively, the Follow_Up branch can be called in the next Main function.
alt EthTSynHardwareTimestampSupport
EthIf_GetCurrentTime(Std_ReturnType, uint8, Eth_TimeStampQualType**, Eth_TimeStampType**)
[EthTSynHardwareTimestampSupport == TRUE]
T2 =*timeStampPtr
alt Time Source of StbM
[Time Source of StbM != ETH free running counter]
StbM_GetCurrentVirtualLocalTime(Std_ReturnType, StbM_SynchronizedTimeBaseType, StbM_VirtualLocalTimeType**)
T2vlt = *localTimePtr
For (EthTSynHardwareTimestampSupport == TRUE) minimize delay, between sampling the Virtual Local Time in StbM and Eth:
Prevent interruption from the beginning of EthIf_GetCurrentTime() until StbM_GetCurrentVirtualLocalTime() returns.
[Time Source of StbM == ETH free running counter]
convert T2 into T2vlt
StbM_GetCurrentVirtualLocalTime(Std_ReturnType, StbM_SynchronizedTimeBaseType, StbM_VirtualLocalTimeType**)
T2 = T2vlt = *localTimePtr
*globalTimePtr = preciseOriginTimestamp + correctionField + (T2-T1) + Pdelay;
*localTimePtr = T2vlt, *measureDataPtr = Pdelay
StbM_BusSetGlobalTime(Std_ReturnType, StbM_SynchronizedTimeBaseType, const StbM_TimeStampType*, const StbM_UserDataType*, const StbM_VirtualLocalTimeType*, const StbM_MeasurementType*)

Figure 7: EthTSyn Ingress Timestamping

9.6 Time measurement with Switches

9.6.1 Time Aware Bridge with GTM as Management CPU – Tx

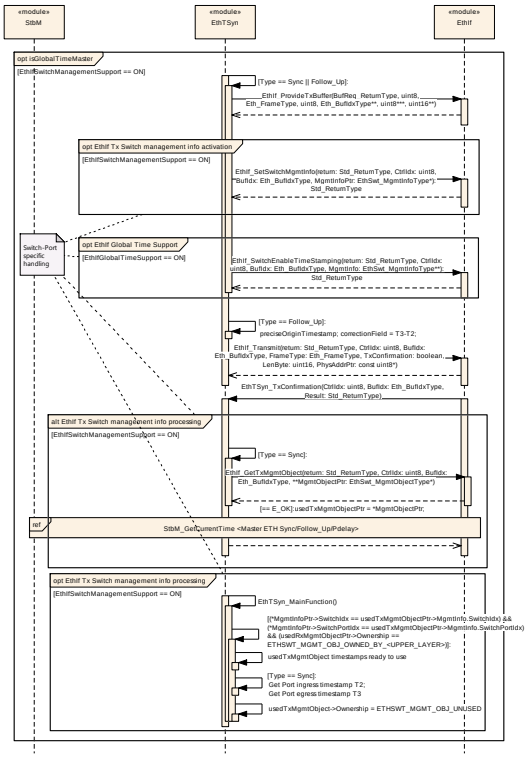


Figure 8: Time Aware Bridge with GTM as Management CPU [`Sync`/`Follow_Up` Tx]

9.6.2 Time Aware Bridge without GTM as Management CPU – Tx

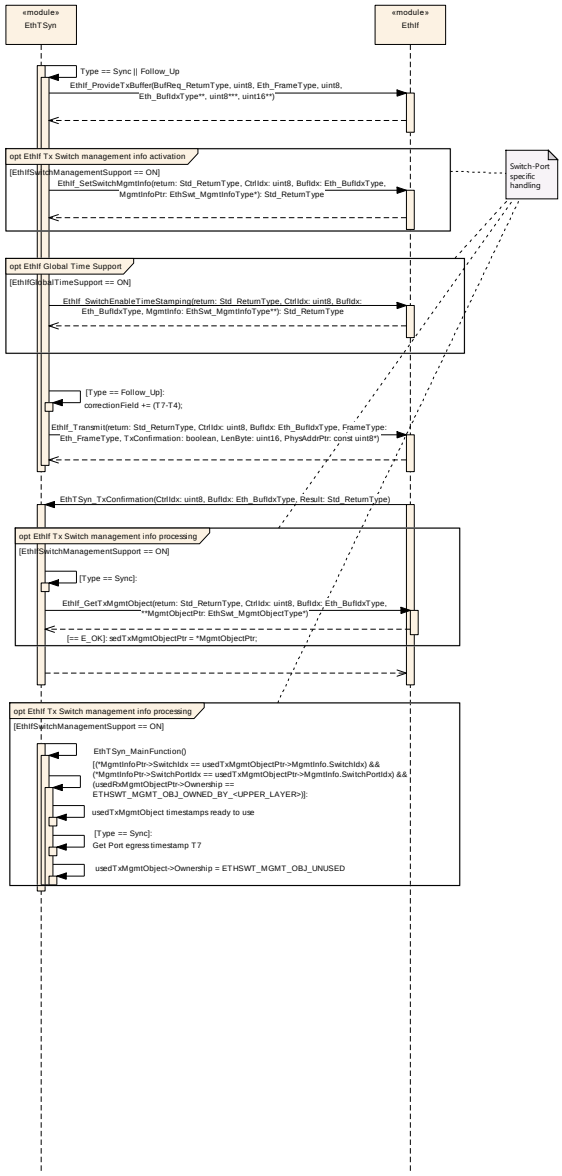


Figure 9: Time Aware Bridge without GTM as Management CPU [`Sync`/`Follow_Up` Tx]

