

|                                   |   |
|-----------------------------------|---|
| <b>Document Title</b>             | Time Synchronization Protocol Specification |
| <b>Document Owner</b>             | AUTOSAR                                     |
| <b>Document Responsibility</b>    | AUTOSAR                                     |
| <b>Document Identification No</b> | 897   |

|                                 |            |
|---------------------------------|------------|
| <b>Document Status</b>          | published  |
| <b>Part of AUTOSAR Standard</b> | Foundation |
| <b>Part of Standard Release</b> | R20-11     |

| <b>Document Change History</b> |                |                            |  |
|--------------------------------|----------------|----------------------------|--|
| <b>Date</b>                    | <b>Release</b> | <b>Changed by</b>          | <b>Description</b>   |
| 2020-11-30                     | R20-11         | AUTOSAR Release Management | <ul style="list-style-type: none"> <li>Added Sequence Counter handling</li> <li>New configuration parameters</li> </ul>                                      |
| 2019-11-28                     | R19-11         | AUTOSAR Release Management | <ul style="list-style-type: none"> <li>Clarified SGW value handling for missing Sub-TLVs</li> <li>Changed Document Status from Final to published</li> </ul> |
| 2019-03-29                     | 1.5.1          | AUTOSAR Release Management | <ul style="list-style-type: none"> <li>Minor changes</li> </ul>  |
| 2018-10-31                     | 1.5.0          | AUTOSAR Release Management | <ul style="list-style-type: none"> <li>Initial release</li> </ul>  |

## **Disclaimer**

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

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

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

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

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

## Table of Contents

|         |   |    |
|---------|---|----|
| 1       | Introduction and overview                           | 5  |
| 1.1     | Protocol purpose and objectives                     | 5  |
| 1.2     | Applicability of the protocol                       | 5  |
| 1.2.1   | Constraints and assumptions                         | 6  |
| 1.2.2   | Limitations   | 6  |
| 1.2.3   | Accuracy  | 6  |
| 1.3     | Dependencies  | 6  |
| 1.3.1   | Dependencies to other protocol layers               | 6  |
| 1.3.2   | Dependencies to other standards and norms           | 7  |
| 1.3.3   | Dependencies to the Application Layer               | 7  |
| 2       | Use Cases   | 7  |
| 3       | Protocol Requirements                               | 7  |
| 3.1     | Requirements Traceability                           | 7  |
| 4       | Definition of terms and acronyms                    | 10 |
| 4.1     | Acronyms and abbreviations                          | 10 |
| 5       | Protocol specification                              | 10 |
| 5.1     | General   | 10 |
| 5.2     | VLAN Support  | 11 |
| 5.3     | Message format                                      | 11 |
| 5.3.1   | Header format                                       | 12 |
| 5.3.1.1 | Sync and Follow_Up acc. to IEEE 802.1AS             | 12 |
| 5.3.1.2 | Sync and Follow_Up acc. to AUTOSAR                  | 14 |
| 5.3.1.3 | Follow_Up Message Header [AUTOSAR]                  | 19 |
| 5.3.1.4 | AUTOSAR TLV Header                                  | 20 |
| 5.3.1.5 | AUTOSAR TLV Sub-TLVs                                | 20 |
| 5.3.1.6 | AUTOSAR TLV Sub-TLV: Time Secured                   | 20 |
| 5.3.1.7 | AUTOSAR TLV Sub-TLV: Status Secured / Not Secured   | 20 |
| 5.3.1.8 | AUTOSAR TLV Sub-TLV: UserData Secured / Not Secured | 21 |
| 5.3.1.9 | AUTOSAR TLV Sub-TLV: OFS Secured / Not Secured      | 21 |
| 5.3.2   | Body/Payload format                                 | 21 |
| 5.3.3   | Data Types  | 22 |
| 5.4     | Message types                                       | 22 |
| 5.4.1   | Data Messages                                       | 22 |
| 5.4.2   | Control Messages                                    | 22 |
| 5.5     | Services / Commands                                 | 22 |
| 5.6     | Sequences (lower layer)                             | 23 |
| 5.6.1   | Pdelay Protocol for Latency Calculation             | 23 |
| 5.6.2   | Acting as Time Master                               | 25 |

|             |   |    |
|-------------|---|----|
| 5.6.2.1     | Message Processing . . . . .                                | 25 |
| 5.6.2.1.1   | Frame Debouncing . . . . .                                  | 25 |
| 5.6.2.2     | Message Field Calculation and Assembling . . . . .          | 26 |
| 5.6.2.2.1   | SGW Calculation . . . . .                                   | 26 |
| 5.6.2.2.2   | OFS Calculation . . . . .                                   | 26 |
| 5.6.2.2.3   | CRC Calculation . . . . .                                   | 27 |
| 5.6.2.2.3.1 | AUTOSAR TLV Sub-TLV: Time Secured                           | 27 |
| 5.6.2.2.3.2 | AUTOSAR TLV Sub-TLV: Status secured                         | 29 |
| 5.6.2.2.3.3 | AUTOSAR TLV Sub-TLV: UserData se-<br>cured . . . . .        | 29 |
| 5.6.2.2.3.4 | AUTOSAR TLV Sub-TLV: OFS secured                            | 29 |
| 5.6.2.2.4   | Sequence Counter Calculation . . . . .                      | 29 |
| 5.6.2.2.5   | Message Assembling . . . . .                                | 30 |
| 5.6.3       | Acting as Time Slave . . . . .                              | 30 |
| 5.6.3.1     | Message processing . . . . .                                | 30 |
| 5.6.3.2     | Message Field Validation and Disassembling . . . . .        | 31 |
| 5.6.3.2.1   | SGW Calculation . . . . .                                   | 32 |
| 5.6.3.2.2   | OFS Calculation . . . . .                                   | 33 |
| 5.6.3.2.3   | CRC Validation . . . . .                                    | 33 |
| 5.6.3.2.3.1 | AUTOSAR TLV Sub-TLV: Time Secured                           | 33 |
| 5.6.3.2.3.2 | AUTOSAR TLV Sub-TLV: Status secured                         | 35 |
| 5.6.3.2.3.3 | AUTOSAR TLV Sub-TLV: UserData se-<br>cured . . . . .        | 35 |
| 5.6.3.2.3.4 | AUTOSAR TLV Sub-TLV: OFS secured                            | 36 |
| 5.6.3.2.4   | Sequence Counter (sequenceld) Validation . .                | 36 |
| 5.6.3.2.5   | Message Disassembling . . . . .                             | 37 |
| 5.7         | Time measurement with Switches . . . . .                    | 38 |
| 5.8         | Pdelay and Time Synchronization measurement point . . . . . | 40 |
| 5.9         | Time Aware Bridge with GTM as Management CPU . . . . .      | 41 |
| 5.10        | Time Aware Bridge with GTM not as Management CPU . . . . .  | 43 |
| 5.11        | Error messages . . . . .                                    | 45 |
| 6           | Configuration parameters . . . . .                          | 45 |
| 7           | Protocol usage and guidelines . . . . .                     | 47 |
| 8           | References . . . . .  | 48 |

# 1 Introduction and overview

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

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

- Role and functions of ECUs is known and defined a priori
- The network is static, i.e. components like ECUs, switches and characteristics like cable length, don't change during operation or even after switching off and switching on the vehicle. Components of course may be unavailable (due to failure situations or by purpose) but mostly only change when the vehicle is at a service facility.

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

## 1.1 Protocol purpose and objectives

The Time synchronization protocol is used to

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

## 1.2 Applicability of the protocol

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

### 1.2.1 Constraints and assumptions

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

### 1.2.2 Limitations

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

### 1.2.3 Accuracy

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

## 1.3 Dependencies

### 1.3.1 Dependencies to other protocol layers

There are no dependencies to other protocols.

### 1.3.2 Dependencies to other standards and norms

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

### 1.3.3 Dependencies to the Application Layer

There are no dependencies to the application layer.

## 2 Use Cases

| <i>ID</i> | <i>Name</i>          | <i>Description</i>                            |
|-----------|----------------------|---|
| 0001      | Pdelay measurement   | Measuring of delays between Ethernet messages |
| 0002      | Time Synchronization | Time synchronization of different time bases. |

## 3 Protocol Requirements

### 3.1 Requirements Traceability

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

| Requirement   | Description  | Satisfied by   |
|---------------|--|--|
|               |  | [PRS_TS_00067] [PRS_TS_00068]<br>[PRS_TS_00069] [PRS_TS_00070]<br>[PRS_TS_00071] [PRS_TS_00072]<br>[PRS_TS_00075] [PRS_TS_00077]<br>[PRS_TS_00079] [PRS_TS_00086]<br>[PRS_TS_00141] [PRS_TS_00142]<br>[PRS_TS_00149] [PRS_TS_00154]<br>[PRS_TS_00163] [PRS_TS_00164]<br>[PRS_TS_00166] [PRS_TS_00167]<br>[PRS_TS_00168] [PRS_TS_00169]<br>[PRS_TS_00170] [PRS_TS_00171]<br>[PRS_TS_00181] [PRS_TS_00206]<br><br>[PRS_TS_00207] [PRS_TS_00208]<br>[PRS_TS_00209] [PRS_TS_00210] |
| [RS_TS_20051] | The Timesync over Ethernet module shall detect and handle errors in synchronization protocol / communication       | [PRS_TS_00004] [PRS_TS_00025]<br>[PRS_TS_00164] [PRS_TS_00210]   |
| [RS_TS_20052] | The configuration of the Time Synchronization over Ethernet module shall allow the module to work as a Time Master | [PRS_TS_00094]   |
| [RS_TS_20054] | The Implementation of the Time Synchronization shall evaluate and propagate Time Gateway relevant information      | [PRS_TS_00094] [PRS_TS_00156]<br>[PRS_TS_00211] [PRS_TS_00212]<br>[PRS_TS_00213]   |
| [RS_TS_20059] | The Timesync over Ethernet module shall access all communication ports belonging to Time Synchronization           | [PRS_TS_00053] [PRS_TS_00054]<br>[PRS_TS_00055] [PRS_TS_00056]<br>[PRS_TS_00057] [PRS_TS_00058]<br>[PRS_TS_00059] [PRS_TS_00060]<br>[PRS_TS_00166] [PRS_TS_00167]<br>[PRS_TS_00168] [PRS_TS_00169]<br>[PRS_TS_00170] [PRS_TS_00171]<br>[PRS_TS_00207] [PRS_TS_00208]<br>[PRS_TS_00209]   |
| [RS_TS_20061] | The Timesync over Ethernet module shall support means to protect the Time Synchronization protocol                 | [PRS_TS_000104] [PRS_TS_00028]<br>[PRS_TS_00062] [PRS_TS_00063]<br>[PRS_TS_00066] [PRS_TS_00067]<br>[PRS_TS_00068] [PRS_TS_00069]<br>[PRS_TS_00070] [PRS_TS_00071]<br>[PRS_TS_00072] [PRS_TS_00074]<br>[PRS_TS_00075] [PRS_TS_00076]<br>[PRS_TS_00077] [PRS_TS_00078]<br>[PRS_TS_00079] [PRS_TS_00081]<br>[PRS_TS_00082] [PRS_TS_00084]<br>[PRS_TS_00085] [PRS_TS_00086]<br>[PRS_TS_00088] [PRS_TS_00089]  |



| Requirement          | Description   | Satisfied by  |
|----------------------|---|---|
|                      |   | <a href="#">[PRS_TS_00091]</a> <a href="#">[PRS_TS_00092]</a><br><a href="#">[PRS_TS_00093]</a> <a href="#">[PRS_TS_00097]</a><br><a href="#">[PRS_TS_00098]</a> <a href="#">[PRS_TS_00099]</a><br><a href="#">[PRS_TS_00100]</a> <a href="#">[PRS_TS_00101]</a><br><a href="#">[PRS_TS_00102]</a> <a href="#">[PRS_TS_00103]</a><br><a href="#">[PRS_TS_00105]</a> <a href="#">[PRS_TS_00106]</a><br><a href="#">[PRS_TS_00107]</a> <a href="#">[PRS_TS_00108]</a><br><a href="#">[PRS_TS_00109]</a> <a href="#">[PRS_TS_00112]</a><br><a href="#">[PRS_TS_00113]</a> <a href="#">[PRS_TS_00114]</a><br><a href="#">[PRS_TS_00115]</a> <a href="#">[PRS_TS_00116]</a><br><a href="#">[PRS_TS_00117]</a> <a href="#">[PRS_TS_00118]</a><br><a href="#">[PRS_TS_00119]</a> <a href="#">[PRS_TS_00120]</a><br><br><a href="#">[PRS_TS_00157]</a> <a href="#">[PRS_TS_00181]</a><br><a href="#">[PRS_TS_00182]</a> <a href="#">[PRS_TS_00183]</a><br><a href="#">[PRS_TS_00184]</a> <a href="#">[PRS_TS_00185]</a><br><a href="#">[PRS_TS_00187]</a> <a href="#">[PRS_TS_00188]</a><br><a href="#">[PRS_TS_00189]</a> <a href="#">[PRS_TS_00190]</a><br><a href="#">[PRS_TS_00191]</a> <a href="#">[PRS_TS_00192]</a><br><a href="#">[PRS_TS_00193]</a> <a href="#">[PRS_TS_00194]</a><br><a href="#">[PRS_TS_00195]</a> <a href="#">[PRS_TS_00196]</a><br><a href="#">[PRS_TS_00197]</a> <a href="#">[PRS_TS_00198]</a><br><a href="#">[PRS_TS_00199]</a> |
| <b>[RS_TS_20062]</b> | The Timesync over Ethernet module shall support user specific data within the time measurement and synchronization protocol                         | <a href="#">[PRS_TS_000104]</a> <a href="#">[PRS_TS_00028]</a><br><a href="#">[PRS_TS_00062]</a> <a href="#">[PRS_TS_00063]</a><br><a href="#">[PRS_TS_00066]</a> <a href="#">[PRS_TS_00067]</a><br><a href="#">[PRS_TS_00068]</a> <a href="#">[PRS_TS_00069]</a><br><a href="#">[PRS_TS_00070]</a> <a href="#">[PRS_TS_00071]</a><br><a href="#">[PRS_TS_00072]</a> <a href="#">[PRS_TS_00074]</a><br><a href="#">[PRS_TS_00075]</a> <a href="#">[PRS_TS_00076]</a><br><a href="#">[PRS_TS_00077]</a> <a href="#">[PRS_TS_00078]</a><br><a href="#">[PRS_TS_00079]</a> <a href="#">[PRS_TS_00081]</a><br><a href="#">[PRS_TS_00082]</a> <a href="#">[PRS_TS_00084]</a><br><a href="#">[PRS_TS_00085]</a> <a href="#">[PRS_TS_00086]</a><br><a href="#">[PRS_TS_00088]</a> <a href="#">[PRS_TS_00089]</a><br><br><a href="#">[PRS_TS_00092]</a> <a href="#">[PRS_TS_00103]</a><br><a href="#">[PRS_TS_00105]</a> <a href="#">[PRS_TS_00106]</a><br><a href="#">[PRS_TS_00118]</a> <a href="#">[PRS_TS_00119]</a><br><a href="#">[PRS_TS_00120]</a> <a href="#">[PRS_TS_00181]</a>   |
| <b>[RS_TS_20063]</b> | The Timesync over Ethernet module shall use the Time Synchronization protocol for Synchronized Time Bases to transmit and receive Offset Time Bases | <a href="#">[PRS_TS_000104]</a> <a href="#">[PRS_TS_00092]</a><br><a href="#">[PRS_TS_00095]</a> <a href="#">[PRS_TS_00103]</a><br><a href="#">[PRS_TS_00105]</a> <a href="#">[PRS_TS_00106]</a><br><a href="#">[PRS_TS_00110]</a> <a href="#">[PRS_TS_00117]</a><br><a href="#">[PRS_TS_00118]</a> <a href="#">[PRS_TS_00119]</a><br><a href="#">[PRS_TS_00120]</a>  |
| <b>[RS_TS_20066]</b> | The Timesync over Ethernet module shall support a static (pre)configuration of IEEE 802.1AS Pdelay  | <a href="#">[PRS_TS_00003]</a> <a href="#">[PRS_TS_00011]</a><br><a href="#">[PRS_TS_00012]</a> <a href="#">[PRS_TS_00140]</a><br><a href="#">[PRS_TS_00141]</a> <a href="#">[PRS_TS_00142]</a><br><a href="#">[PRS_TS_00143]</a> <a href="#">[PRS_TS_00149]</a>  |

## 4 Definition of terms and acronyms

### 4.1 Acronyms and abbreviations

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

## 5 Protocol specification

### 5.1 General

**[PRS\_TS\_00002]** [The Time Master and Time Slave shall use the default configuration values as defined by [1, IEEE 802.1 AS] (e.g. MAC destination address or Ethernet frame type), if not otherwise specified within this specification.] ([RS\\_TS\\_20048](#))

**[PRS\_TS\_00005]** [The Time Master and Time Slave shall start their protocol state machines without Announce message recognition.] ([RS\\_TS\\_20048](#))

**[PRS\_TS\_00206]** [All messages belonging to the IEEE Rapid Spanning Tree Protocol (`PortAnnounceReceive`, `PortAnnounceInformation`, `PortRoleSelection`, `PortAnnounceTransmit`) shall be ignored on the receiver side.] ([RS\\_TS\\_20048](#))

**Note:** AUTOSAR implementations shall not send those messages.

## 5.2 VLAN Support

**[PRS\_TS\_00163]** [If `FramePrio` exists, a frame format with priority and VLAN tags shall be used. Otherwise a frame format without priority and VLAN tags shall be used.] ([RS\\_TS\\_20048](#))

## 5.3 Message format

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

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

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

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

**[PRS\_TS\_00028]** [The message format, etc. shall be derived from [1, IEEE 802.1 AS] chapter 10. Media-independent layer specification and chapter 11. Media-dependent layer specification for full-duplex, point-to-point links, if not otherwise specified.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

**[PRS\_TS\_00181]** [The byte order for multibyte values is Big Endian, which is equal to the byte order defined by [1, IEEE 802.1 AS].] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

### 5.3.1 Header format

#### 5.3.1.1 Sync and Follow\_Up acc. to IEEE 802.1AS

**[PRS\_TS\_00061]** [If `MessageCompliance` is set to TRUE, `Sync` and `Follow_Up` format shall be supported acc. to [1, IEEE 802.1 AS].] ([RS\\_TS\\_20048](#))

**Note:** This implies one Time Domain (0).

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

| <b>Sync Message Header [IEEE 802.1AS]</b> |              |        |        |   |
|---|--------------|--------|--------|---|
| High Nibble                               | Low Nibble   | Octets | Offset | Value   |
| transportSpecific                         | message-type | 1      | 0      | 0x10  |
| reserved                                  | versionPTP   | 1      | 1      | 2   |
| length of the message                     |              | 2      | 2      | 44  |
| domainNumber                              |              | 1      | 4      | (UInteger8)<br>domainNumber = 0   |
| reserved                                  |              | 1      | 5      | 0   |
| flags                                     |              | 2      | 6      | Octet 0: 0x02, Octet 1: 0x08  |
| correctionField                           |              | 8      | 8      | 0..281474976710655ns<br>[1ns = 2 <sup>16</sup> = 0x0000 0000 0001 0000] |
| reserved                                  |              | 4      | 16     | 0   |
| sourcePortIdentity                        |              | 10     | 20     | (PortIdentity)<br>portIdentity from origin<br>Time Aware End Station    |
| sequenceId                                |              | 2      | 30     | (UInteger16)<br>SyncSequenceId =<br>(UInteger16) (prevSyncSequenceId+1) |
| control                                   |              | 1      | 32     | 0   |
| logMessageInterval                        |              | 1      | 33     | (Integer8) current-<br>LogSyncInterval                                  |
| <b>Sync Message Fields [IEEE 802.1AS]</b> |              |        |        |   |



△

| High Nibble        | Low Nibble | Octets | Offset | Value                       |
|--------------------|------------|--------|--------|-----------------------------|
| PTP Message Header |            | 34     | 0      | [refer Sync Message Header] |
| reserved           |            | 10     | 34     | 0                           |

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

### Follow\_Up Message Header [IEEE 802.1AS]

| <i>Follow_Up Message Header [IEEE 802.1AS]</i> |              |        |        |   |
|--|--------------|--------|--------|---|
| High Nibble                                    | Low Nibble   | Octets | Offset | Value   |
| transportSpecific                              | message-type | 1      | 0      | 0x18  |
| reserved                                       | versionPTP   | 1      | 1      | 0x02  |
| length of the message                          |              | 2      | 2      | 76  |
| domainNumber                                   |              | 1      | 4      | (UInteger8)<br>domainNumber = 0   |
| reserved                                       |              | 1      | 5      | 0   |
| flags  |              | 2      | 6      | Octet 0: 0x00, Octet 1: 0x08  |
| correctionField                                |              | 8      | 8      | 0..281474976710655ns<br>[1ns = 2 <sup>16</sup> = 0x0000 0000 0001 0000] |
| reserved                                       |              | 4      | 16     | 0   |
| sourcePortIdentity                             |              | 10     | 20     | (PortIdentity)<br>portIdentity from origin<br>Time Aware End Station    |
| sequenceId                                     |              | 2      | 30     | UInteger16)<br>SyncSequenceId   |
| control  |              | 1      | 32     | 2   |
| logMessageInterval                             |              | 1      | 33     | (Integer8) current-<br>LogSyncInterval                                  |

### *Follow\_Up Message Fields [IEEE 802.1AS]*

| High Nibble            | Low Nibble | Octets | Offset | Value                                 |
|------------------------|------------|--------|--------|---------------------------------------|
| PTP Message Header     |            | 34     | 0      | [refer Follow_Up Message Header]      |
| preciseOriginTimestamp |            | 10     | 34     | (Timestamp)<br>preciseOriginTimestamp |

▽

△

|  |                   |               |                                 |   |
|--|-------------------|---------------|---------------------------------|---|
| <b>Follow_Up information TLV</b>                       | 32                | 44            | refer Follow_Up information TLV |   |
| <b><i>Follow_Up information TLV [IEEE 802.1AS]</i></b> |                   |               |                                 |   |
| <b>High Nibble</b>                                     | <b>Low Nibble</b> | <b>Octets</b> | <b>Offset</b>                   | <b>Value</b>                                  |
| <b>tlvType</b>   |                   | 2             | 0                               | 3   |
| <b>lengthField</b>                                     |                   | 2             | 2                               | 28  |
| <b>organizationId</b>                                  |                   | 3             | 4                               | 0x0080c2                                      |
| <b>organizationSub-Type</b>                            |                   | 3             | 7                               | 1   |
| <b>cumulativeScaledRateOffset</b>                      |                   | 4             | 10                              | (Integer32) ( (RateRatio-1)*2 <sup>41</sup> ) |
| <b>gmTimeBaseIndicator</b>                             |                   | 2             | 14                              | 0   |
| <b>lastGm-PhaseChange</b>                              |                   | 12            | 16                              | 0   |
| <b>scaledLastGm-FreqChange</b>                         |                   | 4             | 28                              | 0   |

### 5.3.1.2 Sync and Follow\_Up acc. to AUTOSAR

**[PRS\_TS\_00062]** [If [MessageCompliance](#) is set to FALSE, the Sync and Follow\_Up format shall be supported acc. to: Follow\_Up Message Header [AUTOSAR] and Sync Message Header [AUTOSAR] depending on configuration.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

**[PRS\_TS\_00063]** [If [MessageCompliance](#) is set to FALSE, the Follow\_Up shall contain an AUTOSAR TLV, depending on configuration.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