9.6.3 Time Aware Bridge without GTM as Management CPU – Rx

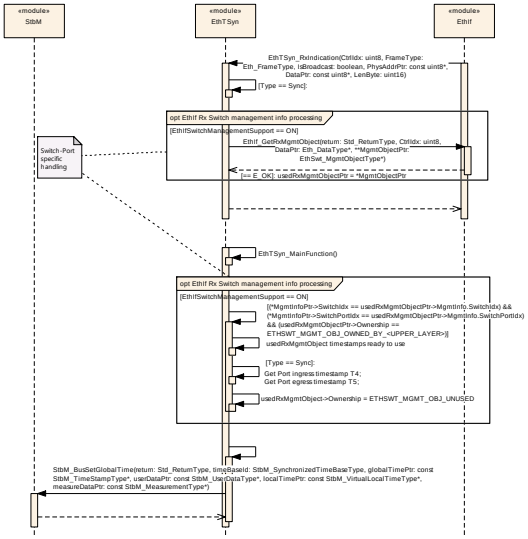


Figure 10: Time Aware Bridge without GTM as Management CPU [`Sync`/`Follow_Up` Rx]

10 Configuration specification

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

Section 10.2 specifies the structure (containers) and the parameters of the Global Time Synchronization over Ethernet.

Section 10.3 specifies published information of the Global Time Synchronization over Ethernet.

10.1 How to read this chapter

For details refer to the chapter 10.1 “Introduction to configuration specification” in [5].

[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

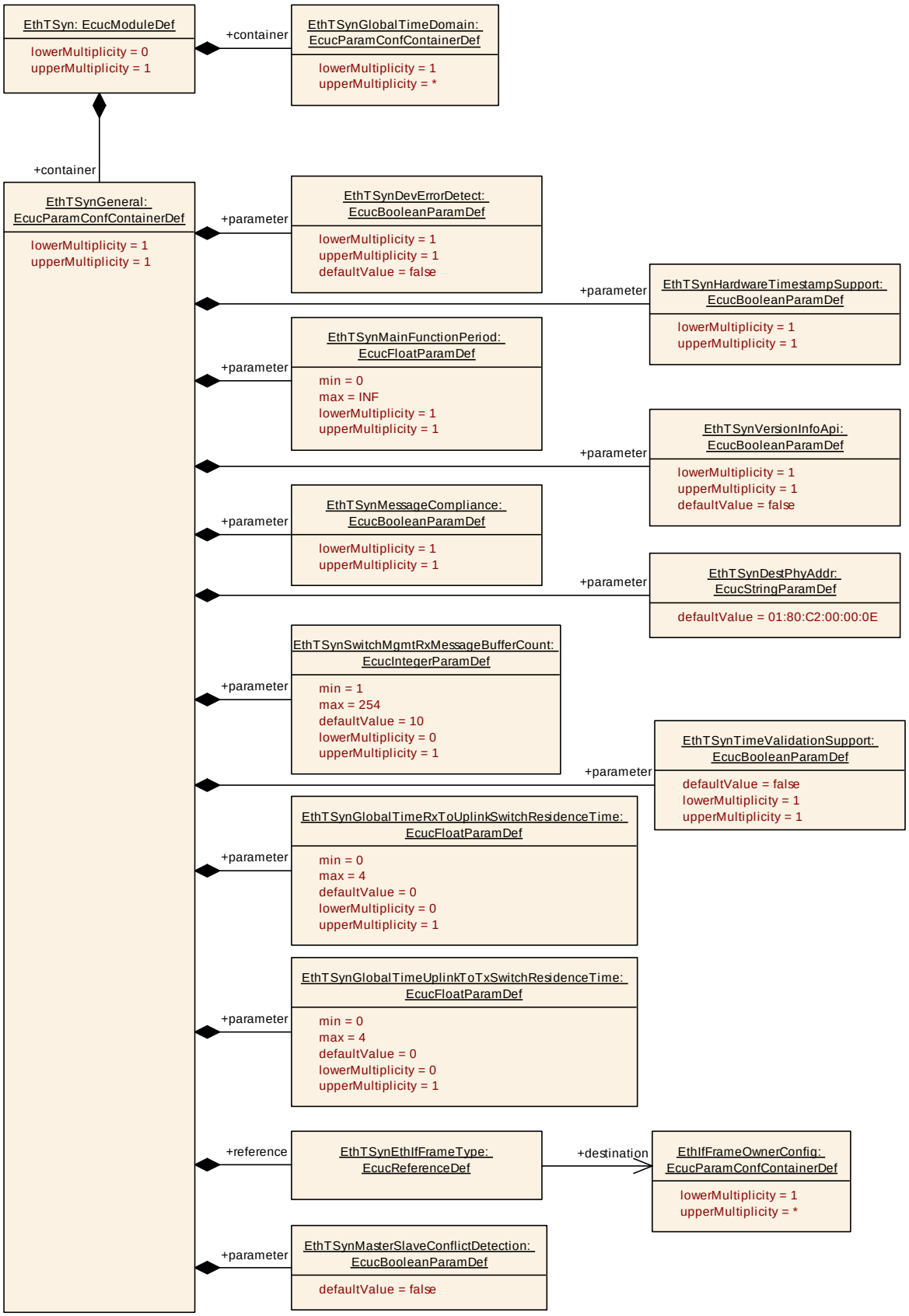
The following sections summarize all configuration parameters of the Global Time Synchronization over Ethernet. The detailed meaning of the parameters is described in chapters 7 and 8.
The module supports different post-build variants (previously known as post-build selectable configuration sets), but not post-build loadable configuration.