#### Message Header [AUTOSAR]

|   |                     |               |               |              |
|---|---------------------|---------------|---------------|--------------|
| <b><i>Sync Message Header [AUTOSAR]</i></b> |                     |               |               |              |
| <b>High Nibble</b>                          | <b>Low Nibble</b>   | <b>Octets</b> | <b>Offset</b> | <b>Value</b> |
| <b>transportSpecific</b>                    | <b>message-type</b> | 1             | 0             | 0x10         |
| <b>reserved</b>                             | <b>versionPTP</b>   | 1             | 1             | 2            |
| <b>length of the message</b>                |                     | 2             | 2             | 44           |

▽

△

|                                      |                   |               |   |                             |
|--------------------------------------|-------------------|---------------|---|-----------------------------|
| <b>domainNumber</b>                  | 1                 | 4             | (UInteger8)<br>domainNumber = 0..15   |                             |
| <b>reserved</b>                      | 1                 | 5             | 0   |                             |
| <b>flags</b>                         | 2                 | 6             | Octet 0: 0x02,<br>Octet 1: 0x08   |                             |
| <b>correctionField</b>               | 8                 | 8             | 0..281474976710655ns<br>[1ns = 2 <sup>16</sup> = 0x0000<br>0000 0001 0000]  |                             |
| <b>reserved</b>                      | 4                 | 16            | 0   |                             |
| <b>sourcePortIdentity</b>            | 10                | 20            | (PortIdentity)<br>portIdentity from origin<br><b>Time Aware End Station</b> |                             |
| <b>sequenceId</b>                    | 2                 | 30            | (UInteger16)<br>SyncSequenceId =<br>(UInteger16) (prevSyncSequenceId+1)     |                             |
| <b>control</b>                       | 1                 | 32            | 0   |                             |
| <b>logMessageInterval</b>            | 1                 | 33            | (Integer8) current-<br>LogSyncInterval                                      |                             |
| <b>Sync Message Fields [AUTOSAR]</b> |                   |               |   |                             |
| <b>High Nibble</b>                   | <b>Low Nibble</b> | <b>Octets</b> | <b>Offset</b>   | <b>Value</b>                |
| <b>PTP Message Header</b>            |                   | 34            | 0   | [refer Sync Message Header] |
| <b>reserved</b>                      |                   | 10            | 34  | 0                           |

|   |                     |               |               |  |
|---|---------------------|---------------|---------------|--|
| <b>Follow_Up Message Header [AUTOSAR]</b> |                     |               |               |  |
| <b>High Nibble</b>                        | <b>Low Nibble</b>   | <b>Octets</b> | <b>Offset</b> | <b>Value</b>   |
| <b>transportSpecific</b>                  | <b>message-type</b> | 1             | 0             | 0x18   |
| <b>reserved</b>                           | <b>versionPTP</b>   | 1             | 1             | 0x02   |
| <b>length of the message</b>              |                     | 2             | 2             | 76+10+Sum(Sub-TLVs)  |
| <b>domainNumber</b>                       |                     | 1             | 4             | (UInteger8)<br>domainNumber = 0..15  |
| <b>reserved</b>                           |                     | 1             | 5             | 0  |
| <b>flags</b>                              |                     | 2             | 6             | Octet 0: 0x00, Octet<br>1: 0x08  |
| <b>correctionField</b>                    |                     | 8             | 8             | 0..281474976710655ns<br>[1ns = 2 <sup>16</sup> = 0x0000<br>0000 0001 0000] |

▽

△

|   |                   |                            |  |   |
|---|-------------------|----------------------------|--|---|
| <b>reserved</b>                                 | 4                 | 16                         | 0  |   |
| <b>sourcePortIdentity</b>                       | 10                | 20                         | (PortIdentity)<br>portIdentity from origin<br>Time Aware End Station |   |
| <b>sequenceId</b>                               | 2                 | 30                         | (UInteger16)<br>SyncSequenceId                                       |   |
| <b>control</b>                                  | 1                 | 32                         | 2  |   |
| <b>logMessageInterval</b>                       | 1                 | 33                         | (Integer8) current-<br>LogSyncInterval                               |   |
| <b>Follow_Up Message Fields [AUTOSAR]</b>       |                   |                            |  |   |
| <b>High Nibble</b>                              | <b>Low Nibble</b> | <b>Octets</b>              | <b>Offset</b>  | <b>Value</b>                                    |
| <b>PTP Message Header</b>                       |                   | 34                         | 0  | [refer Follow_Up Message Header]                |
| <b>preciseOriginTimestamp</b>                   |                   | 10                         | 34   | (Timestamp)<br>preciseOriginTimestamp           |
| <b>Follow_Up information TLV</b>                |                   | 32 + 10 +<br>sum(Sub-TLVs) | 44   | [refer Follow_Up information TLV]               |
| <b>Follow_Up information TLV [IEEE 802.1AS]</b> |                   |                            |  |   |
| <b>High Nibble</b>                              | <b>Low Nibble</b> | <b>Octets</b>              | <b>Offset</b>  | <b>Value</b>                                    |
| <b>tlvType</b>                                  |                   | 2                          | 0  | 3   |
| <b>length of the field</b>                      |                   | 2                          | 2  | 28  |
| <b>organizationId</b>                           |                   | 3                          | 4  | 0x0080C2 [ IEEE<br>802.1AS]                     |
| <b>organizationSub-Type</b>                     |                   | 3                          | 7  | 1   |
| <b>cumulativeScaledRateOffset</b>               |                   | 4                          | 10   | (Integer32)((RateRatio-1)<br>*2 <sup>41</sup> ) |
| <b>gmTimeBaseIndicator</b>                      |                   | 2                          | 14   | 0   |
| <b>lastGmPhaseChange</b>                        |                   | 12                         | 16   | 0   |
| <b>scaledLastGmFreqChange</b>                   |                   | 4                          | 28   | 0   |
| <b>Follow_Up information TLV [AUTOSAR]</b>      |                   |                            |  |   |
| <b>High Nibble</b>                              | <b>Low Nibble</b> | <b>Octets</b>              | <b>Offset</b>  | <b>Value</b>                                    |
| <b>AUTOSAR TLV Header</b>                       |                   |                            |  |   |
| <b>tlvType</b>                                  |                   | 2                          | 0  | 3   |
| <b>length of the field</b>                      |                   | 2                          | 0  | 6 + Sum( <i>Sub-TLVs</i> )                      |
| <b>organizationId</b>                           |                   | 3                          | 4  | 0x1A75FB [AUTOSAR]                              |

▽



△

|  |                   |               |  |  |
|--|-------------------|---------------|--|--|
| <b>organizationSub-Type</b>                    | 3                 | 7             | 0x605676 [BCD coded GlobalTimeEthTSyn] |  |
| <b>AUTOSAR TLV Sub-TLV: Time Secured</b>       |                   |               |  |  |
| <b>High Nibble</b>                             | <b>Low Nibble</b> | <b>Octets</b> | <b>Offset</b>                          | <b>Value</b>   |
| <b>Type</b>                                    |                   | 1             | 0                                      | 0x28 [Time secured]  |
| <b>Length</b>                                  |                   | 1             | 1                                      | 3  |
| <b>CRC_Time_Flags</b>                          |                   | 1             | 2                                      | BitMask 0x01 [length of the message]<br>BitMask 0x02 [domainNumber]<br>BitMask 0x04 [correctionField]<br>BitMask 0x08 [sourcePortIdentity]<br>BitMask 0x10 [sequenceId]<br>BitMask 0x20 [preciseOriginTimestamp]<br>BitMask 0x40 [reserved]<br>BitMask 0x80 [reserved] |
| <b>CRC_Time_0</b>                              |                   | 1             | 3                                      | 0..255   |
| <b>CRC_Time_1</b>                              |                   | 1             | 4                                      | 0..255   |
| <b>AUTOSAR TLV Sub-TLV: Status Secured</b>     |                   |               |  |  |
| <b>High Nibble</b>                             | <b>Low Nibble</b> | <b>Octets</b> | <b>Offset</b>                          | <b>Value</b>   |
| <b>Type</b>                                    |                   | 1             | 0                                      | 0x50[Status secured]   |
| <b>Length</b>                                  |                   | 1             | 1                                      | 2  |
| <b>Status</b>                                  |                   | 1             | 2                                      | BitMask 0x01 [SGW with SyncToGTM = 0<br>SyncToSubDomain = 1]<br>BitMask 0x02 [reserved]<br>BitMask 0x04 [reserved]<br>BitMask 0x08 [reserved]<br>BitMask 0x10 [reserved]<br>BitMask 0x20 [reserved]<br>BitMask 0x40 [reserved]<br>BitMask 0x80 [reserved]              |
| <b>CRC_Status</b>                              |                   | 1             | 3                                      | 0..255   |
| <b>AUTOSAR TLV Sub-TLV: Status Not Secured</b> |                   |               |  |  |
| <b>High Nibble</b>                             | <b>Low Nibble</b> | <b>Octets</b> | <b>Offset</b>                          | <b>Value</b>   |
| <b>Type</b>                                    |                   | 1             | 0                                      | 0x51 [Status Not Secured]  |
| <b>Length</b>                                  |                   | 1             | 1                                      | 2  |

▽

△

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

▽

▽

△

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

### 5.3.1.3 Follow\_Up Message Header [AUTOSAR]

**[PRS\_TS\_00066]** [The `messageLength` of the `Follow_Up` Message Header has to be adapted according to the length of all existing *TLVs*.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

#### 5.3.1.4 AUTOSAR TLV Header

**[PRS\_TS\_00067]** [The AUTOSAR TLV Header has a multiplicity of 1.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

**[PRS\_TS\_00068]** [If an AUTOSAR TLV Header exists, at least one AUTOSAR Sub-TLV must exist as well.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

**[PRS\_TS\_00069]** [If an AUTOSAR TLV Header exists, the `lengthField` shall be adapted according the number of existing AUTOSAR Sub-TLVs.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

#### 5.3.1.5 AUTOSAR TLV Sub-TLVs

**[PRS\_TS\_00070]** [If an AUTOSAR Sub-TLV exists, it shall be placed after the AUTOSAR TLV Header.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

**[PRS\_TS\_00071]** [If more than one AUTOSAR Sub-TLV exists, each Sub-TLV shall be placed after the preceding Sub-TLV without gaps.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

**[PRS\_TS\_00072]** [If more than one AUTOSAR Sub-TLV exists, the position of each Sub-TLV is arbitrary.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

#### 5.3.1.6 AUTOSAR TLV Sub-TLV: Time Secured

**[PRS\_TS\_00074]** [The AUTOSAR Sub-TLV: Time Secured has a multiplicity of 1 and is only available, if CRC protection is required.] ([RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

**[PRS\_TS\_00075]** [If `MessageCompliance` is `FALSE` and `TLVFollowUpTimeSubTLV` is set to `TRUE`, the Time Master shall send a `FollowUp`, which contains an AUTOSAR Sub-TLV: Time Secured.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

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

**[PRS\_TS\_00076]** [The AUTOSAR Sub-TLV: Status has a multiplicity of 1 and can either be CRC protected (Status Secured) or not (Status Not Secured).] ([RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

**[PRS\_TS\_00077]** [If `MessageCompliance` is set to `FALSE` and `TLVFollowUpStatusSubTLV` is set to `TRUE`, the Time Master shall send a `FollowUp`, which contains an AUTOSAR Sub-TLV: Status.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

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

**[PRS\_TS\_00078]** [The AUTOSAR *Sub-TLV: UserData* has a multiplicity of 1 and can either be *CRC* protected (*UserData Secured*) or not (*UserData Not Secured*).] ([RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

**[PRS\_TS\_00079]** [If *MessageCompliance* is set to `FALSE` and *FollowUp* TLV User Data configuration is set to `TRUE`, the Time Master shall send a `FollowUp`, which contains an AUTOSAR *Sub-TLV: UserData*.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

**[PRS\_TS\_00081]** [The AUTOSAR *Sub-TLV: UserData* shall be read from the current incoming message consistently.] ([RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

**[PRS\_TS\_00082]** [The AUTOSAR *Sub-TLV: UserData* shall be written to the next outgoing message consistently.] ([RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

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

**[PRS\_TS\_00084]** [The AUTOSAR *Sub-TLV: OFS* has a multiplicity of 16 and can either be *CRC* protected (*OFS Secured*) or not (*OFS Not Secured*).] ([RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

**[PRS\_TS\_00085]** [The element `OfsTimeDomain` of the AUTOSAR *Sub-TLV: OFS* shall contain the Offset Time Domain identifier, which is in a range between 16 and 31.] ([RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

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

**[PRS\_TS\_00086]** [If *MessageCompliance* is set to `FALSE` and *TSynTLVFollowUpOFSSubTLV* is set to `TRUE`, the Time Master shall send a `FollowUp`, which contains at least one AUTOSAR *Sub-TLV: OFS*.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

**[PRS\_TS\_00088]** [The User Data of the AUTOSAR *Sub-TLV: OFS* shall be read from an incoming message consistently.] ([RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

**[PRS\_TS\_00089]** [The User Data of the AUTOSAR *Sub-TLV: OFS* shall be written to an outgoing message consistently.] ([RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#))

## 5.3.2 Body/Payload format

Placeholder for upcoming Autosar releases.