10.2.1 EthTSyn

SWS Item	**ECUC_EthTSyn_00001 :**
Module Name	*EthTSyn*
Module 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.



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 :**		
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		
maxLength	--		
minLength	--		
regularExpression	--		
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 :**		
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_00060 :**
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 :**		
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 :**		
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 :**
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 :**		
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 :**		
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 :**		
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	false		

Multiplicity			
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 :**		
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 :**		
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 :**		
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		

No Included Containers

10.2.3 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 :**		
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 :**		
Name	EthTSynGlobalTimeDebounceTime		
Parent Container	EthTSynGlobalTimeDomain		
Description	This represents the configuration of a TX debounce time for Sync, Follow_Up, and pDelay 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 :**		
Name	EthTSynGlobalTimeDomainId		
Parent Container	EthTSynGlobalTimeDomain		
Description	The global time domain ID.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 31		
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		

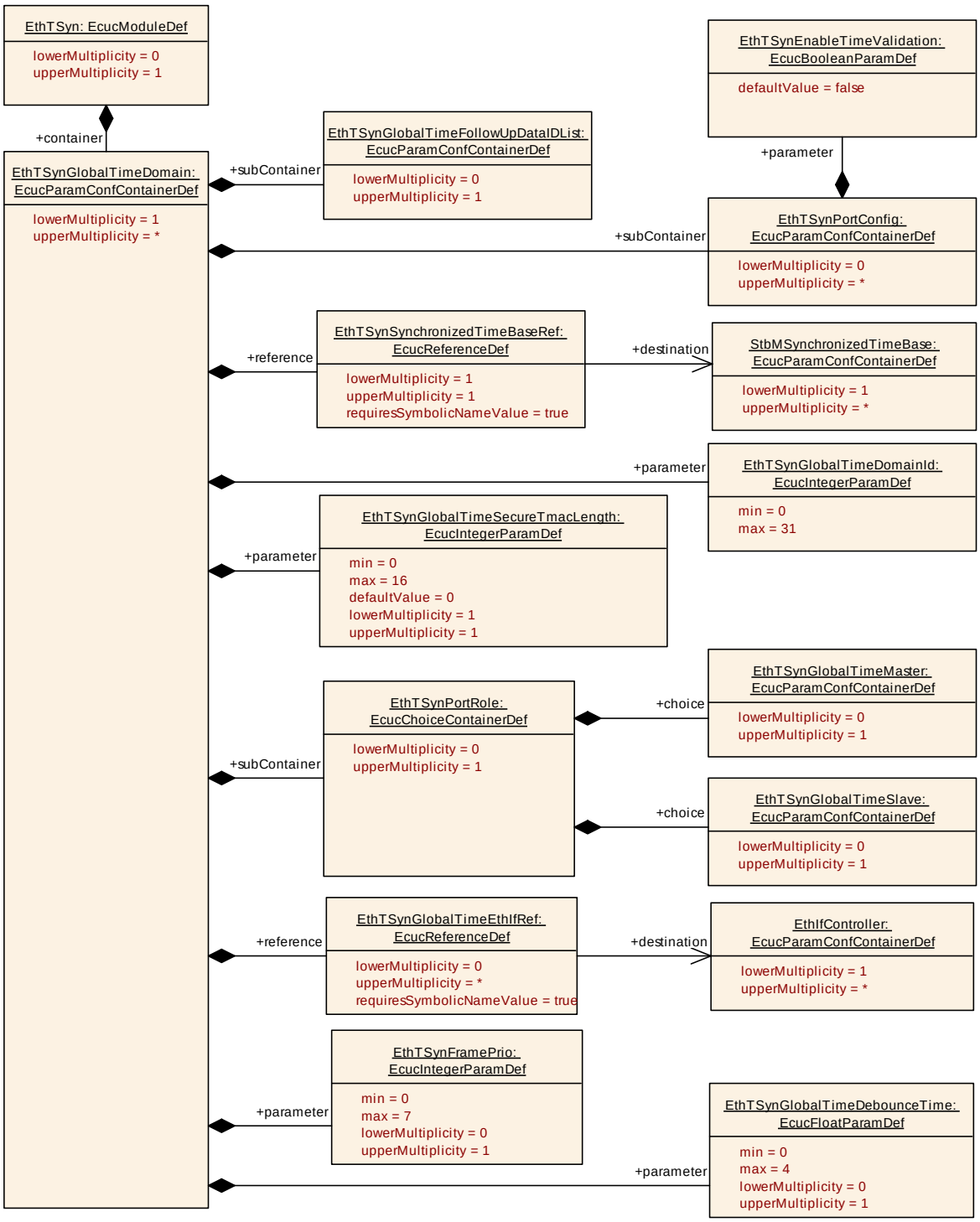
SWS Item	**ECUC_EthTSyn_00077 :**		
Name	EthTSynGlobalTimeSecureTmacLength		
Parent Container	EthTSynGlobalTimeDomain		
Description	Represents the number of bytes for the used Truncated Message Authentication Code (TMAC). If 0, no message authentication will be used. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 16		
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_00065 :**		
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 :**		
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).