### **5.3.3 Data Types**

Refer to [1, IEEE 802.1 AS].

## **5.4 Message types**

Refer to [1, IEEE 802.1 AS].

### **5.4.1 Data Messages**

Refer to [1, IEEE 802.1 AS].

### **5.4.2 Control Messages**

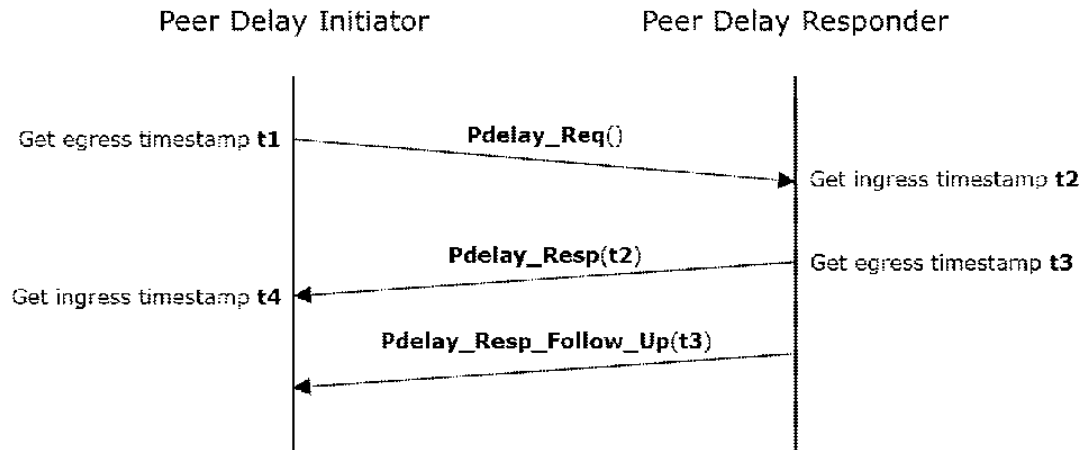
Refer to [1, IEEE 802.1 AS].

## **5.5 Services / Commands**

Placeholder for upcoming Autosar releases.

## 5.6 Sequences (lower layer)

### 5.6.1 Pdelay Protocol for Latency Calculation



**Figure 5.1: Propagation Delay Measurement (Pdelay)**

**[PRS\_TS\_00003]** [The Time Sync module shall use for latency calculation

- either static Pdelay values (GlobalTimePropagationDelay)
- or runtime-based values calculated by Pdelay\_Req, Pdelay\_Resp, Pdelay\_Resp\_Follow\_Up according to Figure 5.1: Propagation Delay Measurement (Pdelay)

depending on configuration of GlobalTimeTxPdelayReqPeriod](RS\_TS\_20048, RS\_TS\_20066)

**Note::** Due to the limitation [1] and [3] 1.2.2 "Limitations", it is sufficient that only the Time Slave initiates the Pdelay measurement

**[PRS\_TS\_00154]** [If GlobalTimeTxPdelayReqPeriod is not equal to 0 and if the Pdelay latency calculation result exceeds PdelayLatencyThreshold, the measured value shall be discarded and the previous value shall be kept.](RS\_TS\_20048)

**[PRS\_TS\_00004]** [A Pdelay\_Resp timeout or incomplete Pdelay protocol shall stop the latency calculation algorithm. In such cases, the device shall use the latest successful calculated latency value.](RS\_TS\_20048, RS\_TS\_20051)

**Note:** A timeout is detected, when sending the next subsequent Pdelay\_Req before receiving the Pdelay\_Resp resp. Pdelay\_Resp\_Follow\_Up belonging to the Pdelay\_Req before.

**[PRS\_TS\_00164]** [Time Master and Time Slave shall observe the `Pdelay` timeout as given by `PdelayRespAndRespFollowUpTimeout` , if a `Pdelay_Req` has been transmitted (waiting for `Pdelay_Resp`) or if a `Pdelay_Resp` has been received (waiting for `Pdelay_Resp_Follow_Up`). A value of 0 deactivates this timeout observation.]([RS\\_TS\\_20048](#), [RS\\_TS\\_20051](#))

**[PRS\_TS\_00210]** [If a reception timeout occurs (refer to [\[PRS\\_TS\\_00164\]](#)), any received `Pdelay_Resp` resp `Pdelay_Resp_Follow_Up` shall be ignored, until a new `Pdelay_Req` has been sent.]([RS\\_TS\\_20048](#), [RS\\_TS\\_20051](#))

**[PRS\_TS\_00140]** [If `GlobalTimeTxPdelayReqPeriod` equals 0, Time Master and Time Slave shall not measure the propagation delay. The Time Slave shall use a static value `GlobalTimePropagationDelay` as propagation delay instead.]([RS\\_TS\\_20066](#))

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

**[PRS\_TS\_00141]** [If `GlobalTimeTxPdelayReqPeriod` is greater than 0, Time Master and Time Slave shall cyclically measure the propagation delay using `Pdelay_Req`, `Pdelay_Resp`, `Pdelay_Resp_Follow_Up` as defined in [\[1, IEEE802.1 AS\]](#) chapter 11.1.2 "Propagation delay measurement".]([RS\\_TS\\_20048](#), [RS\\_TS\\_20066](#))

**[PRS\_TS\_00149]** [If `GlobalTimeTxPdelayReqPeriod` is greater than 0, Time Master and Time Slave shall cyclically measure the propagation delay only on that Time Domain with the lowest Time Domain ID and shall use this value to adjust all corresponding Time Bases.]([RS\\_TS\\_20048](#), [RS\\_TS\\_20066](#))

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

**[PRS\_TS\_00142]** [If `GlobalTimeTxPdelayReqPeriod` is greater than 0, `GlobalTimePropagationDelay` shall be used as default value for the propagation delay, until first valid propagation delay has been measured.]([RS\\_TS\\_20048](#), [RS\\_TS\\_20066](#))

**[PRS\_TS\_00011]** [If `GlobalTimeTxPdelayReqPeriod` is greater than 0, Time Master and Time Slave shall periodically transmit `Pdelay_Req` for latency calculation with the cycle `GlobalTimeTxPdelayReqPeriod` as defined in [\[1, IEEE 802.1 AS\]](#) chapter 11.1.2 "Propagation delay measurement".]([RS\\_TS\\_20048](#), [RS\\_TS\\_20066](#))

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



**[PRS\_TS\_00012]** [If `GlobalTimePdelayRespEnable` is set to `TRUE`, Time Master and Time Slave shall react to `Pdelay_Req` by transmitting `Pdelay_Resp` for latency calculation as defined in [1, IEEE 802.1 AS] chapter 11.1.2 "Propagation delay measurement".] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20066](#))

**[PRS\_TS\_00143]** [If `GlobalTimePdelayRespEnable` is set to `FALSE`, `Pdelay_Resp` and `Pdelay_Resp_Follow_Up` shall be omitted.] ([RS\\_TS\\_20066](#))

## 5.6.2 Acting as Time Master

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

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

### 5.6.2.1 Message Processing

**[PRS\_TS\_00050]** [The Time Master shall support the transmission of `Sync` and `Follow_Up` according as well as the transmission and reception of `Pdelay_Req`, `Pdelay_Resp` and `Pdelay_Resp_Follow_Up`.] ([RS\\_TS\\_20047](#), [RS\\_TS\\_20048](#))

**[PRS\_TS\_00016]** [The Time Master shall periodically transmit `Sync` with the cycle `GlobalTimeTxPeriod` as defined in [1, IEEE 802.1 AS] chapter 11.1.3 "Transport of time-synchronization information", if the `GLOBAL_TIME_BASE` bit within the `time-BaseStatus`, which is read from the corresponding Time Base, is set and `GlobalTimeTxPeriod` is not 0.] ([RS\\_TS\\_20047](#), [RS\\_TS\\_20048](#))

**[PRS\_TS\_00018]** [The `preciseOriginTimestamp` as calculated above, shall be used in the transmission of the `Follow_Up` as defined in [1, IEEE 802.1 AS] chapter 11.1.3 "Transport of time-synchronization information".] ([RS\\_TS\\_20048](#))

#### 5.6.2.1.1 Frame Debouncing

**[PRS\_TS\_00186]** [If multiple frames are triggered at the same time, the frames shall be sent in the following order:

1. `Sync`
2. `Follow_Up`
3. `Pdelay_Req`

4. Pdelay\_Resp, Pdelay\_Resp\_Follow\_Up  
](RS\_TS\_20047)

### 5.6.2.2 Message Field Calculation and Assembling

**[PRS\_TS\_00092]** [If MessageCompliance is set to FALSE, a Time Master shall add an AUTOSAR TLV to the Follow\_Up frame.](RS\_TS\_20061, RS\_TS\_20062, RS\_TS\_20063)

**[PRS\_TS\_00091]** [If MessageCompliance is set to FALSE, CRC\_SUPPORT shall be considered.](RS\_TS\_20061)

**[PRS\_TS\_00093]** [Depending on CRC\_SUPPORT the Follow\_Up.TLV[AUTOSAR].Sub-TLV.Type shall be:](RS\_TS\_20061)

Follow\_Up Message Header [IEEE 802.1AS]

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

#### 5.6.2.2.1 SGW Calculation

**[PRS\_TS\_00094]** [The SGW value (Time Gateway synchronization status) shall be mapped to the Status element of the AUTOSAR Sub-TLV: Status resp. the AUTOSAR Sub-TLV: OFS. If the SYNC\_TO\_GATEWAY is set, the SGW value shall be SyncToSubDomain. Otherwise, it shall be SyncToGTM.](RS\_TS\_20052, RS\_TS\_20054)

#### 5.6.2.2.2 OFS Calculation

**[PRS\_TS\_00095]** [The Time Master of an Offset Time Base shall send the "second" part of the Offset Time Base value via the ofsTimeSec element of the corresponding AUTOSAR Sub-TLV: OFS and the "nanosecond" part of the Offset Time Base value via the ofsTimeNSec element of the corresponding AUTOSAR Sub-TLV: OFS](RS\_TS\_20063)

### 5.6.2.2.3 CRC Calculation

**[PRS\_TS\_00097]** [The `DataID` shall be calculated as:  $\text{DataID} = \text{DataIDList} [\text{Follow\_Up}.\text{sequenceId} \bmod 16]$ , where `DataIDList` is given by configuration for the `Follow_Up`.] ([RS\\_TS\\_20061](#))

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

**[PRS\_TS\_00182]** [If applying the *CRC* calculation on multibyte values, the byte order shall be such, that the byte containing the most significant bit of the value shall be used first.] ([RS\\_TS\\_20061](#))

**[PRS\_TS\_00184]** [If applying the *CRC* calculation on multibyte message data, the byte order shall be in ascending order of the octets, i.e., the octet with the lowest offset shall be used first.] ([RS\\_TS\\_20061](#))

#### 5.6.2.2.3.1 AUTOSAR TLV *Sub-TLV*: Time Secured

**[PRS\_TS\_00098]** [If `GlobalTimeTxCrcSecured` is `CRC_SUPPORTED`, the Time Master shall write the contents of `CrcTimeFlagsTxSecured` to `CRC_Time_Flags` acc. to the following rule.] ([RS\\_TS\\_20061](#))