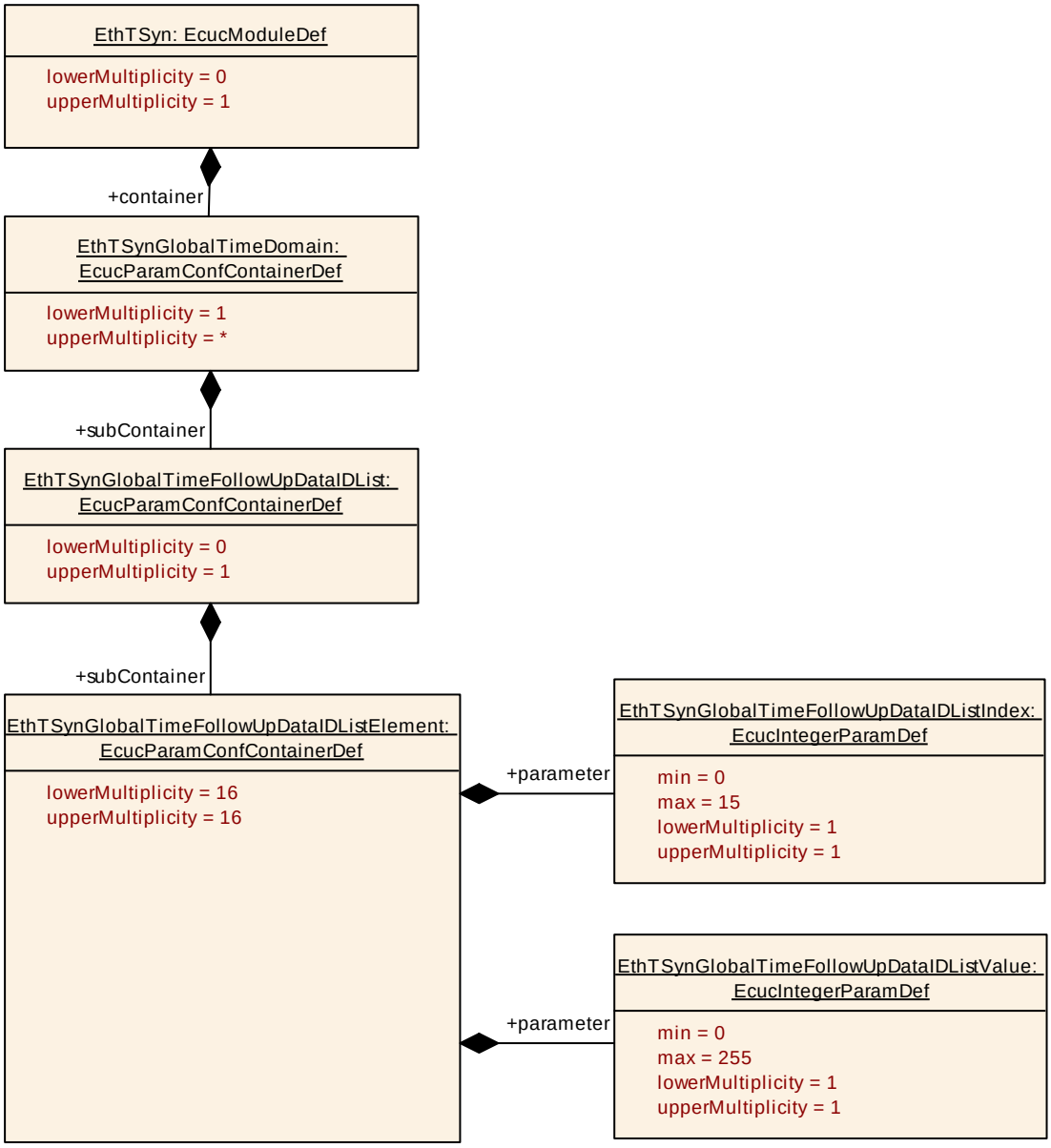


10.2.4 EthTSynGlobalTimeFollowUpDataIDList

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	true

Multiplicity			
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.



10.2.5 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 :**		
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 :**		
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

10.2.6 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

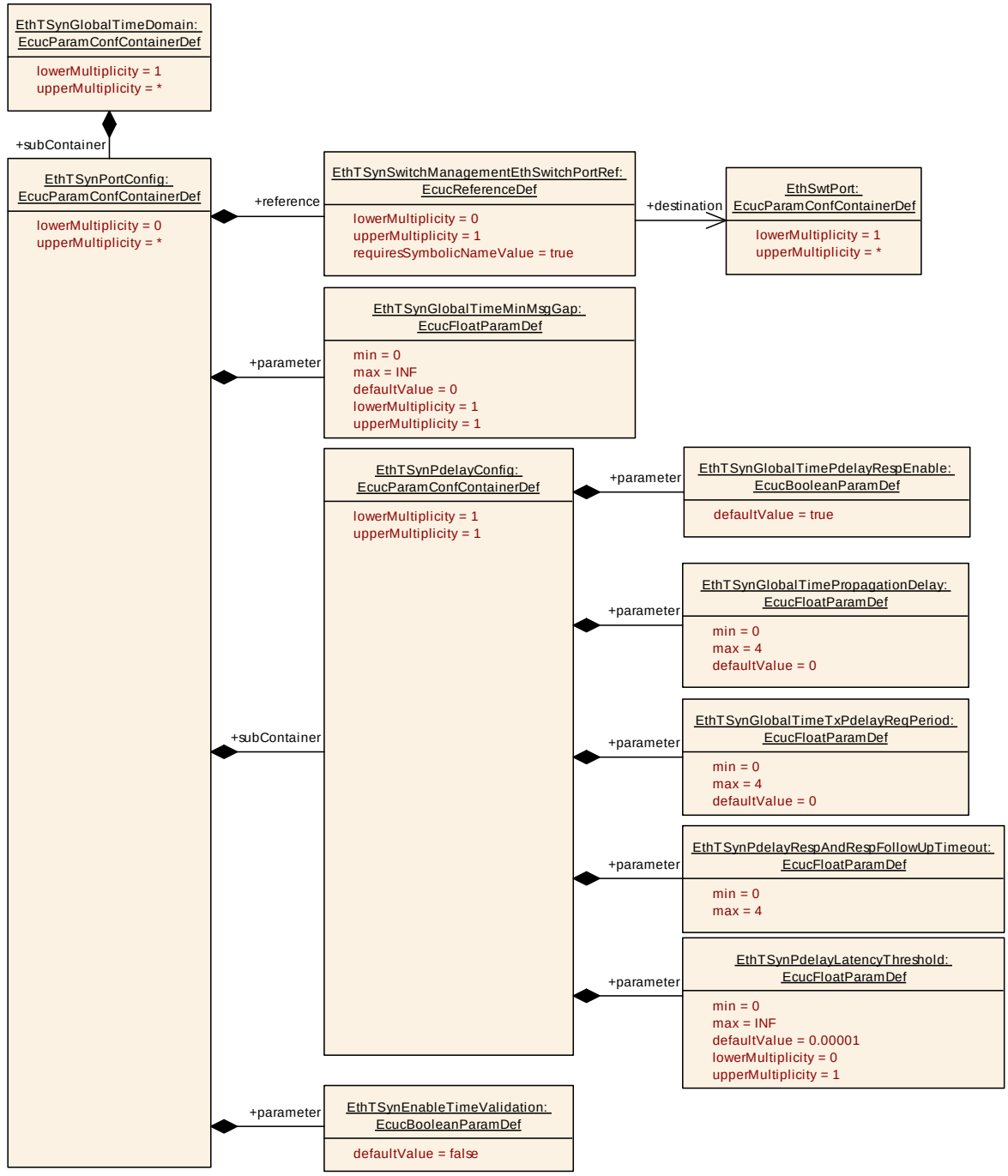
Multiplicity Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Configuration Parameters			

SWS Item	**ECUC_EthTSyn_00082 :**		
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_00078 :**		
Name	EthTSynGlobalTimeMinMsgGap		
Parent Container	EthTSynPortConfig		
Description	This parameter represents the configuration of a minimum message gap time for received Timesync messages compared to a message before with the same PDU. If PDUs are received more often in between than this parameter allows, they shall be ignored. Unit: seconds **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. INF[		
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_00066 :**		
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***
EthTSynPdelayConfig	1	Configuration of cyclic propagation delay measurement.