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

**[PRS\_TS\_00099]** [If `CrcTimeFlagsTxSecured` is supported, the Time Master shall calculate the *CRC* for `CRC_Time_0` by considering the contents of `CRC_Time_Flags` itself, the contents of the dependent fields as defined in `CrcTimeFlagsTxSecured` acc. to the rule in the table below and the `DataID`. The data elements used for the calculation of the *CRC* shall apply the following order:

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

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

](RS\_TS\_20061)

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

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

**[PRS\_TS\_00100]** [If `GlobalTimeTxCrcSecured` is set to `CRC_SUPPORTED`, the Time Master shall calculate the `CRC` for `CRC_Time_1` by considering the contents of `CRC_Time_Flags` itself, the contents of the dependent fields as defined in `Crc-TimeFlagsTxSecured` acc. to the rule in the table below and the `DataID`. The data elements used for the calculation of the `CRC` shall apply the following order:

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

](RS\_TS\_20061)

|                                      | <b>For CRC_Time_1 calculation considered contents:</b> |                                |
|--------------------------------------|--|--------------------------------|
| <b>If CRC_Time_Flags is set to 1</b> | <b>Follow_Up Message Header</b>                        | <b>Follow_Up Message Field</b> |
| BitMask 0x01                         | messageLength  | n.a.                           |
| BitMask 0x02                         | n.a.   | n.a.                           |
| BitMask 0x04                         | <a href="#">CrcCorrectionField.</a>                    | n.a.                           |
| BitMask 0x08                         | n.a.   | n.a.                           |
| BitMask 0x10                         | sequenceId   | n.a.                           |
| BitMask 0x20                         | n.a.   | n.a.                           |
| BitMask 0x40                         | n.a.   | n.a.                           |
| BitMask 0x80                         | n.a.   | n.a.                           |

**Note:** CRC\_Time\_Flags has the same value as the configuration item [CrcTime-FlagsTxSecured](#).

#### 5.6.2.2.3.2 AUTOSAR TLV Sub-TLV: Status secured

**[PRS\_TS\_00101]** [If [GlobalTimeTxCrcSecured](#) is set to CRC\_SUPPORTED, the Time Master shall calculate the CRC for CRC\_STATUS by considering the contents of Status and DataID (in this order).] ([RS\\_TS\\_20061](#))

#### 5.6.2.2.3.3 AUTOSAR TLV Sub-TLV: UserData secured

**[PRS\_TS\_00102]** [If [GlobalTimeTxCrcSecured](#) is set to CRC\_SUPPORTED, the Time Master shall calculate the CRC for CRC\_UserData by considering the contents of UserDataLength, UserByte\_0, UserByte\_1, UserByte\_2 and DataID (in this order).] ([RS\\_TS\\_20061](#))

#### 5.6.2.2.3.4 AUTOSAR TLV Sub-TLV: OFS secured

**[PRS\_TS\_00103]** [If [GlobalTimeTxCrcSecured](#) is set to CRC\_SUPPORTED, the Time Master shall calculate the CRC for CRC\_OFS by considering the contents of OfstimeDomain, OfstimeSec, OfstimeNsec, Status, UserDataLength, UserByte\_0, UserByte\_1, UserByte\_2 and DataID (in this order).] ([RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#), [RS\\_TS\\_20063](#))

#### 5.6.2.2.4 Sequence Counter Calculation

**[PRS\_TS\_00187]** [The Sequence Counter (sequenceId) of a Sync and Pdelay\_Req message shall be initialized with 0.] ([RS\\_TS\\_20061](#))

**[PRS\_TS\_00188]** [The Peer Delay Initiator shall increment the Sequence Counter of a `Pdelay_Req` message by 1 on each transmission request for a `Pdelay_Req` message. The Sequence Counter shall wrap around at 65535 to 0 again.] ([RS\\_TS\\_20061](#))

**[PRS\_TS\_00189]** [The Time Master shall increment the Sequence Counter of a `Sync` message by 1 on each transmission request for a `Sync` message of a given Time Domain. The Sequence Counter shall wrap around at 65535 to 0 again.] ([RS\\_TS\\_20061](#))

**[PRS\_TS\_00190]** [The Time Master shall set the Sequence Counter (`sequenceId`) value for a `Follow_Up` message to the Sequence Counter (`sequenceId`) value of the corresponding `Sync` message.] ([RS\\_TS\\_20061](#))

**[PRS\_TS\_00191]** [The Peer Delay Responder shall set the Sequence Counter (`sequenceId`) value for a `Pdelay_Resp` and `Pdelay_Resp_Follow_Up` message to the Sequence Counter (`sequenceId`) value of the corresponding `Pdelay_Req` message.] ([RS\\_TS\\_20061](#))

#### 5.6.2.2.5 Message Assembling

**[PRS\_TS\_000104]** [For each transmission of a Time Synchronization message, the Time Synchronization module shall assemble the message as follows:

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

For 4: Calculate `CRC` (configuration dependent) and copy all data to the appropriate position within the related message] ([RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#), [RS\\_TS\\_20063](#))

### 5.6.3 Acting as Time Slave

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

#### 5.6.3.1 Message processing

**[PRS\_TS\_00023]** [The Time Slave shall support the reception of `Sync` and `Follow_Up` according [1, IEEE 802.1 AS] as well as the transmission and reception of `Pdelay_Req`, `Pdelay_Resp` and `Pdelay_Resp_Follow_Up`, [\[PRS\\_TS\\_00140\]](#), [\[PRS\\_TS\\_00141\]](#),[\[PRS\\_TS\\_00004\]](#).] ([RS\\_TS\\_20048](#))

**[PRS\_TS\_00025]** [For each configured Time Slave the Ethernet module shall observe the reception timeout `GlobalTimeFollowUpTimeout` between the `Sync` and its `Follow_Up`. If the reception timeout occurs, the sequence shall be reset (i.e. waiting for a new `Sync`). A value of 0 deactivates this timeout observation.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20051](#))

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

### 5.6.3.2 Message Field Validation and Disassembling

**[PRS\_TS\_00105]** [If `MessageCompliance` is set to `FALSE`, `RxCrcValidated` shall be considered.] ([RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#), [RS\\_TS\\_20063](#))

**[PRS\_TS\_00106]** [If `MessageCompliance` is set to `FALSE`, a Time Slave shall check if an AUTOSAR `TLV` in the `Follow_Up` frame exists.] ([RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#), [RS\\_TS\\_20063](#))

**[PRS\_TS\_00107]** [The `CRC` of the `Follow_Up TLV` shall be validated, depending on `RxCrcValidated` and the `Follow_Up.TLV[AUTOSAR].Sub-TLV.Type` acc. to:] ([RS\\_TS\\_20061](#))

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

**[PRS\_TS\_00108]** [The `CRC` of the `Follow_Up TLV` shall be ignored, if `RxCrcValidated` is set to `CRC_IGNORED` and the `Follow_Up.TLV[AUTOSAR].Sub-TLV.Type` contains any of the following defined values:] ([RS\\_TS\\_20061](#))

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

**[PRS\_TS\_00109]** [The *CRC* of the `Follow_Up TLV` shall be either validated or not validated, if `RxCrcValidated` is set to `CRC_OPTIONAL` and the `Follow_Up.TLV [AUTOSAR].Sub-TLV.Type` contains any of the following defined values:] ([RS\\_TS\\_20061](#))

|                             | Sub-TLV.Type                              |   |
|-----------------------------|---|---|
| <code>RxCrcValidated</code> | CRC_OPTIONAL                              |   |
|                             | CRC shall be validated                    | CRC shall not be validated                |
|                             | 0x28 Sub-TLV: Time Secured is CRC secured | n.a.                                      |
|                             | 0x50 Sub-TLV: Status is CRC secured       | 0x51 Sub-TLV: Status is not CRC secured   |
|                             | 0x60 Sub-TLV: UserData is CRC secured     | 0x61 Sub-TLV: UserData is not CRC secured |
|                             | 0x44 Sub-TLV: OFS is CRC secured          | 0x34 Sub-TLV: OFS is not CRC secured      |

### 5.6.3.2.1 SGW Calculation

**[PRS\_TS\_00211]** [If `MessageCompliance` is set to `TRUE` the `SYNC_TO_GATEWAY` bit within `timeBaseStatus` shall be set to zero.] ([RS\\_TS\\_20054](#))

**[PRS\_TS\_00156]** [For a Synchronized Time Base and if `MessageCompliance` is set to `FALSE` the `SGW` value (Time Gateway synchronization status) shall be retrieved from the Status element of the *AUTOSAR Sub-TLV: Status* if an *AUTOSAR TLV* in the `Follow_Up` message exists and if this *Sub-TLV* is part of the *AUTOSAR TLV*. If the `SGW` value is set to `SyncToSubDomain`, the `SYNC_TO_GATEWAY` bit within `timeBaseStatus` shall be set to one. Otherwise, it shall be set to zero.] ([RS\\_TS\\_20054](#))

**Note:** Since a Global Time Master will not set the Time Gateway synchronization status to `SYNC_TO_GATEWAY` it is superfluous to transmit an *AUTOSAR Sub-TLV: Status* in this case.



[PRS\_TS\_00212] [If `MessageCompliance` is set to `FALSE` and if an *AUTOSAR Sub-TLV*: `Status` in the `Follow_Up` message does not exist, the `SYNC_TO_GATEWAY` bit within `timeBaseStatus` shall be set to zero.] ([RS\\_TS\\_20054](#))

[PRS\_TS\_00213] [If `MessageCompliance` is set to `FALSE` and if an *AUTOSAR TLV* in the `Follow_Up` message exists the `SGW` value (Time Gateway synchronization status) shall be retrieved from the `Status` element of each *AUTOSAR Sub-TLV*: `OFS` that is part of the *AUTOSAR TLV*. If the `SGW` value is set to `SyncToSubDomain`, the `SYNC_TO_GATEWAY` bit within `timeBaseStatus` shall be set to one. Otherwise, it shall be set to zero.] ([RS\\_TS\\_20054](#))

### 5.6.3.2.2 OFS Calculation

[PRS\_TS\_00110] [The Time Slave of an Offset Time Base shall calculate the Offset Time Base from the `OfsTimeSec` element of the corresponding *AUTOSAR Sub-TLV*: `OFS` and the `OfsTimeNSec` element of the corresponding *AUTOSAR Sub-TLV*: `OFS`.] ([RS\\_TS\\_20063](#))

### 5.6.3.2.3 CRC Validation

[PRS\_TS\_00112] [The `DataID` shall be calculated as:  $\text{DataID} = \text{DataIDList} [\text{Follow\_Up.sequenceId} \bmod 16]$ , where `DataIDList` is given by configuration for the `Follow_Up`.] ([RS\\_TS\\_20061](#))

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

[PRS\_TS\_00183] [If applying the *CRC* calculation on multibyte values, the byte order shall be such that the byte containing the most significant bit of the value shall be used first.] ([RS\\_TS\\_20061](#))

[PRS\_TS\_00185] [If applying the *CRC* calculation on multibyte message data, the byte order shall be in ascending order of the octets, i.e., the octet with the lowest offset shall be used first.] ([RS\\_TS\\_20061](#))

#### 5.6.3.2.3.1 AUTOSAR TLV Sub-TLV: Time Secured

[PRS\_TS\_00157] [If `RxCrcValidated` is set to `CRC_VALIDATED` or `CRC_OPTIONAL`, the Time Slave shall validate the *CRC* as defined in `CrcFlagsRxValidated` acc. to the following rule.] ([RS\\_TS\\_20061](#))

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

**[PRS\_TS\_00113]** [If `RxCrcValidated` is set to `CRC_VALIDATED` or `CRC_OPTIONAL`, , the Time Slave shall validate the *CRC* for `CRC_Time_0` by considering the contents of `CRC_Time_Flags` itself, the contents of the dependent fields as defined in `CrcFlagsRxValidated` acc. to the rule in the table below and the `DataID`. The data elements used for the calculation and thus validation of the *CRC* shall apply the following order:

1. the value of `CRC_Time_Flags`
2. the `domainNumber` inside the `Follow_Up Message Header`, if `CrcDomainNumber` is set to TRUE
3. the `preciseOriginTimestamp` inside the `Follow_Up Message Field`, if `CrcPreciseOriginTimestamp` is set to TRUE
4. the `DataID` (refer to [PRS\_TS\_00112])
5. the `sourcePortIdentity` inside the `Follow_Up Message Header`, if `CrcSourcePortIdentity` is set to TRUE

]([RS\\_TS\\_20061](#))

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

**[PRS\_TS\_00114]** [If `RxCrcValidated` is set to `CRC_VALIDATED` or `CRC_OPTIONAL`, the Time Slave shall validate the *CRC* for `CRC_Time_1` by considering the

contents of `CRC_Time_Flags` itself, the contents of the dependent fields as defined in `CrcFlagsRxValidated` acc. to the rule in the table below and the `DataID`. The data elements used for the calculation and thus validation of the `CRC` shall apply the following order:

1. the value of `CRC_Time_Flags`
2. the `messageLength` inside the `Follow_Up Message Header`, if `CrcMessageLength` is set to `TRUE`
3. the `CrcCorrectionField` inside the `Follow_Up Message Header`, if `CrcCorrectionField` is set to `TRUE`
4. the `sequenceId` inside the `Follow_Up Message Field`, if `CrcSequenceId` is set to `TRUE`
5. the `DataID` (refer to [PRS\_TS\_00112])

](RS\_TS\_20061)

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

#### 5.6.3.2.3.2 AUTOSAR TLV Sub-TLV: Status secured

[PRS\_TS\_00115] [If `RxCrcValidated` is set to `CRC_VALIDATED` or `CRC_OPTIONAL`, the Time Slave shall validate the `CRC` for `CRC_Status` by considering the contents of `Status` and `DataID` (in this order).](RS\_TS\_20061)

#### 5.6.3.2.3.3 AUTOSAR TLV Sub-TLV: UserData secured

[PRS\_TS\_00116] [If `RxCrcValidated` is set to `CRC_VALIDATED` or `CRC_OPTIONAL`, the Time Slave shall validate the `CRC` for `CRC_UserData` by considering the contents of `UserDataLength`, `UserByte_0`, `UserByte_1`, `UserByte_2` and `DataID` (in this order).](RS\_TS\_20061)

#### 5.6.3.2.3.4 AUTOSAR TLV Sub-TLV: OFS secured

**[PRS\_TS\_00117]** [If `RxCrcValidated` is set to `CRC_VALIDATED` or `CRC_OPTIONAL`, the Time Slave shall validate the `CRC` for `CRC_OFS` by considering the contents of `OfsTimeDomain`, `OfsTimeSec`, `OfsTimeNSec`, `Status`, `UserDataLength`, `UserByte_0`, `UserByte_1`, `UserByte_2` and `DataID` (in this order).] ([RS\\_TS\\_20061](#), [RS\\_TS\\_20063](#))

#### 5.6.3.2.4 Sequence Counter (sequenceId) Validation

**[PRS\_TS\_00192]** [If the Sequence Counter (`sequenceId`) of a received `Pdelay_Resp` message does not match the Sequence Counter (`sequenceId`) of the corresponding `Pdelay_Req` message, the Peer Delay Initiator shall ignore the `Pdelay_RResp` message.] ([RS\\_TS\\_20061](#))

**[PRS\_TS\_00193]** [The Peer Delay Initiator shall ignore a `Pdelay_Resp` message, if the `Pdelay_Resp` message has not been received within the timeout interval `GlobalTimePdelayRespAndRespFollowUpTimeout`.] ([RS\\_TS\\_20061](#))

**[PRS\_TS\_00194]** [If the Sequence Counter (`sequenceId`) of a received `Pdelay_Resp_Follow_Up` message does not match the Sequence Counter (`sequenceId`) of the transmitted `Pdelay_Req` message, the Peer Delay Initiator shall ignore the received `Pdelay_Resp_Follow_Up` message.] ([RS\\_TS\\_20061](#))

**[PRS\_TS\_00195]** [The Peer Delay Initiator shall discard the content of a `Pdelay_Resp` message, if no `Pdelay_Resp_Follow_Up` message with a matching Sequence Counter (`sequenceId`) has been received within the timeout interval `GlobalTimePdelayRespAndRespFollowUpTimeout`.] ([RS\\_TS\\_20061](#))

**[PRS\_TS\_00196]** [ If the Sequence Counter (`sequenceId`) of a received `Follow_Up` message does not match the Sequence Counter (`sequenceId`) of the previously received Sync message of the same Time Domain (`domainNumber`), the Time Slave shall ignore the `Follow_Up` message.] ([RS\\_TS\\_20061](#))

**[PRS\_TS\_00197]** [If no `Follow_Up` message with a matching Sequence Counter (`sequenceId`) and Time Domain (`domainNumber`) has been received within the timeout interval `GlobalTimeFollowUpTimeout`, the Time Slave shall discard the contents of the already received Sync message.] ([RS\\_TS\\_20061](#))

**[PRS\_TS\_00198]** [If the Sequence Counter Jump Width between two consecutive Sync messages of the same Time Domain (`domainNumber`) is 0 or greater than `GlobalTimeSequenceCounterJumpWidth`, the Time Slave shall ignore the Sync message. If `GlobalTimeSequenceCounterJumpWidth` is set to its maximum allowed value, the Time Slave shall accept any jump width greater than 0.] ([RS\\_TS\\_20061](#))

**[PRS\_TS\_00199]** [A Time Slave shall check the Sequence Counter (`sequenceId`) of a received Sync message per Time Domain (`domainNumber`) against the config-

ured value of `GlobalTimeSequenceCounterJumpWidth`, unless it is the first Sync message after Initialization or after a Synchronization Timeout.]([RS\\_TS\\_20061](#))

**Note:** A Synchronization Timeout means, that a Time Slave could not (re-)synchronize within a given timeout interval due to missing or invalid `Sync` or `Follow_up` messages.

**Note:** There are scenarios when it makes sense to skip the check of the Sequence Counter Jump Width, e.g. at startup (Time Slaves start asynchronously to the Time Master) or after a message timeout to allow for Sequence Counter (re-)synchronization. In case of a timeout the error has been detected already by the timeout monitoring, there is no benefit in generating a subsequent error by the jump width check.

**Note:** During Time Base update timeout the Sequence Counter validation will still discard messages with a Sequence Counter Jump Width being zero (i.e., stuck Sequence Counter).

#### 5.6.3.2.5 Message Disassembling

[**PRS\_TS\_00118**] [If the Type of a *Sub-TLV* cannot be recognized at the receiver side, it shall be ignored and the next subsequent *Sub-TLV* shall be evaluated.]([RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#), [RS\\_TS\\_20063](#))

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

[**PRS\_TS\_00119**] [If any of the following conditions is not met, a Time Slave shall discard a received `Sync` or `Follow_Up` message:

1. Validation of Sequence Counter (`sequenceId`) is successful (refer to: [[PRS\\_TS\\_00196](#)], [[PRS\\_TS\\_00197](#)], [[PRS\\_TS\\_00198](#)], [[PRS\\_TS\\_00199](#)]).
2. If `Follow_Up`: `Follow_Up.TLV[AUTOSAR].Sub-TLV.Type` matches depending on configuration of `RxCrcValidated`
3. The Time Domain matches to the defined Time Domain range for each `domain-Number` resp. to the element `OfsTimeDomain` of the AUTOSAR *Sub-TLV*: OFS (configuration dependent).
4. The Time Domain matches to one of the configured Time Domains
5. If `Follow_Up`: The element `OfsTimeNSec` of the AUTOSAR *Sub-TLV*: OFS is within the allowed range of 0..999999999.
6. If `Follow_Up`: All CRCs are successfully validated depending on the configuration of `RxCrcValidated` and `CrcFlagsRxValidated`.

]([RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#), [RS\\_TS\\_20063](#))

**[PRS\_TS\_00120]** [For each received Time Synchronization message, the Time synchronization protocol shall disassemble the message after successful validation.] ([RS\\_TS\\_20061](#), [RS\\_TS\\_20062](#), [RS\\_TS\\_20063](#))

## 5.7 Time measurement with Switches

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

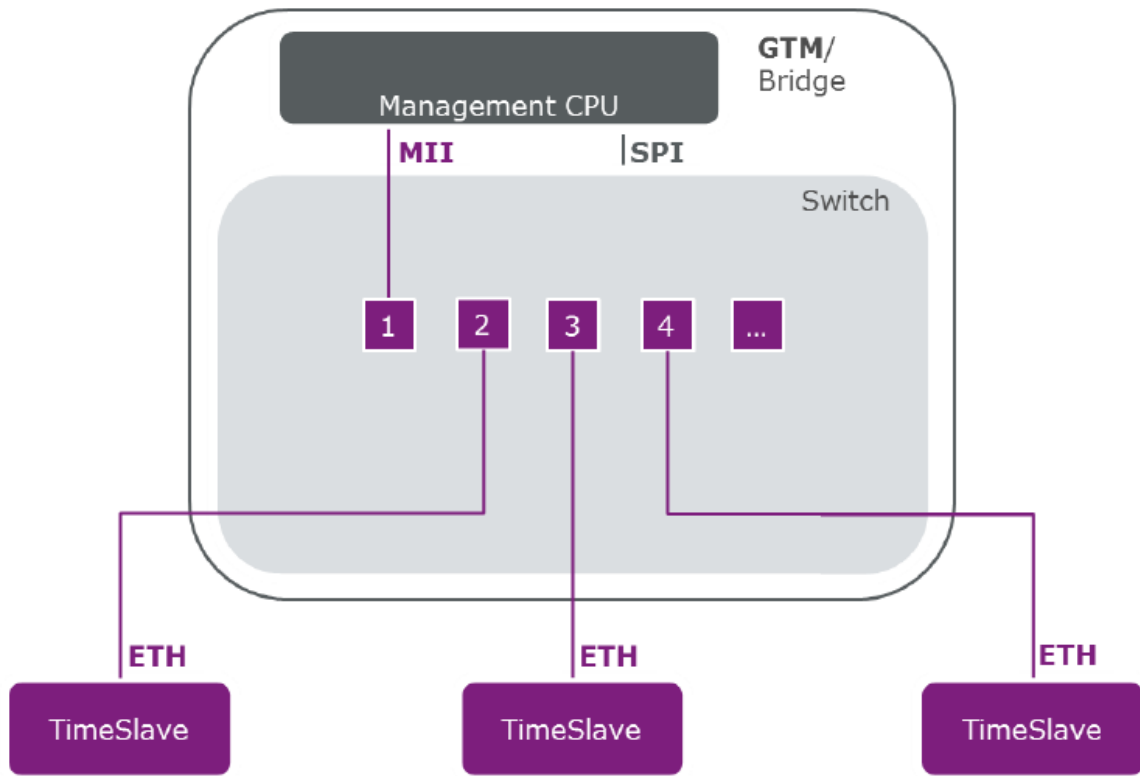
1. Endpoints directly working on a local Ethernet-Controller
2. Time Gateways, resp. Time Aware Bridges, where the local Ethernet-Controller connects to an external Switch device.