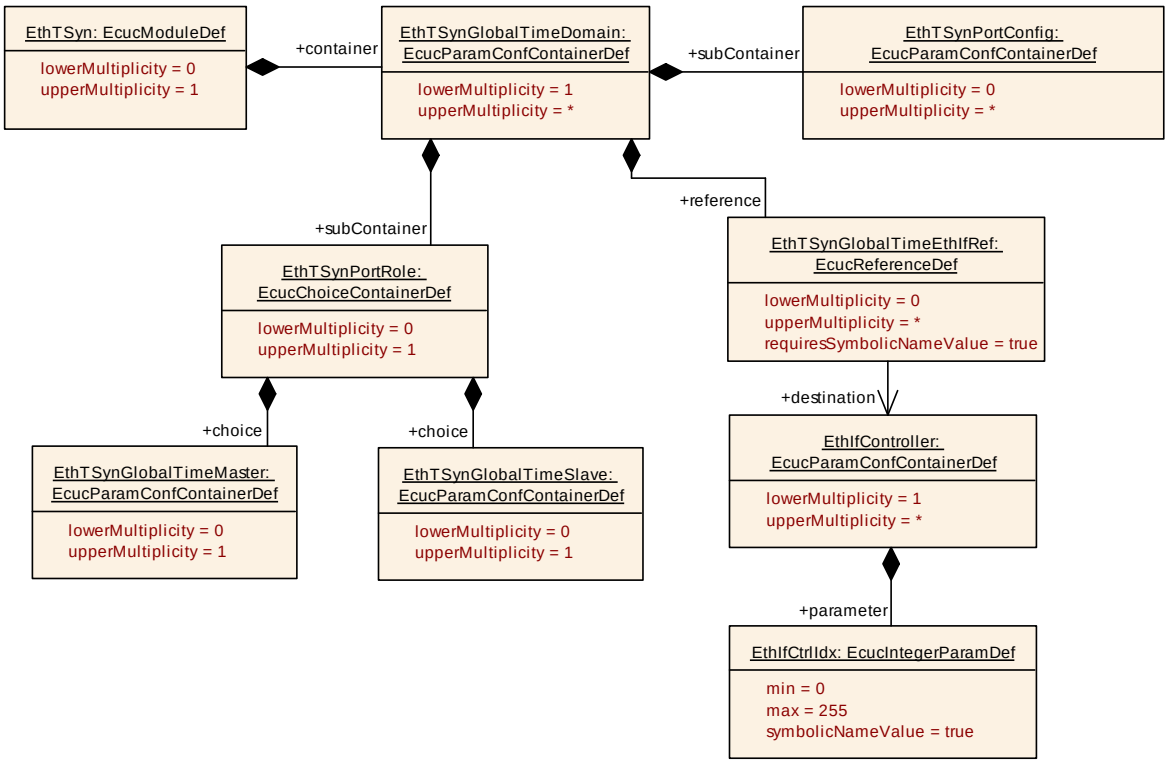


10.2.7 EthTSynPortRole

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.



10.2.8 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 :**		
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 :**		
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 :**		
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 :**		
Name	EthTSynPdelayLatencyThreshold		
Parent Container	EthTSynPdelayConfig		
Description	Threshold for calculated Pdelay. If a measured Pdelay exceeds EthTSynPdelayLatencyThreshold, 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 :**		
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		

No Included Containers

10.2.9 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 :**		
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 :**		
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_00010 :**		
Name	EthTSynGlobalTimeTxPeriod		
Parent Container	EthTSynGlobalTimeMaster		
Description	This represents configuration of the TX period. 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_00046 :**		
Name	EthTSynImmediateTimeSync		
Parent Container	EthTSynGlobalTimeMaster		
Description	Enables/Disables the cyclic polling of StbM_GetTimeBaseUpdateCounter() within EthTSyn_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_00038 :**		
Name	EthTSynTLVFollowUpOFSSubTLV		
Parent Container	EthTSynGlobalTimeMaster		
Description	This represents the configuration of whether an AUTOSAR Follow_Up TLV OFS Sub-TLV is used or not. • true: This represents a configuration where an AUTOSAR Follow_Up TLV OFS Sub-TLV is used. • false: This represents a configuration where an AUTOSAR Follow_Up TLV OFS Sub-TLV is not used.		
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 :**		
Name	EthTSynTLVFollowUpStatusSubTLV		
Parent Container	EthTSynGlobalTimeMaster		
Description	This represents the configuration of whether an AUTOSAR Follow_Up TLV Status Sub-TLV is used or not. • true: This represents a configuration where an AUTOSAR Follow_Up TLV Status Sub-TLV is used. • false: This represents a configuration where an AUTOSAR Follow_Up TLV Status Sub-TLV is not used.		
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		

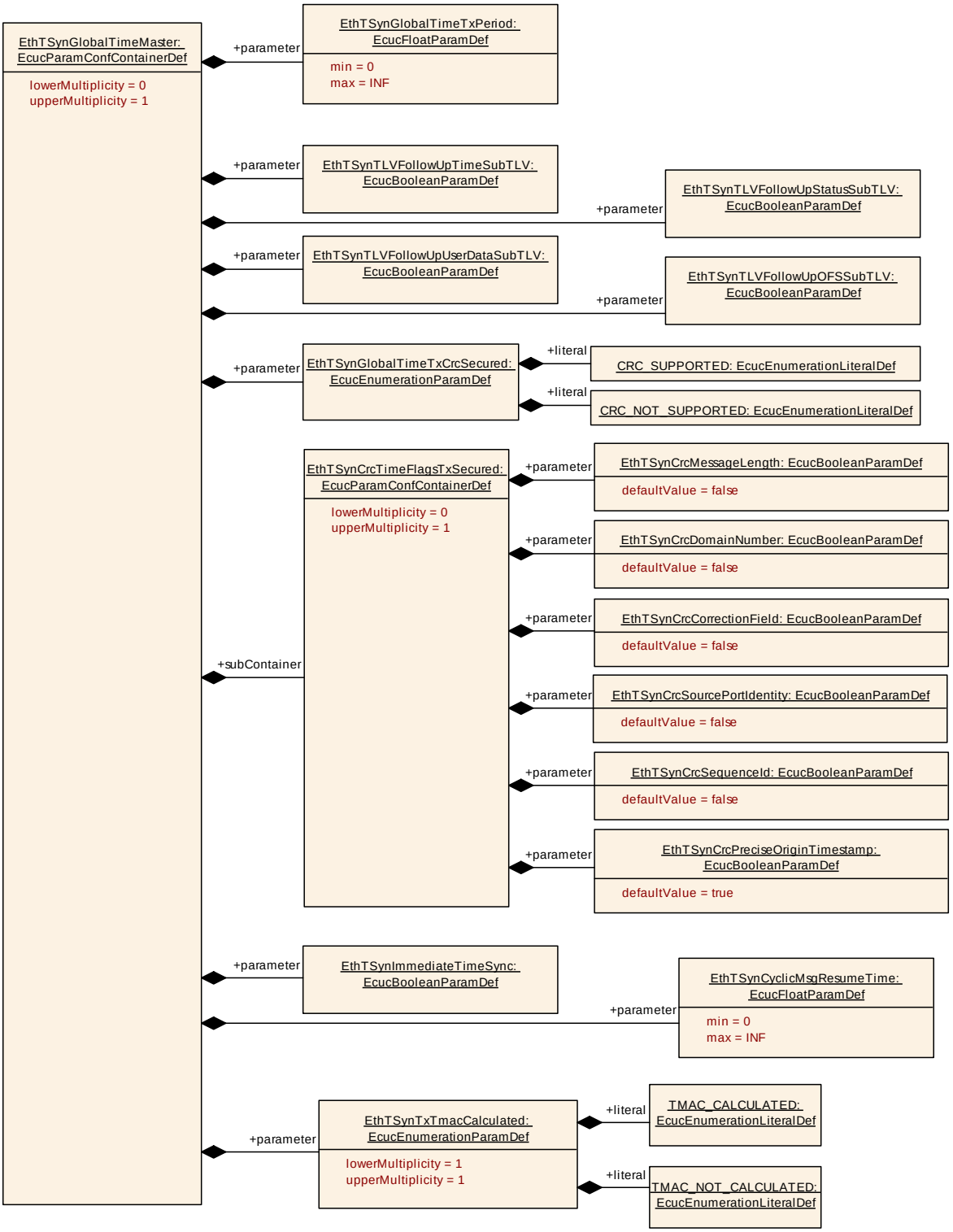
SWS Item	**ECUC_EthTSyn_00035 :**		
Name	EthTSynTLVFollowUpTimeSubTLV		
Parent Container	EthTSynGlobalTimeMaster		
Description	This represents the configuration of whether an AUTOSAR Follow_Up TLV Time Sub-TLV is used or not. • true: This represents a configuration where an AUTOSAR Follow_Up TLV Time Sub-TLV is used. • false: This represents a configuration where an AUTOSAR Follow_Up TLV Time Sub-TLV is not used.		
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 :**		
Name	EthTSynTLVFollowUpUserDataSubTLV		
Parent Container	EthTSynGlobalTimeMaster		
Description	This represents the configuration of whether an AUTOSAR Follow_Up TLV UserData Sub-TLV is used or not. • true: This represents a configuration where an AUTOSAR Follow_Up TLV UserData Sub-TLV is used. • false: This represents a configuration where an AUTOSAR Follow_Up TLV UserData Sub-TLV is not used.		
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_00079 :**		
Name	EthTSynTxTmacCalculated		
Parent Container	EthTSynGlobalTimeMaster		
Description	This parameter controls whether or not TMAC calculation shall be supported. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	TMAC_CALCULATED	The Timesync module shall calculate the TMAC.	
	TMAC_NOT_CALCULATED	The Timesync module shall not calculate any TMAC.	
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.



10.2.10 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 :**		
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 :**		
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 :**		
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 :**
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 :**		
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 :**		
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

10.2.11 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 :**		
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		

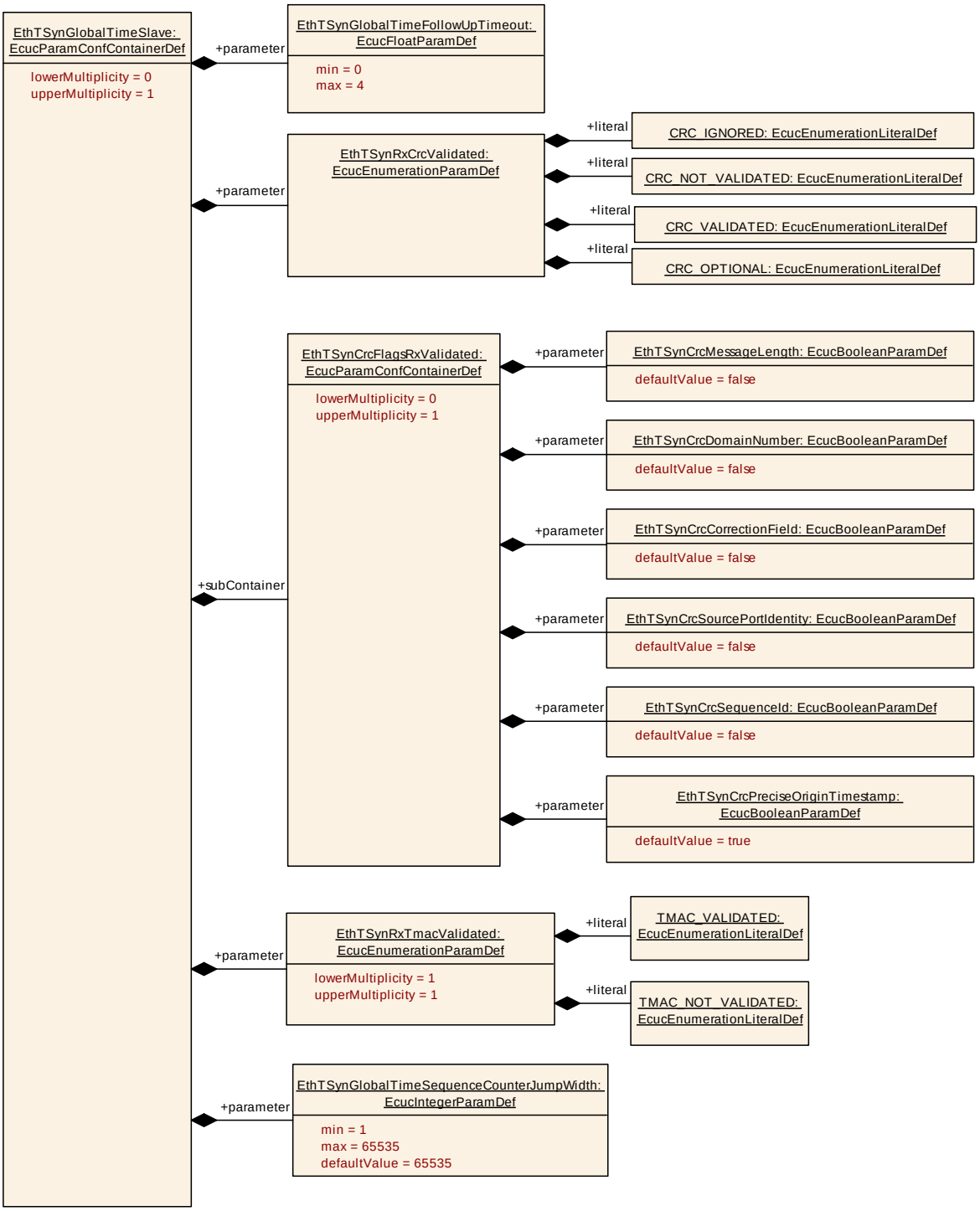
SWS Item	**ECUC_EthTSyn_00083 :**		
Name	EthTSynGlobalTimeSequenceCounterJumpWidth		
Parent Container	EthTSynGlobalTimeSlave		
Description	The SequenceCounterJumpWidth specifies the maximum allowed jump of the Sequence Counter between two consecutive Sync messages.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 65535		
Default value	65535		
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 :**	
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_00080 :**		
Name	EthTSynRxTmacValidated		
Parent Container	EthTSynGlobalTimeSlave		
Description	This parameter controls whether or not TMAC validation shall be supported. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	TMAC_NOT_VALIDATED	The Timesync module shall not validate the TMAC.	
	TMAC_VALIDATED	The Timesync module shall validate the TMAC.	
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***
EthTSynCrcFlagsRxValidated	0..1	This container collects definitions which parts of the Follow_Up message elements shall be included in CRC validation.



10.2.12 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_00053 :**		
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_00052 :**		
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_00051 :**		
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_00056 :**
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_00055 :**		
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_00054 :**		
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

[SWS_EthTSyn_CONSTR_00001]
The EthTSynPortConfig container exists for Synchronized Time Domains (`EthTSynGlobalTimeDomain 0 .. 15`) only.

10.4 Published Information

For details refer to the chapter 10.3 “Published Information” in *SWS_BSWGeneral.*