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

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

**[PRS\_TS\_00053]** [Time measurement with Switches supports the use case "Time Aware Bridge with GTM as Management CPU"]

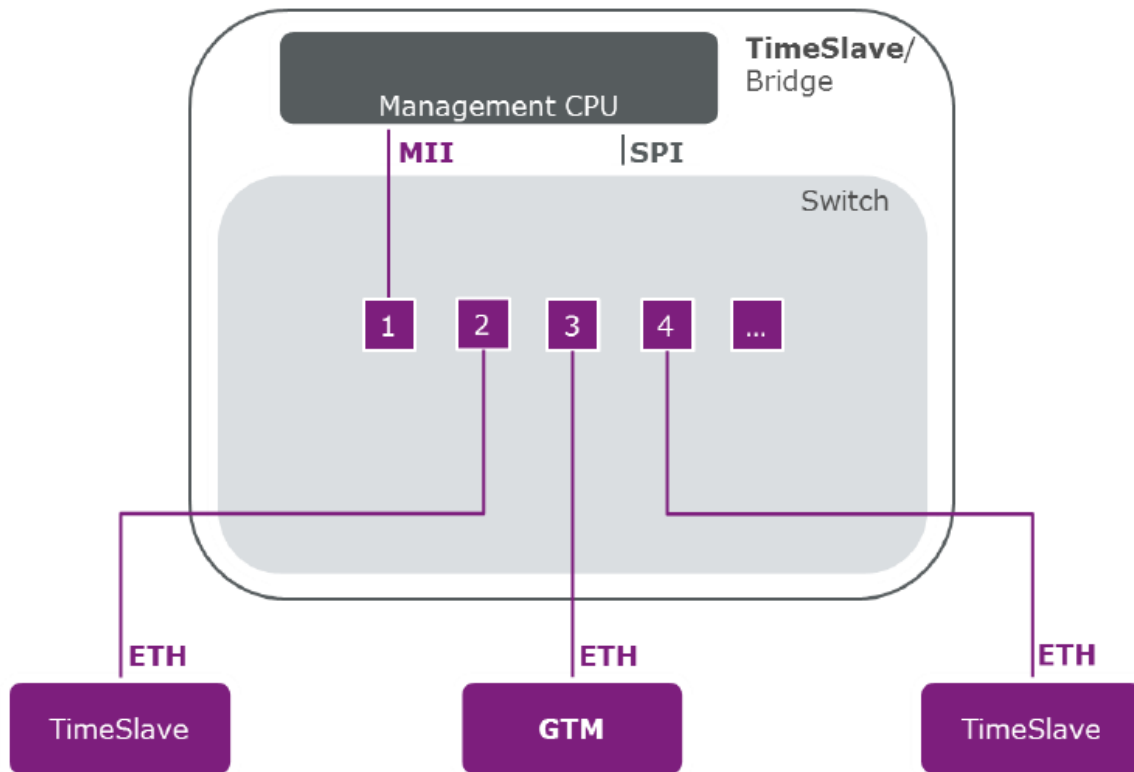
like shown in Figure [Figure 5.2.](#) ([RS\\_TS\\_20048](#), [RS\\_TS\\_20059](#))



**Figure 5.2: Time Aware Bridge with GTM as Management CPU**

**[PRS\_TS\_00054]** [Time measurement with Switches supports the use case "Time Aware Bridge with GTM not as Management

CPU" like shown in [Figure 5.3.](#) ([RS\\_TS\\_20048](#), [RS\\_TS\\_20059](#))



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

## 5.8 Pdelay and Time Synchronization measurement point

**[PRS\_TS\_00055]** [The path delay measurement will be done always as Port-to-Port measurement like specified in in [1, IEEE 802.1 AS] chapter 11.1.2 Propagation delay measurement for the device external Ethernet path.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20059](#))

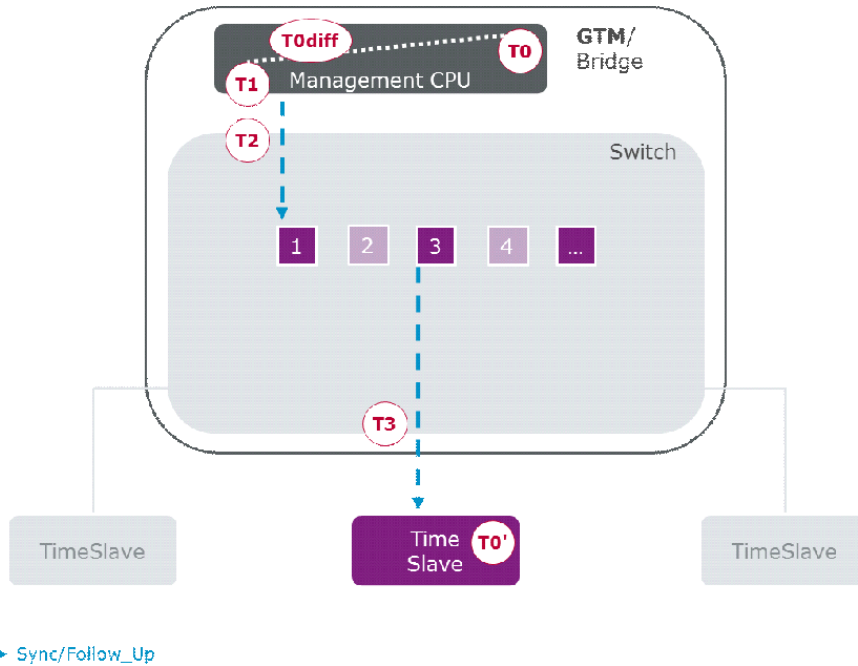
**[PRS\_TS\_00056]** [The inner delay of the Ethernet path (Residence Time) is determined at the time where `Sync` is received and transmitted, by using the message specific ingress and egress timestamps.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20059](#))

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

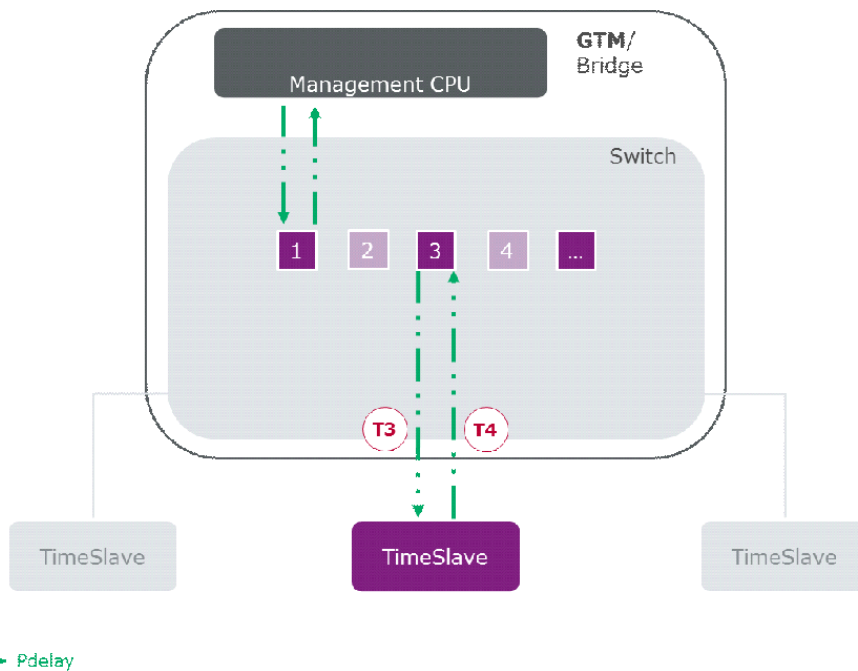


### 5.9 Time Aware Bridge with GTM as Management CPU

[PRS\_TS\_00057] [Time measurement with Switches supporting the use case "Time Aware Bridge with GTM as Management CPU" following the given timestamping points like shown in Figure 5.4 and Figure 5.5] (RS\_TS\_20048, RS\_TS\_20059)



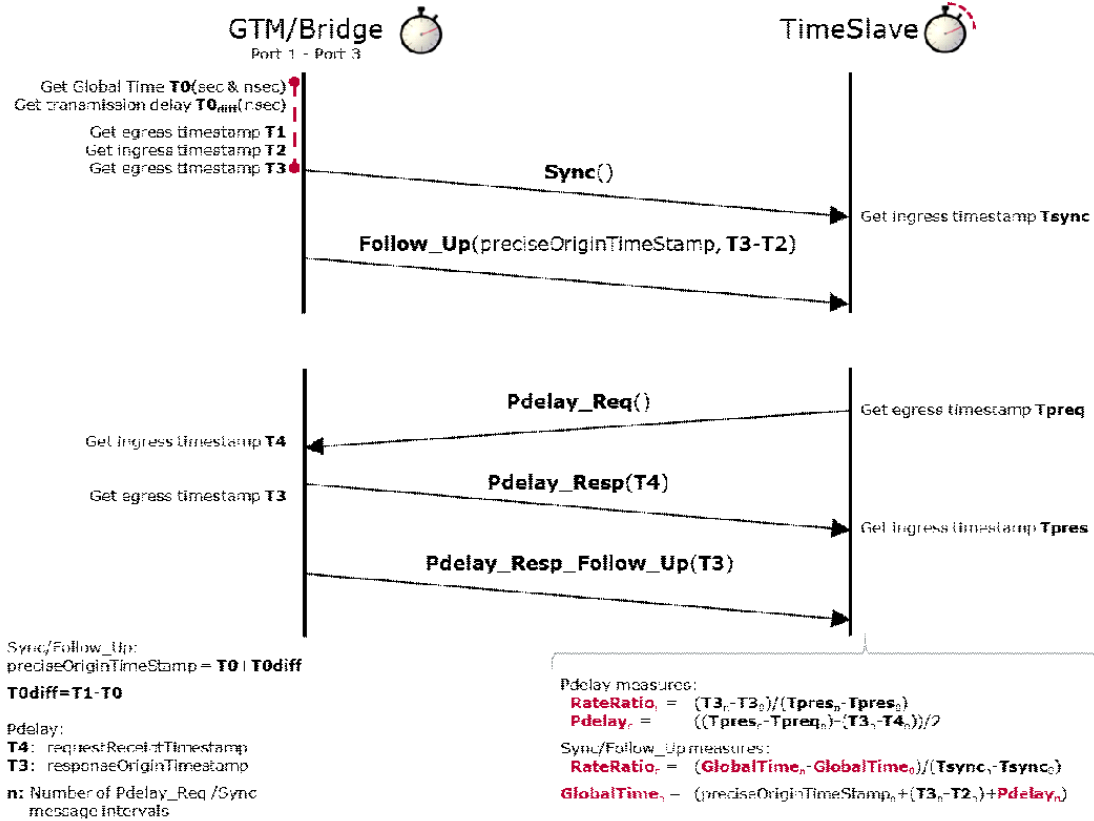
**Figure 5.4: Sync/Follow\_Up message flow with Timestamping points for Sync for Time Aware Bridge with GTM as Management CPU**



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

**Note:** The picture [Figure 5.4](#) and [Figure 5.5](#) shows an example Port selection as simplification.

**[PRS\_TS\_00058]** [Time measurement with Switches supporting the use case "Time Aware Bridge with GTM as Management CPU" considers the inner Switch delay by a modification of the [CrcCorrectionField](#) as well as Pdelay timestamping for requestReceiptTimestamp and responseOriginTimestamp like shown in [Figure 5.6](#).] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20059](#))



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

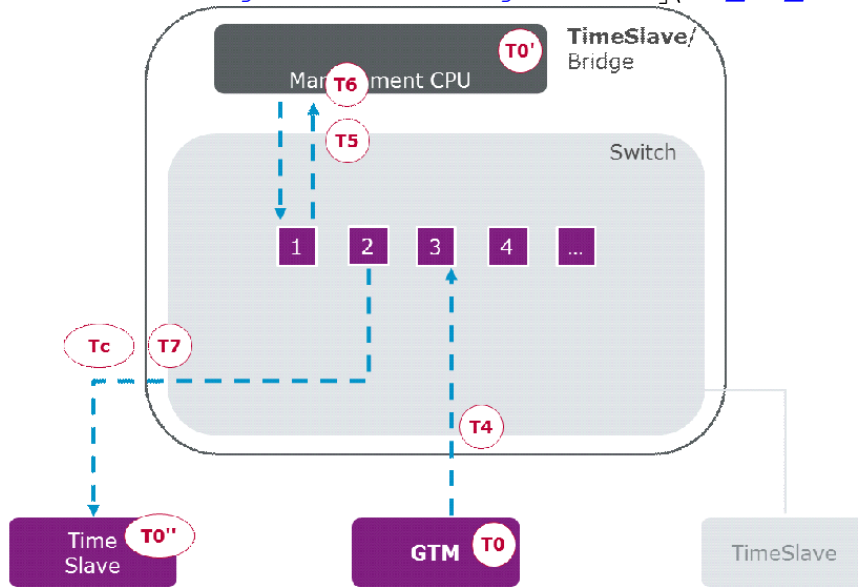
**Note:** The calculation in [Figure 5.6](#) shows an example Port selection as simplification.

**[PRS\_TS\_00166]** [If [GlobalTimeUplinkToTxSwitchResidenceTime](#) is set to 0, the Ethernet module shall ignore this parameter and measure the inner delay of the Switch egress Ethernet path (Uplink to Tx Residence Time ( $T3 - T2$ )) by using always the ingress ( $T2$ ) and egress ( $T3$ ) timestamp as given in [Figure 5.6](#).] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20059](#))

**[PRS\_TS\_00167]** [If [GlobalTimeUplinkToTxSwitchResidenceTime](#) is greater than 0, the Ethernet module shall use this parameter as value for the inner delay of the Switch egress Ethernet path (Uplink to Tx Residence Time ( $T3 - T2$ )) instead of using the measurement method described in [\[PRS\\_TS\\_00166\]](#).] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20059](#))

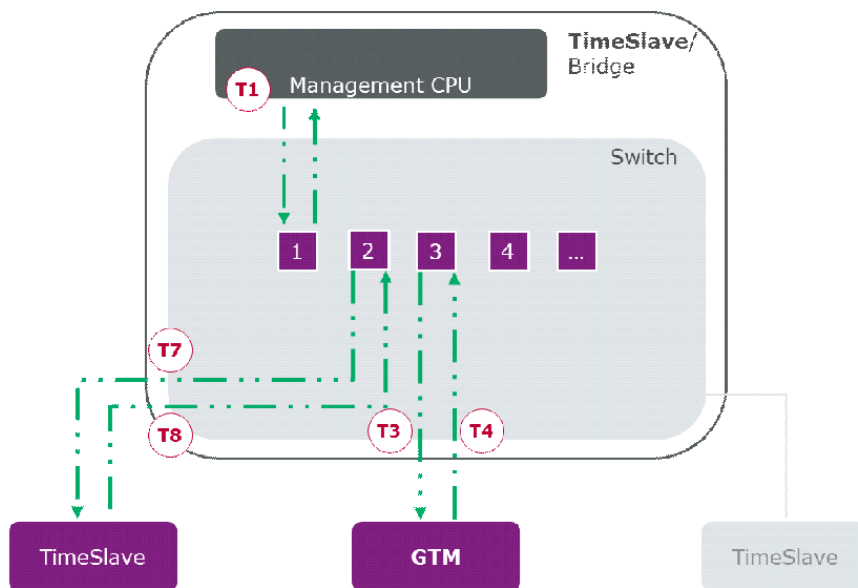
### 5.10 Time Aware Bridge with GTM not as Management CPU

[PRS\_TS\_00059] [Time measurement with Switches supporting the use case Time Aware Bridge with GTM not as Management CPU following the given timestamping points like shown in Figure 5.7 and Figure 5.8.] (RS\_TS\_20048, RS\_TS\_20059)



Legend:  
- - -> Sync/Follow\_Up

**Figure 5.7: Sync/Follow\_Up message flow with Timestamping points for Sync for Time Aware Bridge with GTM not as Management CPU**



Legend:  
- . -> Pdelay

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

**[PRS\_TS\_00060]** [Time measurement with Switches supporting the use case Time Aware Bridge with GTM not as Management CPU considers the inner Switch delay by a modification of the `CrcCorrectionField` as well as `Pdelay` timestamping for `requestReceiptTimestamp` and `responseOriginTimestamp`.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20059](#))

**[PRS\_TS\_00207]** [If the `Follow_Up` message contains an AUTOSAR TLV, which contains a `Sub-TLV: Time Secured` it shall be checked, if the element `CRC_Time_Flags` contains `BitMask 0x04` (i.e., the content of `CrcCorrectionField` is CRC protected). If this bit is set then the validation of the `CRC_Time_1` element shall be done as follows: The CRC Validation shall be done as specified in section 5.6.3.2.3. The data elements used for the calculation and thus validation of the CRC shall be applied with the following order:

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

] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20059](#))

**[PRS\_TS\_00208]** [If the validation CRC validation of an AUTOSAR TLV fails, the `Follow_Up` message shall be dropped instead of being forwarded.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20059](#))

**[PRS\_TS\_00209]** [If the validation CRC validation of an AUTOSAR TLV is successful, the `CrcCorrectionField` shall be modified and the element `CRC_Time` inside the `Sub-TLV: Time Secured` shall be calculated according to the content of the `CRC_Time_Flags` element.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20059](#))

**[PRS\_TS\_00168]** [If `rx_residence_time` is set to 0, the Time Synchronization over Ethernet shall ignore this parameter and measure the inner delay of the Switch ingress Ethernet path (Rx to Uplink Residence Time (T5 - T4)) by using always the ingress (T4) and egress (T5) timestamp.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20059](#))

**[PRS\_TS\_00171]** [If `rx_residence_time` is greater than 0, the Time Synchronization over Ethernet shall use this parameter as value for the inner delay of the Switch ingress Ethernet path (Rx to Uplink Residence Time (T5 - T4)) instead of using the measurement method.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20059](#))

**[PRS\_TS\_00169]** [If `rx_residence_time` and `tx_residence_time` are set to 0, the Ethernet module shall ignore both parameter and measure the inner delay of the Switch ingress and egress Ethernet path (Rx to Uplink and Uplink to Tx Residence Time (T7 to T4)) by using always the ingress (T4) and egress (T7) timestamp.] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20059](#))

**[PRS\_TS\_00170]** [If `rx_residence_time` and `tx_residence_time` are greater than 0, the Ethernet module shall use the sum of both parameter for the value of the inner delay of the Switch ingress and egress Ethernet path (Rx to Uplink and Uplink to Tx Residence Time (T7 to T4)) instead of using the measurement method] ([RS\\_TS\\_20048](#), [RS\\_TS\\_20059](#))

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

## 5.11 Error messages

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

## 6 Configuration parameters

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

| Name                              | Description  |
|-----------------------------------|--|
| CRCSupport                        | represents whether the configuration is supported or not   |
| <code>rx_residence_time</code>    | This parameter is specifying the default value used for the residence time   |
| <code>tx_residence_time</code>    | This parameter is specifying the default value used for the residence time   |
| FramePrio                         | This optional parameter, if present, indicates the priority of outgoing messages, if sent via VLAN (used for the 3-bit PCP field of the VLAN tag). If this optional parameter is not present, frames are sent without a priority and VLAN field. |
| GlobalTimeTxPdelayReqPeriod       | This parameter represents configuration of the TX period for Pdelay_Req messages. A value of 0 disables the cyclic Pdelay measurement.   |
| PdelayLatencyThreshold            | Threshold for calculated Pdelay. If a measured Pdelay exceeds PdelayLatencyThreshold, this value is discarded.   |
| PdelayRespAndResp-FollowUpTimeout | Timeout value for Pdelay_Resp and Pdelay_Resp-Follow-Up after a Pdelay_Req has been transmitted resp. a Pdelay_Resp has been received. A value of 0 deactivates this timeout observation.  |

|                              |   |
|------------------------------|---|
| GlobalTimePropagationDelay   | If cyclic propagation delay measurement is enabled, this parameter represents the default value of the propagation delay until the first actually measured propagation delay is available. If cyclic propagation delay measurement is disabled, this parameter replaces a measured propagation delay by a fixed value.  |
| GlobalTimePdelayRespEnable   | This parameter allows disabling Pdelay_Resp, Pdelay_Resp_Follow_Up transmission, if no Pdelay_Req messages are expected. FALSE: No Pdelay requests expected. Pdelay_Resp / Pdelay_Resp_Follow_Up transmission is disabled. TRUE: Pdelay requests expected. Pdelay_Resp, Pdelay_Resp_Follow_Up transmission is enabled.  |
| GlobalTimeTxPeriod           | This parameter represents configuration of the TX period.   |
| GlobalTimeFollowUpTimeout    | Timeout value of the Follow_Up message (of the subsequent Sync message). A value of 0 deactivates this timeout observation.   |
| MasterSlaveConflictDetection | Enables master / slave conflict detection and notification. true: detection and notification is enabled. false: detection and notification is disabled.   |
| MessageCompliance            | true: IEEE 802.1AS compliant message format will be used. false: IEEE 802.1AS message format with AUTOSAR extension will be used.   |
| RxCrcValidated               | <ul style="list-style-type: none"> <li>• CRC_IGNORED (ignores any CRC inside the Sub-TLVs)</li> <li>• CRC_NOT_VALIDATED (If MessageCompliance is set to FALSE: Ethernet discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60)</li> <li>• CRC_OPTIONAL ( If MessageCompliance is set to FALSE: Ethernet discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an incorrect CRC value.)</li> <li>• CRC_VALIDATED (If MessageCompliance is set to FALSE: Ethernet discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x44, 0x50 or 0x60, that contain an incorrect CRC value. Ethernet rejects Follow_Up messages with Sub-TLVs of Type 0x34, 0x51 or 0x61)</li> </ul> |
| CrcFlagsRxValidated          | This container collects definitions which parts of the Follow_Up message elements shall be included in the CRC validation.  |
| CrcMessageLength             | messageLength from the Follow_Up Message Header shall be included in CRC calculation.   |
| CrcDomainNumber              | domainNumber from the Follow_Up Message Header shall be included in CRC calculation.  |
| CrcCorrectionField           | correctionField from the Follow_Up Message Header shall be included in CRC calculation.   |

|   |  |
|---|--|
| CrcSourcePortIdentity                       | sourcePortIdentity from the Follow_Up Message Header shall be included in CRC calculation.   |
| CrcSequenceId                               | sequenceId from the Follow_Up Message Header shall be included in CRC calculation.   |
| CrcPreciseOriginTimestamp                   | preciseOriginTimestamp from the Follow_Up Message Field shall be included in CRC calculation.  |
| GlobalTimeUplinkTo-TxSwitchResidenceTime    | This parameter is specifying the default value used for the residence time of the Ethernet Switch [Uplink to Egress]. This value is used by the Ethernet module if the calculation of the residence time failed. |
| TLVFollowUpTimeSubTLV                       | This represents the configuration of whether an AUTOSAR Follow_Up TLV Time Sub-TLV is used or not.   |
| TLVFollowUpStatusSubTLV                     | This represents the configuration of whether an AUTOSAR Follow_Up TLV Status Sub-TLV is used or not.   |
| TLVFollowUpUserDataSubTLV                   | This represents the configuration of whether an AUTOSAR Follow_Up TLV UserData Sub-TLV is used or not.   |
| TSynTLVFollowUpOFSSubTLV                    | This represents the configuration of whether an AUTOSAR Follow_Up TLV OFS Sub-TLV is used or not.  |
| CrcTimeFlagsTxSecured                       | This item collects definitions which parts of the Follow_Up message elements shall be used for CRC calculation.  |
| CRCcorrectionField                          | correctionField from the Follow_Up Message Header shall be included in CRC calculation.  |
| GlobalTimeTxCrcSecured                      | This represents the configuration of whether or not CRC is supported.  |
| GlobalTimeSequenceCounterJump-Width         | GlobalTimeSequenceCounterJumpWidth specifies the maximum allowed jump of the Sequence Counter between consecutive two Sync messages.   |
| GlobalTimePdelayRespAndResp-FollowUpTimeout | Timeout value for Pdelay_Resp and Pdelay_Resp_Follow_Up after a Pdelay_Req has been transmitted resp. a Pdelay_Resp has been received. .   |

**Table 6.1: Configuration Parameters**

## 7 Protocol usage and guidelines

Please note that chapter 5 provides several requirements on usage.

## 8 References

### References

- [1] IEEE Standard 802.1AS-30  
<http://standards.ieee.org/getieee802/download/802.1AS-2011.pdf>
- [2] IEEE 802.1Q-2011 - IEEE Standard for Local and metropolitan area networks - Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks