

| | |
|-----------------------------------|-------------------------------|
| Document Title | Specification of TCP/IP Stack |
| Document Owner | AUTOSAR |
| Document Responsibility | AUTOSAR |
| Document Identification No | 617 |
| Document Classification | Standard |
| Document Status | Final |
| Part of AUTOSAR Release | 4.2.2 |

| Document Change History | | |
|--------------------------------|----------------------------|--|
| Release | Changed by | Change Description |
| 4.2.2 | AUTOSAR Release Management | <ul style="list-style-type: none"> • Support for transmission of fragmented IPv4/IPv6 frames • Clarifications and corrections of requirements • Editorial changes |
| 4.2.1 | AUTOSAR Release Management | <ul style="list-style-type: none"> • Introduction of IPv6 for in-vehicle communication • Support for Switch Control/Configuration, Semi-Static Auto-Configuration • TcpIp generic upper layer support (CDD) • Clarifications and corrections of requirements and sequence charts |
| 4.1.3 | AUTOSAR Release Management | <ul style="list-style-type: none"> • Clarifications and corrections of requirements • Editorial changes |
| 4.1.2 | AUTOSAR Release Management | <ul style="list-style-type: none"> • Added control functions for ARP • Clarifications and corrections of requirements • Editorial changes • Removed chapter(s) on change documentation |
| 4.1.1 | AUTOSAR Administration | <ul style="list-style-type: none"> • Initial Release |

Disclaimer

This specification 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 specification.

The material contained in this specification is protected by copyright and other types of Intellectual Property Rights. The commercial exploitation of the material contained in this specification requires a license to such Intellectual Property Rights.

This specification 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 specification may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The AUTOSAR specifications have been developed for automotive applications only. They have neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Advice for users

AUTOSAR specifications may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the specifications for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such specifications, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

Table of Contents

| | | |
|-------|--|----|
| 1 | Introduction and functional overview | 6 |
| 2 | Acronyms and abbreviations | 7 |
| 3 | Related documentation..... | 8 |
| 3.1 | Input documents | 8 |
| 3.2 | Related standards and norms | 8 |
| 4 | Constraints and assumptions | 11 |
| 4.1 | Limitations | 11 |
| 4.2 | Applicability to car domains | 11 |
| 5 | Dependencies to other modules..... | 12 |
| 5.1 | EthIf..... | 12 |
| 5.2 | EthSM..... | 12 |
| 5.3 | Socket Adaptor | 12 |
| 5.4 | File structure..... | 13 |
| 5.4.1 | Code file structure..... | 13 |
| 5.4.2 | Header file structure..... | 13 |
| 5.5 | Version check..... | 13 |
| 6 | Requirements traceability | 14 |
| 7 | Functional specification | 20 |
| 7.1 | System Scalability | 20 |
| 7.1.1 | Background & Rationale | 20 |
| 7.1.2 | Requirements..... | 21 |
| 7.2 | Internet Protocol Version 4 | 21 |
| 7.2.1 | Internet Protocol (IPv4)..... | 21 |
| 7.2.2 | Address Resolution Protocol (ARP) | 22 |
| 7.2.3 | Dynamic Configuration of IPv4 Link-Local Addresses (Auto-IP) | 23 |
| 7.2.4 | Internet Control Message Protocol (ICMPv4) | 23 |
| 7.3 | Internet Protocol Version 6 | 23 |
| 7.3.1 | Internet Protocol (IPv6)..... | 24 |
| 7.3.2 | Internet Control Message Protocol (ICMPv6) | 25 |
| 7.3.3 | Neighbor Discovery Protocol (NDP)..... | 25 |
| 7.4 | IP Based Protocols | 25 |
| 7.4.1 | Local Address Table | 25 |
| 7.4.2 | User Datagram Protocol (UDP)..... | 26 |
| 7.4.3 | Transmission Control Protocol (TCP) | 26 |
| 7.4.4 | Dynamic Host Configuration Protocol | 26 |
| 7.5 | Message Reception..... | 28 |
| 7.6 | Message Transmission..... | 29 |
| 7.7 | TCP/IP Stack state handling..... | 32 |
| 7.8 | Error classification | 34 |
| 7.8.1 | Development Errors | 34 |
| 7.8.2 | Production Errors | 34 |
| 7.8.3 | Extended Production Errors | 35 |
| 7.9 | Application notes | 38 |

| | | |
|---------|---|-----|
| 7.10 | Debugging Concept | 38 |
| 7.11 | Version checking..... | 38 |
| 8 | API specification..... | 39 |
| 8.1 | Imported types..... | 39 |
| 8.2 | Type definitions | 39 |
| 8.3 | Function definitions..... | 43 |
| 8.3.1 | General | 43 |
| 8.3.2 | Core Communication Control..... | 44 |
| 8.3.3 | Extended Communication Control and Information..... | 49 |
| 8.3.4 | Transmission..... | 61 |
| 8.4 | Call-back notifications..... | 63 |
| 8.4.1 | Tcplp_RxIndication | 63 |
| 8.5 | Scheduled functions | 64 |
| 8.5.1 | Terms and definitions..... | 64 |
| 8.5.2 | Tcplp_MainFunction | 64 |
| 8.6 | Expected Interfaces..... | 64 |
| 8.6.1 | Mandatory Interfaces | 64 |
| 8.6.2 | Optional Interfaces..... | 65 |
| 8.6.3 | Configurable interfaces | 65 |
| 9 | Sequence diagrams | 71 |
| 9.1 | TCP Connection Setup – Client..... | 72 |
| 9.2 | TCP Connection Setup – Server | 73 |
| 9.3 | Reception | 74 |
| 9.4 | Transmission TCP | 75 |
| 9.5 | Transmission UDP..... | 77 |
| 10 | Configuration specification..... | 78 |
| 10.1 | How to read this chapter | 78 |
| 10.2 | Containers and configuration parameters | 79 |
| 10.2.1 | Variants | 79 |
| 10.2.2 | Tcplp..... | 79 |
| 10.2.3 | TcplpGeneral..... | 81 |
| 10.2.4 | TcplpV4General | 84 |
| 10.2.5 | TcplpV6General | 85 |
| 10.2.6 | TcplpConfig | 89 |
| 10.2.7 | TcplpCtrl..... | 90 |
| 10.2.8 | TcplpCtrlDemEventParameterRefs | 91 |
| 10.2.9 | TcplpVXCtrl | 93 |
| 10.2.10 | TcplpV4Ctrl..... | 94 |
| 10.2.11 | TcplpV6Ctrl..... | 96 |
| 10.2.12 | TcplpV6MtuConfig..... | 97 |
| 10.2.13 | TcplpDhcpServerConfig | 99 |
| 10.2.14 | TcplpDhcpAddressAssignment | 101 |
| 10.2.15 | TcplpConfig | 102 |
| 10.2.16 | TcplpV4Config..... | 102 |
| 10.2.17 | TcplpArpConfig..... | 103 |
| 10.2.18 | TcplpAutoIpConfig..... | 105 |
| 10.2.19 | TcplpDhcpConfig..... | 106 |
| 10.2.20 | TcplpIcmpConfig | 106 |

| | | |
|---------|---|-----|
| 10.2.21 | TcplplcmpMsgHandler..... | 107 |
| 10.2.22 | TcplplpFragmentationConfig | 108 |
| 10.2.23 | TcplplpV6Config..... | 110 |
| 10.2.24 | TcplpDhcpV6Config..... | 111 |
| 10.2.25 | TcplplcmpV6Config | 114 |
| 10.2.26 | TcplplcmpV6MsgHandler | 116 |
| 10.2.27 | TcplpNdpConfig..... | 118 |
| 10.2.28 | TcplpNdpArNudConfig..... | 118 |
| 10.2.29 | TcplpNdpPrefixRouterDiscoveryConfig | 121 |
| 10.2.30 | TcplpNdpSlaacConfig..... | 124 |
| 10.2.31 | TcplpLocalAddr | 128 |
| 10.2.32 | TcplpAddrAssignment | 129 |
| 10.2.33 | TcplpStaticIpAddressConfig | 130 |
| 10.2.34 | TcplpNvmBlock | 132 |
| 10.2.35 | TcplpPhysAddrConfig..... | 133 |
| 10.2.36 | TcplpPhysAddrChgHandler | 133 |
| 10.2.37 | TcplpSocketOwnerConfig..... | 136 |
| 10.2.38 | TcplpSocketOwner | 136 |
| 10.2.39 | TcplpTcpConfig | 140 |
| 10.2.40 | TcplpUdpConfig..... | 145 |
| 10.3 | Published Information..... | 146 |

1 Introduction and functional overview

The AUTOSAR TCP/IP module offers functionality to send and receive Internet Protocol data.

The TCP/IP Stack (TCPIP) is located between the Socket Adaptor (SoAd) and the Ethernet Interface (EthIf) modules.

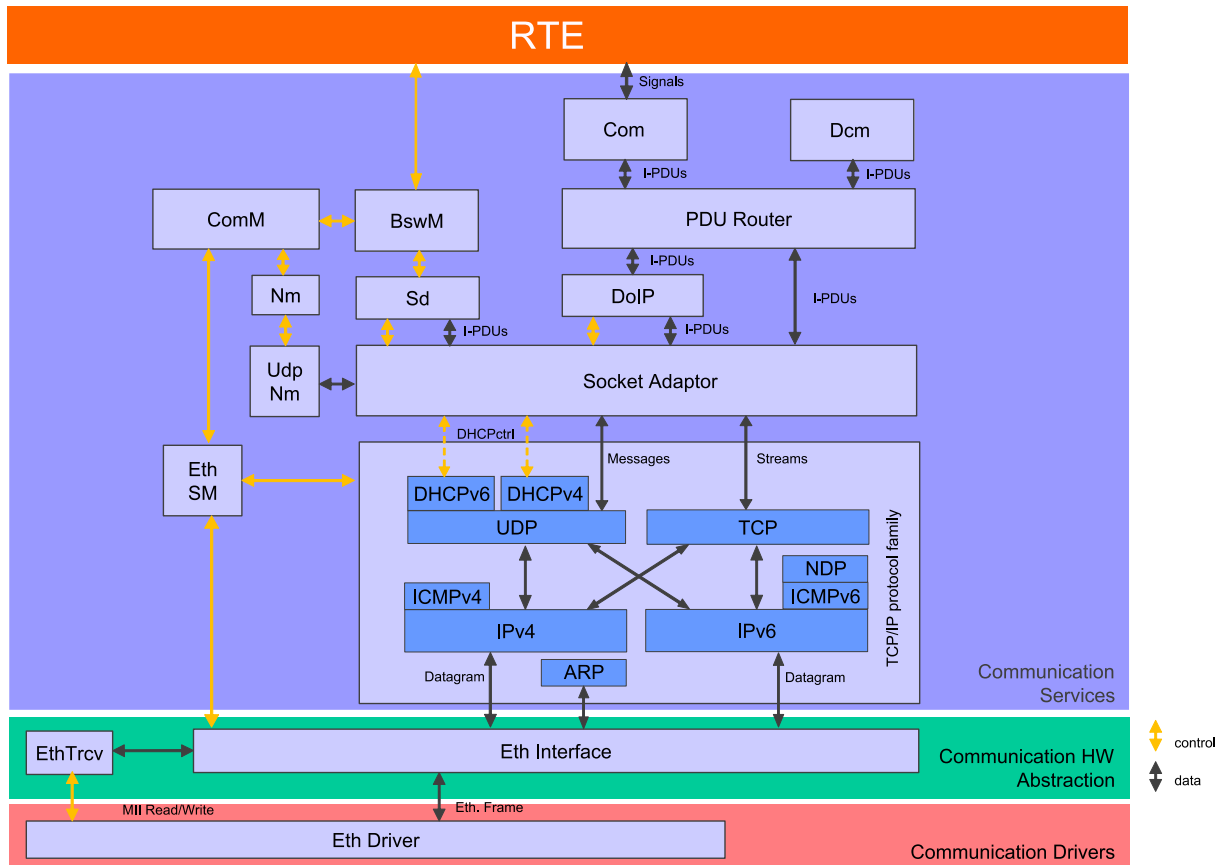


Figure 1: Extended AUTOSAR Communication Stack.

2 Acronyms and abbreviations

| Abbreviation / Acronym: | Description: |
|--------------------------------|---|
| ARP | Address Resolution Protocol |
| DAD | Duplicate Address Detection |
| DEM | Diagnostic Event Manager |
| DET | Development Error Tracer |
| DHCP | Dynamic Host Configuration Protocol |
| DHCPv4 | Dynamic Host Configuration Protocol for Internet Protocol Version 4 |
| DHCPv6 | Dynamic Host Configuration Protocol for Internet Protocol Version 6 |
| ECU | Electronic Control Unit |
| EthIf | Ethernet Interface |
| EthSM | Ethernet State Manager |
| HTTP | HyperText Transfer Protocol |
| IANA | Internet Assigned Numbers Authority |
| ICMP | Internet Control Message Protocol |
| ICMPv4 | Internet Control Message Protocol for Internet Protocol Version 4 |
| ICMPv6 | Internet Control Message Protocol for Internet Protocol Version 6 |
| IETF | Internet Engineering Task Force |
| IP | Internet Protocol |
| IPv4 | Internet Protocol version 4 |
| IPv6 | Internet Protocol version 6 |
| MTU | Maximum Transmission Unit |
| NDP | Neighbor Discovery Protocol |
| SoAd | Socket Adaptor |
| TCP | Transmission Control Protocol |
| TCP/IP | A family of communication protocols used in computer networks |
| TP | Transport Protocol |
| UDP | User Datagram Protocol |

3 Related documentation

3.1 Input documents

[1] AUTOSAR Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf

[2] AUTOSAR Basis Software Mode Manager
AUTOSAR_SWS_BSWModeManager.pdf

[3] AUTOSAR Socket Adaptor
AUTOSAR_SWS_SocketAdaptor.pdf

[4] AUTOSAR SRS BSW General
AUTOSAR_SRS_BSWGeneral.pdf

[5] AUTOSAR SRS Ethernet
AUTOSAR_SRS_Ethernet.pdf

[6] AUTOSAR General Specification for Basic Software Modules
AUTOSAR_SWS_BSWGeneral.pdf

[7] Specification of ECU Configuration
AUTOSAR_TPS_ECUConfiguration.pdf

[8] List of Basic Software Modules
AUTOSAR_TR_BSWModuleList.pdf

3.2 Related standards and norms

[9] IETF RFC 3927
<http://tools.ietf.org/html/rfc3927>

[10] IETF RFC 1122
<http://tools.ietf.org/html/rfc1122>

[11] IETF RFC 826
<http://tools.ietf.org/html/rfc826>

[12] IETF RFC 894
<http://tools.ietf.org/html/rfc894>

[13] IETF RFC 791
<http://tools.ietf.org/html/rfc791>

- [14] IETF RFC 815
<http://tools.ietf.org/html/rfc815>
- [15] IETF RFC 4632
<http://tools.ietf.org/html/rfc4632>
- [16] IETF RFC 1112
<http://tools.ietf.org/html/rfc1112>
- [17] IETF RFC 792
<http://tools.ietf.org/html/rfc792>
- [18] IETF RFC 1191
<http://tools.ietf.org/html/rfc1191>
- [19] IETF RFC 2131
<http://tools.ietf.org/html/rfc2131>
- [20] IETF RFC 768
<http://tools.ietf.org/html/rfc768>
- [21] IETF RFC 793
<http://tools.ietf.org/html/rfc793>
- [22] IETF RFC 813
<http://tools.ietf.org/html/rfc813>
- [23] IETF RFC 896
<http://tools.ietf.org/html/rfc896>
- [24] IETF RFC 5681
<http://tools.ietf.org/html/rfc5681>
- [25] IETF RFC 2460
<http://tools.ietf.org/html/rfc2460>
- [26] IETF RFC [4291](http://tools.ietf.org/html/rfc4291)
<http://tools.ietf.org/html/rfc4291>
- [27] IETF RFC [2464](http://tools.ietf.org/html/rfc2464)
<http://tools.ietf.org/html/rfc2464>
- [28] IETF RFC [6724](http://tools.ietf.org/html/rfc6724)
<http://tools.ietf.org/html/rfc6724>
- [29] IETF RFC [5722](http://tools.ietf.org/html/rfc5722)
<http://tools.ietf.org/html/rfc5722>
- [30] IETF RFC [5095](http://tools.ietf.org/html/rfc5095)
<http://tools.ietf.org/html/rfc5095>

[31] IETF RFC [4862](http://tools.ietf.org/html/rfc4862)
<http://tools.ietf.org/html/rfc4862>

[32] IETF RFC [1981](http://tools.ietf.org/html/rfc1981)
<http://tools.ietf.org/html/rfc1981>

[33] IETF RFC [4429](http://tools.ietf.org/html/rfc4429)
<http://tools.ietf.org/html/rfc4429>

[34] IETF RFC [4443](http://tools.ietf.org/html/rfc4443)
<http://tools.ietf.org/html/rfc4443>

[35] IETF RFC [4861](http://tools.ietf.org/html/rfc4861)
<http://tools.ietf.org/html/rfc4861>

[36] IETF RFC [3315](http://tools.ietf.org/html/rfc3315)
<http://tools.ietf.org/html/rfc3315>

[37] IETF RFC [4702](http://tools.ietf.org/html/rfc4702)
<http://tools.ietf.org/html/rfc4702>

[38] IETF RFC [4704](http://tools.ietf.org/html/rfc4704)
<http://tools.ietf.org/html/rfc4704>

[39] IETF RFC [6582](http://tools.ietf.org/html/rfc6582)
<http://tools.ietf.org/html/rfc6582>

[40] IETF RFC [2132](http://tools.ietf.org/html/rfc2132)
<http://tools.ietf.org/html/rfc2132>

4 Constraints and assumptions

4.1 Limitations

This document does not cover the assignment of UDP or TCP port numbers. There is no reserved space within the IANA assigned number range. Each implementer is responsible for managing the used port numbers.

This document does not cover the management of IP addresses. This might be done dynamically, e.g. by using DHCP, or statically. It is the implementer's responsibility to prevent address conflicts and achieve compliance with IANA address assignments.

This specification does not prescribe a certain physical layer or data rate.

Although a CDD interface is specified, allowing additional upper layer modules, a fan-out of one socket to multiple upper layer modules is not intended to be supported.

4.2 Applicability to car domains

No restrictions.

5 Dependencies to other modules

5.1 EthIf

The Ethernet Interface is the lower layer module of the Tcplp module.

5.2 EthSM

The Ethernet State Manager controls the communication mode of the Tcplp module by requesting communication modes from the Tcplp module. Tcplp notifies the EthSM about communication mode changes.

5.3 Socket Adaptor

The Socket Adaptor is the upper layer module of the Tcplp module.

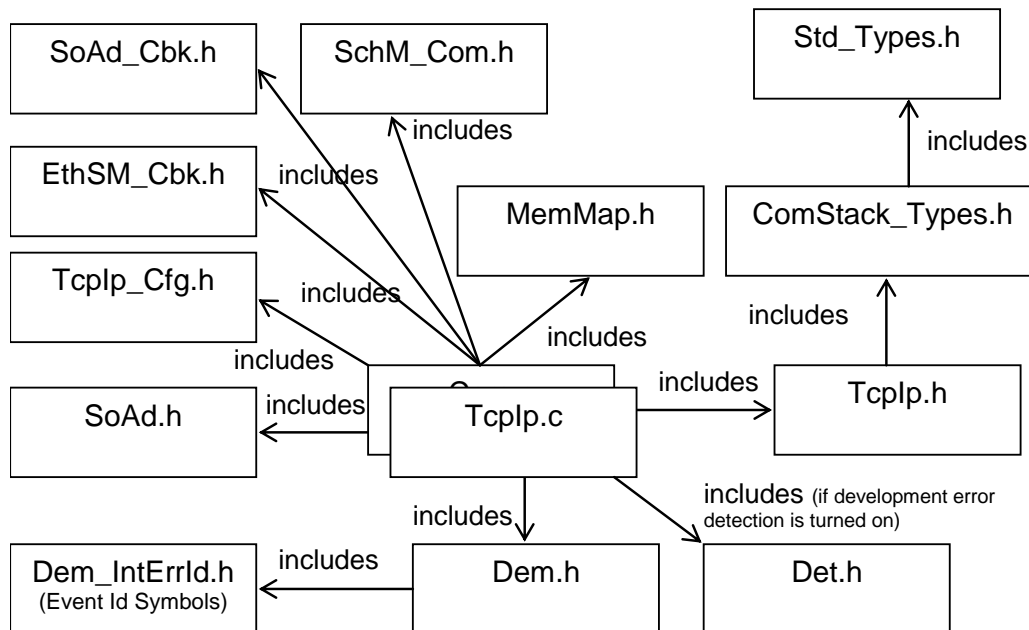
5.4 File structure

5.4.1 Code file structure

For details refer to the chapter 5.1.6 “Code file structure” in *SWS_BSWGeneral*.

5.4.2 Header file structure

This chapter shall contain the h –files especially the h-files which are necessary for configuration. The configuration c-file shall have a naming convention Tcplp_Cfg.h.



[SWS_TCPIP_00068][[The module shall include the Dem.h file.] ()

By this inclusion the APIs to report errors as well as the required Event Id symbols are included. This specification defines the name of the Event Id symbols which are provided by XML to the DEM configuration tool. The DEM configuration tool assigns ECU dependent values to the Event Id symbols and publishes the symbols in Dem_IntErrId.h.

5.5 Version check

For details refer to the chapter 5.1.8 “Version Check” in *SWS_BSWGeneral*.

6 Requirements traceability

| Requirement | Description | Satisfied by |
|-------------|-------------|-----------------|
| - | - | SWS_TCPIP_00002 |
| - | - | SWS_TCPIP_00004 |
| - | - | SWS_TCPIP_00005 |
| - | - | SWS_TCPIP_00006 |
| - | - | SWS_TCPIP_00008 |
| - | - | SWS_TCPIP_00009 |
| - | - | SWS_TCPIP_00010 |
| - | - | SWS_TCPIP_00012 |
| - | - | SWS_TCPIP_00013 |
| - | - | SWS_TCPIP_00014 |
| - | - | SWS_TCPIP_00015 |
| - | - | SWS_TCPIP_00016 |
| - | - | SWS_TCPIP_00017 |
| - | - | SWS_TCPIP_00022 |
| - | - | SWS_TCPIP_00023 |
| - | - | SWS_TCPIP_00024 |
| - | - | SWS_TCPIP_00025 |
| - | - | SWS_TCPIP_00026 |
| - | - | SWS_TCPIP_00027 |
| - | - | SWS_TCPIP_00028 |
| - | - | SWS_TCPIP_00029 |
| - | - | SWS_TCPIP_00030 |
| - | - | SWS_TCPIP_00031 |
| - | - | SWS_TCPIP_00032 |
| - | - | SWS_TCPIP_00033 |
| - | - | SWS_TCPIP_00037 |
| - | - | SWS_TCPIP_00038 |
| - | - | SWS_TCPIP_00039 |
| - | - | SWS_TCPIP_00042 |
| - | - | SWS_TCPIP_00043 |
| - | - | SWS_TCPIP_00050 |
| - | - | SWS_TCPIP_00052 |
| - | - | SWS_TCPIP_00053 |
| - | - | SWS_TCPIP_00054 |
| - | - | SWS_TCPIP_00055 |

| | | |
|---|---|-----------------|
| - | - | SWS_TCPIP_00056 |
| - | - | SWS_TCPIP_00057 |
| - | - | SWS_TCPIP_00059 |
| - | - | SWS_TCPIP_00060 |
| - | - | SWS_TCPIP_00061 |
| - | - | SWS_TCPIP_00062 |
| - | - | SWS_TCPIP_00063 |
| - | - | SWS_TCPIP_00064 |
| - | - | SWS_TCPIP_00065 |
| - | - | SWS_TCPIP_00066 |
| - | - | SWS_TCPIP_00067 |
| - | - | SWS_TCPIP_00068 |
| - | - | SWS_TCPIP_00070 |
| - | - | SWS_TCPIP_00071 |
| - | - | SWS_TCPIP_00072 |
| - | - | SWS_TCPIP_00073 |
| - | - | SWS_TCPIP_00074 |
| - | - | SWS_TCPIP_00075 |
| - | - | SWS_TCPIP_00076 |
| - | - | SWS_TCPIP_00077 |
| - | - | SWS_TCPIP_00078 |
| - | - | SWS_TCPIP_00079 |
| - | - | SWS_TCPIP_00080 |
| - | - | SWS_TCPIP_00082 |
| - | - | SWS_TCPIP_00083 |
| - | - | SWS_TCPIP_00084 |
| - | - | SWS_TCPIP_00085 |
| - | - | SWS_TCPIP_00086 |
| - | - | SWS_TCPIP_00087 |
| - | - | SWS_TCPIP_00088 |
| - | - | SWS_TCPIP_00089 |
| - | - | SWS_TCPIP_00090 |
| - | - | SWS_TCPIP_00091 |
| - | - | SWS_TCPIP_00092 |
| - | - | SWS_TCPIP_00093 |
| - | - | SWS_TCPIP_00094 |
| - | - | SWS_TCPIP_00095 |
| - | - | SWS_TCPIP_00096 |
| - | - | SWS_TCPIP_00097 |

| | | |
|---|---|-----------------|
| - | - | SWS_TCPIP_00098 |
| - | - | SWS_TCPIP_00099 |
| - | - | SWS_TCPIP_00100 |
| - | - | SWS_TCPIP_00101 |
| - | - | SWS_TCPIP_00102 |
| - | - | SWS_TCPIP_00103 |
| - | - | SWS_TCPIP_00104 |
| - | - | SWS_TCPIP_00109 |
| - | - | SWS_TCPIP_00110 |
| - | - | SWS_TCPIP_00111 |
| - | - | SWS_TCPIP_00112 |
| - | - | SWS_TCPIP_00113 |
| - | - | SWS_TCPIP_00114 |
| - | - | SWS_TCPIP_00115 |
| - | - | SWS_TCPIP_00116 |
| - | - | SWS_TCPIP_00117 |
| - | - | SWS_TCPIP_00118 |
| - | - | SWS_TCPIP_00119 |
| - | - | SWS_TCPIP_00120 |
| - | - | SWS_TCPIP_00121 |
| - | - | SWS_TCPIP_00122 |
| - | - | SWS_TCPIP_00123 |
| - | - | SWS_TCPIP_00124 |
| - | - | SWS_TCPIP_00125 |
| - | - | SWS_TCPIP_00126 |
| - | - | SWS_TCPIP_00127 |
| - | - | SWS_TCPIP_00128 |
| - | - | SWS_TCPIP_00129 |
| - | - | SWS_TCPIP_00130 |
| - | - | SWS_TCPIP_00131 |
| - | - | SWS_TCPIP_00132 |
| - | - | SWS_TCPIP_00133 |
| - | - | SWS_TCPIP_00134 |
| - | - | SWS_TCPIP_00135 |
| - | - | SWS_TCPIP_00136 |
| - | - | SWS_TCPIP_00137 |
| - | - | SWS_TCPIP_00138 |
| - | - | SWS_TCPIP_00139 |
| - | - | SWS_TCPIP_00140 |

| | | |
|---|---|-----------------|
| - | - | SWS_TCPIP_00141 |
| - | - | SWS_TCPIP_00142 |
| - | - | SWS_TCPIP_00143 |
| - | - | SWS_TCPIP_00144 |
| - | - | SWS_TCPIP_00146 |
| - | - | SWS_TCPIP_00147 |
| - | - | SWS_TCPIP_00148 |
| - | - | SWS_TCPIP_00149 |
| - | - | SWS_TCPIP_00150 |
| - | - | SWS_TCPIP_00151 |
| - | - | SWS_TCPIP_00152 |
| - | - | SWS_TCPIP_00153 |
| - | - | SWS_TCPIP_00154 |
| - | - | SWS_TCPIP_00155 |
| - | - | SWS_TCPIP_00156 |
| - | - | SWS_TCPIP_00157 |
| - | - | SWS_TCPIP_00158 |
| - | - | SWS_TCPIP_00159 |
| - | - | SWS_TCPIP_00160 |
| - | - | SWS_TCPIP_00161 |
| - | - | SWS_TCPIP_00162 |
| - | - | SWS_TCPIP_00163 |
| - | - | SWS_TCPIP_00164 |
| - | - | SWS_TCPIP_00165 |
| - | - | SWS_TCPIP_00166 |
| - | - | SWS_TCPIP_00167 |
| - | - | SWS_TCPIP_00168 |
| - | - | SWS_TCPIP_00169 |
| - | - | SWS_TCPIP_00170 |
| - | - | SWS_TCPIP_00171 |
| - | - | SWS_TCPIP_00172 |
| - | - | SWS_TCPIP_00173 |
| - | - | SWS_TCPIP_00174 |
| - | - | SWS_TCPIP_00175 |
| - | - | SWS_TCPIP_00176 |
| - | - | SWS_TCPIP_00177 |
| - | - | SWS_TCPIP_00178 |
| - | - | SWS_TCPIP_00179 |
| - | - | SWS_TCPIP_00180 |

| | | |
|---------------|---|--|
| - | - | SWS_TCPIP_00181 |
| - | - | SWS_TCPIP_00182 |
| - | - | SWS_TCPIP_00183 |
| - | - | SWS_TCPIP_00184 |
| - | - | SWS_TCPIP_00185 |
| - | - | SWS_TCPIP_00187 |
| - | - | SWS_TCPIP_00191 |
| - | - | SWS_TCPIP_00192 |
| - | - | SWS_TCPIP_00193 |
| - | - | SWS_TCPIP_00194 |
| - | - | SWS_TCPIP_00195 |
| - | - | SWS_TCPIP_00196 |
| - | - | SWS_TCPIP_00197 |
| - | - | SWS_TCPIP_00198 |
| - | - | SWS_TCPIP_00199 |
| - | - | SWS_TCPIP_00202 |
| - | - | SWS_TCPIP_00203 |
| - | - | SWS_TCPIP_00204 |
| - | - | SWS_TCPIP_00205 |
| - | - | SWS_TCPIP_00206 |
| - | - | SWS_TCPIP_00207 |
| - | - | SWS_TCPIP_00208 |
| - | - | SWS_TCPIP_00209 |
| - | - | SWS_TCPIP_00210 |
| - | - | SWS_TCPIP_00211 |
| - | - | SWS_TCPIP_00212 |
| - | - | SWS_TCPIP_00213 |
| - | - | SWS_TCPIP_00214 |
| - | - | SWS_TCPIP_00215 |
| - | - | SWS_TCPIP_00230 |
| - | - | SWS_TCPIP_00231 |
| - | - | SWS_TCPIP_00232 |
| SRS_Eth_00065 | An API shall be available to fill DHCP field | SWS_TCPIP_00020, SWS_TCPIP_00190, SWS_TCPIP_00243, SWS_TCPIP_00244, SWS_TCPIP_00245, SWS_TCPIP_00246, SWS_TCPIP_00247, SWS_TCPIP_00248, SWS_TCPIP_00249, SWS_TCPIP_00250, SWS_TCPIP_00251, SWS_TCPIP_00252 |
| SRS_Eth_00066 | An API shall be available to read any received DHCP field | SWS_TCPIP_00040, SWS_TCPIP_00189, SWS_TCPIP_00233, SWS_TCPIP_00234, SWS_TCPIP_00235, SWS_TCPIP_00236, SWS_TCPIP_00237, SWS_TCPIP_00238, SWS_TCPIP_00239, SWS_TCPIP_00240, |

| | | |
|---------------|--|---|
| | | SWS_TCPIP_00241, SWS_TCPIP_00242 |
| SRS_Eth_00087 | Semi-Static Auto-Configuration | SWS_TCPIP_00058, SWS_TCPIP_00201, SWS_TCPIP_00216, SWS_TCPIP_00217, SWS_TCPIP_00218, SWS_TCPIP_00219 |
| SRS_Eth_00088 | DHCP Server | SWS_TCPIP_00058, SWS_TCPIP_00200 |
| SRS_Eth_00103 | TcpIp shall support generic upper layers | SWS_TCPIP_00018, SWS_TCPIP_00220, SWS_TCPIP_00221, SWS_TCPIP_00222, SWS_TCPIP_00223, SWS_TCPIP_00224, SWS_TCPIP_00225, SWS_TCPIP_00226, SWS_TCPIP_00227, SWS_TCPIP_00228, SWS_TCPIP_00229 |

7 Functional specification

Figure 2 provides an architecture overview of the AUTOSAR TCP/IP stack. The TCP/IP stack consists of the sub modules within the red box. Furthermore the interaction with other AUTOSAR modules (beside Dem and Det) is shown.

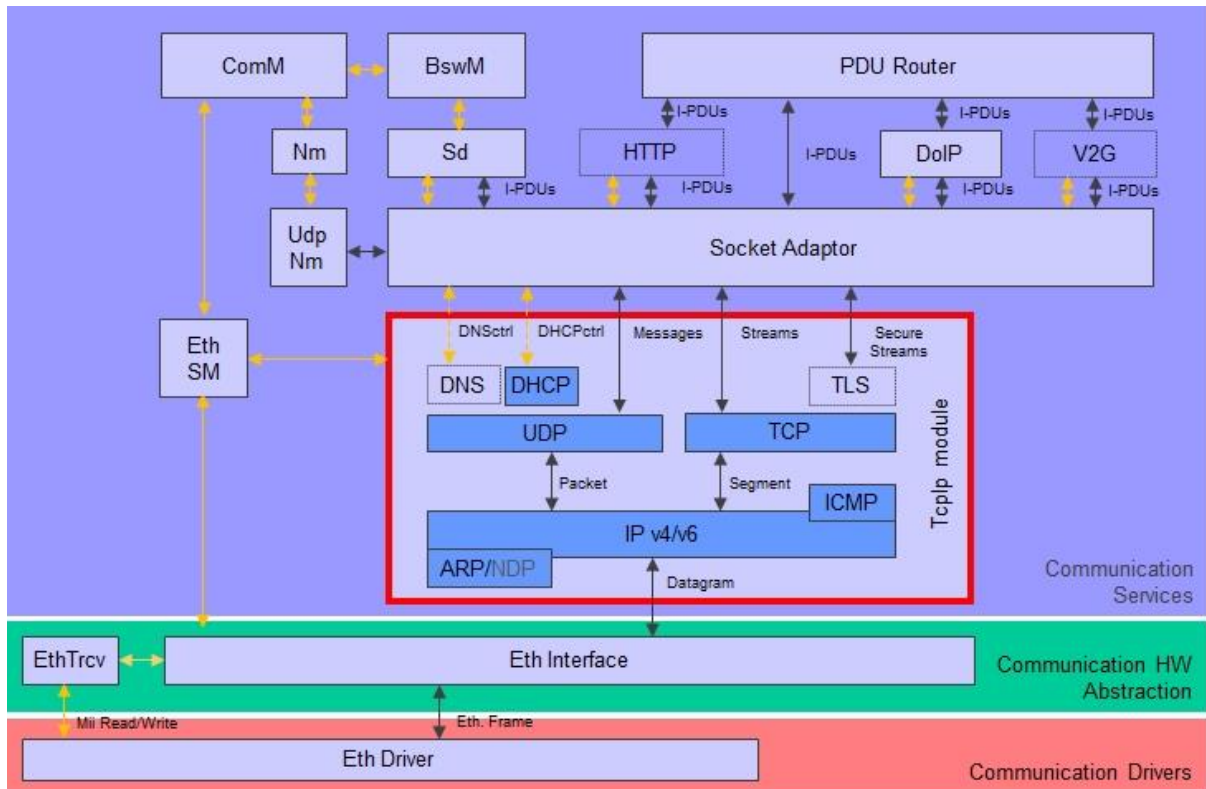


Figure 2: TCP/IP Architecture Overview

[SWS_TCPIP_00052] The TCP/IP stack shall consist of sub modules implementing specific functionalities defined in the subchapters below.] ()

7.1 System Scalability

7.1.1 Background & Rationale

The Tcplp module supports a variety of different use case, not all of them are required by each user. In order to achieve a scalable Tcplp Stack the features shall be grouped according to the following scalability classes:

Scalability Class 1: IPv4 – In-Vehicle and Diagnostic Communication

Scalability Class 2: IPv6 – In-Vehicle and Diagnostic Communication

Scalability Class 3: IPv4 and IPv6 (Dual Stack) – In-Vehicle and Diagnostic Communication

Further functionality regarding for example global communication, security or mobility use cases are not addressed by the currently specified scalability classes and will be considered in future release by additional scalability classes.

The following features shall be available in the respective Scalability Class:

| Feature | Scalability Class 1 | Scalability Class 2 | Scalability Class 3 |
|---------|---------------------|---------------------|---------------------|
| IPv4 | ✓ | | ✓ |
| ARP | ✓ | | ✓ |
| ICMPv4 | ✓ | | ✓ |
| DHCPv4 | ✓ | | ✓ |
| Auto-IP | ✓ | | ✓ |
| UDP | ✓ | ✓ | ✓ |
| TCP | ✓ | ✓ | ✓ |
| IPv6 | | ✓ | ✓ |
| NDP | | ✓ | ✓ |
| ICMPv6 | | ✓ | ✓ |
| DHCPv6 | | ✓ | ✓ |

Figure 3: Tcplp Scalability Classes

7.1.2 Requirements

[SWS_TCPIP_00148] The Tcplp module for IPv4 – In-Vehicle and Diagnostic Communication (Scalability class 1) shall support the features listed in Figure 3: Tcplp Scalability Classes, column Scalability Class 1.] ()

[SWS_TCPIP_00149] The Tcplp module for IPv6 – In-Vehicle and Diagnostic Communication (Scalability class 2) shall support the features listed in Figure 3: Tcplp Scalability Classes, column Scalability Class 2.] ()

[SWS_TCPIP_00150] The Tcplp module for IPv4 and IPv6 (Dual Stack) – In-Vehicle and Diagnostic Communication (Scalability class 3) shall support the features listed in Figure 3: Tcplp Scalability Classes, column Scalability Class 3.] ()

7.2 Internet Protocol Version 4

7.2.1 Internet Protocol (IPv4)

The Internet Protocol (IP) is the main protocol of the TCP/IP stack and is responsible for delivering datagrams from a source host identified by the source address to one or multiple destination hosts identified by the destination address. IP hides the

underlying physical network interface, is an unreliable, best-effort, and connectionless packet delivery protocol.

[SWS_TCPIP_00053] The Tcplp shall implement the Internet Protocol as defined in IETF RFC 791 (Internet Protocol of version 4).] ()

[SWS_TCPIP_00095] The Tcplp shall encapsulate IP packets in Ethernet frames according to IETF RFC 894.] ()

[SWS_TCPIP_00096] The Tcplp shall support the identification of the network an IP address belongs to, by using a network mask (prefix) in addition to the IP address according to IETF RFC 4632, section 3.1.] ()

[SWS_TCPIP_00102] The Tcplp shall fulfill the Internet Protocol related requirements specified by IETF RFC 1122, section 3.2.1.1 (Version number), 3.2.1.2 (Checksum), 3.2.1.3 (Addressing), 3.2.1.7 (TTL), and 3.3.2 (Reassembly).] ()

[SWS_TCPIP_00097] The Tcplp shall be able to transmit IP datagrams to a group of hosts identified by a single IP destination address (multicast address) according to IETF RFC 1112, section 4, 6.2, and 6.4.] ()

[SWS_TCPIP_00098] The Tcplp shall be able to receive multicast IP datagrams identified by a single IP destination address (multicast address) according to IETF RFC 1112, section 4 and 7.2 (excluding the requirement for IGMP).] ()

[SWS_TCPIP_00054] The Tcplp shall be able to reassemble incoming datagrams that are fragmented according to IETF RFC 815 (IP Datagram Reassembly Algorithms).] ()

[SWS_TCPIP_00231] The Tcplp shall fragment oversized IPv4 frames before transmission according to the description in IETF 791 Section Fragmentation and Reassembly.] ()

[SWS_TCPIP_00055] The Tcplp shall discover the maximum transmission unit (MTU) for a path as defined in IETF RFC 1191 (Path MTU Discovery).] ()

7.2.2 Address Resolution Protocol (ARP)

[SWS_TCPIP_00056] The Tcplp shall implement the Address Resolution Protocol (ARP) as defined in IETF RFC 826.] ()

[SWS_TCPIP_00090] The Tcplp shall limit the number of ARP table (address resolution cache) entries to the number specified by the configuration parameter TcplpArpTableSizeMax.] ()

[SWS_TCPIP_00091] The Tcplp shall remove entries of the ARP table if they are not used for the timeout specified by the configuration parameter TcplpArpTableEntryTimeout.] ()

[SWS_TCPIP_00092] The Tcplp shall use the information from each received IP packet to update the ARP table in addition to received ARP packets.] ()

[SWS_TCPIP_00142] The Tcplp shall call <Up_PhysAddrTableChg>() directly after each ARP table change:

- (a) If Tcplp adds a new entry or updates an existing one, the parameter valid shall be set to TRUE and the parameters IpAddrPtr and PhysAddrPtr shall be set according to the new or updated entry.
- (b) In case Tcplp removes an entry, valid shall be set to FALSE and the parameters IpAddrPtr and PhysAddrPtr shall be set according to the removed entry.] ()

[SWS_TCPIP_00093] On assignment of a new IP address the Tcplp shall send a configurable number (TcplpArpNumGratuitousARPOnStartup) of gratuitous ARP replies according to IETF RFC 2002, section 4.6, second indent.] ()

[SWS_TCPIP_00151] If a packet shall be transmitted to a remote host and the link layer address does not exist in the ARP table, the Tcplp shall queue this packet according to IETF RFC 1122, section 2.3.2.2 ARP Packet Queue and transmit the packet when the address has been resolved.] ()

7.2.3 Dynamic Configuration of IPv4 Link-Local Addresses (Auto-IP)

[SWS_TCPIP_00057] The Tcplp shall support the dynamic configuration of IPv4 Link Local addresses as defined in IETF RFC 3927 (Dynamic Configuration of IPv4 Link-Local Addresses).] ()

7.2.4 Internet Control Message Protocol (ICMPv4)

[SWS_TCPIP_00059] The Tcplp shall support the transmission and reception of Internet Control Message Protocol (ICMPv4) messages as defined in IETF RFC 792 (Internet Control Message Protocol in version 4).] ()

7.3 Internet Protocol Version 6

[SWS_TCPIP_00153] The Tcplp shall support the frame format for transmission of IPv6 packets and the method of forming IPv6 link-local addresses and statelessly autoconfigured addresses on Ethernet networks as defined in IETF RFC 2464 (Transmission of IPv6 Packets over Ethernet Networks).] ()

[SWS_TCPIP_00154] The Tcplp shall support the source address selection algorithm as defined in IETF RFC 6724 (Default Address Selection for Internet Protocol Version 6 (IPv6)). Only section 5 Source Address Selection shall be supported.] ()

[SWS_TCPIP_00156] The Tcplp shall support the IETF RFC 5095 (Deprecation of Type 0 Routing Headers in IPv6). The functionality provided by IPv6's Type 0 Routing Header can be exploited in order to achieve traffic amplification over a remote path for the purposes of generating denial-of-service traffic. This document updates the IPv6 specification to deprecate the use of IPv6 Type 0 Routing Headers, in light of this security concern.] ()

[SWS_TCPIP_00157] The Tcplp shall support the section 5.1. Node Configuration Variables, section 5.3. Creation of Link-Local Addresses, section 5.4, Duplicate Address Detection, section 5.5 Creation of Global Addresses and section 5.6 Configuration Consistency of the IETF RFC 4862 (IPv6 Stateless Address Autoconfiguration).] ()

[SWS_TCPIP_00158] The Tcplp shall support the Path MTU Discovery for IPv6 as defined in IETF RFC 1981 (Path MTU Discovery for IP version 6). If the max. MTU is used, the Path MTU Discovery shall not try to increase the value.] ()

[SWS_TCPIP_00159] The Tcplp shall support the Duplicate Address Detection as defined in IETF RFC 4429 (Optimistic Duplicate Address Detection (DAD) for IPv6).] ()

7.3.1 Internet Protocol (IPv6)

[SWS_TCPIP_00160] The Tcplp shall support the basic IPv6 header and the initially defined IPv6 extension headers and options as defined in IETF RFC 2460 (Internet Protocol, Version 6 (IPv6) Specification).] ()

[SWS_TCPIP_00161] The Tcplp shall support the reception and reassembly of fragmented IPv6 frames according to IETF 2460 Section 4.5 Fragment Header.] ()

[SWS_TCPIP_00155] The Tcplp shall support the section 4, first paragraph of the IETF RFC 5722 (Handling of Overlapping IPv6 Fragments). The IETF RFC 5722 demonstrates the security issues associated with allowing overlapping fragments and updates the IPv6 specification to explicitly forbid overlapping fragments (transmission and reception).] ()

[SWS_TCPIP_00232] The Tcplp shall fragment oversized IPv6 frames before transmission according to IETF 2460 Section 4.5 Fragment Header.] ()

[SWS_TCPIP_00162] The Tcplp shall support the section 2, IPv6 Addressing of IETF RFC 4291 (IP Version 6 Addressing Architecture). Section 2.6. Anycast

Addresses shall not be supported. Section 2.8 A Node's Required Addresses shall be limited to the node requirements for host only.] ()

7.3.2 Internet Control Message Protocol (ICMPv6)

[SWS_TCPIP_00163] The Tcplp shall support the Internet Control Message Protocol Version 6 as defined in IETF RFC 4443 (Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification).] ()

7.3.3 Neighbor Discovery Protocol (NDP)

[SWS_TCPIP_00164] The Tcplp shall support the Neighbor Discovery protocol for IP Version 6 as defined in IETF RFC 4861 (Neighbor Discovery for IP version 6 (IPv6)). Section 4.5 Redirect Message Format, section 7.2.8.Proxy Neighbor Advertisements and section 8. Redirect Function shall not be supported. No router functionality shall be supported.] ()

[SWS_TCPIP_00165] If a packet shall be transmitted to a remote host and the link layer address does not exist in the Neighbor Cache, the Tcplp shall queue this packet according to IETF RFC 4861, section 7.2.2. Sending Neighbor Solicitations, 5th paragraph and transmit the packet when the address has been resolved.] ()

7.4 IP Based Protocols

7.4.1 Local Address Table

[SWS_TCPIP_00099] The Tcplp shall maintain a table of local IP addresses, which can be assigned to an EthIf controller during runtime according to the configuration container TcplpLocalAddr (including its subcontainers).] ()

Note: Each entry of the local IP address table is uniquely identified by the configuration parameter TcplpAddrId.

[SWS_TCPIP_00100] In case no TcplpStaticAddressConfig is provided, the Tcplp shall enable to specify a multicast IP address during runtime via Tcplp_RequestIpAddrAssignment().] ()

[SWS_TCPIP_00130] The Local IP address used for a socket is specified via Tcplp_Bind().] ()

[SWS_TCPIP_00219] If a TcplpAddrAssignment configured with TCPIP_STORE is started, Tcplp shall check the NvMBlock (see **ECUC_Tcplp_00184** :) for a valid IP address. If a valid address is present, Tcplp shall assign this address as if it was a static address. If no valid address is present, Tcplp shall start the respective IP

address assignment method related to the TcplpAddrAssignment. Once the procedure is complete, Tcplp shall store the new address in the NvMBlock.] (SRS_Eth_00087)

7.4.2 User Datagram Protocol (UDP)

[SWS_TCPIP_00060] The Tcplp shall implement the User Datagram Protocol (UDP) as defined in IETF RFC 768 (User Datagram Protocol).] ()

[SWS_TCPIP_00103] The Tcplp shall fulfill the UDP related requirements specified by IETF RFC 1122, section 4.1.3.1 (Ports), 4.1.3.4 (UDP Checksums), and 4.1.3.6 (Invalid Addresses).] ()

7.4.3 Transmission Control Protocol (TCP)

[SWS_TCPIP_00061] The Tcplp shall implement the Transmission Control Protocol (TCP) as defined in IETF RFC 793 (Transmission Control Protocol).] ()

[SWS_TCPIP_00104] The Tcplp shall fulfill the TCP related requirements specified by IETF RFC 1122, section 4.2.2.3 (Window Size), 4.2.2.5 (TCP Options), 4.2.2.6 (MSS), 4.2.2.7 (Checksum), 4.2.2.9 (Initial sequence number selection), 4.2.2.10 (Simultaneous Open Attempts), 4.2.2.11 (Recovery from Old Duplicate SYN), 4.2.2.13 (Closing a Connection, excluding “half-duplex close”), 4.2.2.15 (Retransmission Timeout), 4.2.2.16 (Managing the Window), 4.2.2.17 (Probing Zero Windows), 4.2.2.18 (Passive OPEN Calls), 4.2.2.19 (TTL), 4.2.3.2 (delayed ACK), 4.2.3.6 (TCP Keep Alive), and 4.2.3.10 (Remote Address Validation).] ()

[SWS_TCPIP_00062] The Tcplp shall support the Window and Acknowledgment Strategy in TCP as defined in IETF RFC 813.] ()

[SWS_TCPIP_00063] The Tcplp shall implement the Nagle Algorithm as defined in IETF RFC 896 (Congestion Control in IP/TCP Internetworks).] ()

[SWS_TCPIP_00064] The Tcplp shall implement the congestion control strategies slow-start, congestion avoidance, fast retransmit and fast recovery as defined in IETF RFC 5681.] ()

[SWS_TCPIP_00168] The Tcplp shall support the specific algorithm for responding to partial acknowledgments as defined in IETF RFC 6582 (The NewReno Modification to TCP's Fast Recovery Algorithm). The modification shall only be used if the Fast Recovery strategy of IETF RFC 5681 is enabled.] ()

7.4.4 Dynamic Host Configuration Protocol

[SWS_TCPIP_00200] The server part of the Dynamic Host Configuration Protocol shall be pre compile time configurable ON/OFF by the configuration parameter TcplpDhcpServerEnabled (see **ECUC_Tcplp_00183** :) (SRS_Eth_00088)

[SWS_TCPIP_00201] The server part of the Dynamic Host Configuration Protocol shall respond to client requests by assigning an available IP address according to the DHCP server configuration for the related TcplpCtrl.] (SRS_Eth_00087)

[SWS_TCPIP_00218] If the configuration contains TcplpDhcpAddressAssignments that refer to specific ports of an Ethernet Switch, DHCP server shall identify the port the request was received from, by calling EthIf_GetPortMacAddr() with the MAC address of the DHCP client and choose an available IP address of the TcplpDhcpAddressAssignment related to the same port.] (SRS_Eth_00087)

7.4.4.1 Dynamic Host Configuration Protocol (DHCPv4)

[SWS_TCPIP_00058] [The Tcplp shall implement the client and the server part of the Dynamic Host Configuration Protocol (DHCPv4) for the dynamic configuration of IPv4 addresses as defined in IETF RFC 2131 (Dynamic Host Configuration Protocol).] (SRS_Eth_00087, SRS_Eth_00088)

[SWS_TCPIP_00152] The Tcplp shall support the Fully Qualified Domain Name Option for Dynamic Host Configuration Protocol for IPv4 Client requirements as defined in IETF RFC 4702 (The Dynamic Host Configuration Protocol for IPv4 (DHCPv4) Client Fully Qualified Domain Name (FQDN) Option). No DNS shall be supported. Only section 2 The Client FQDN Option and section 3 DHCP Client Behavior shall be supported. Sub-Section 3.2, 3.3, 3.5 shall not be supported.] ()

7.4.4.2 Dynamic Host Configuration Protocol (DHCPv6)

[SWS_TCPIP_00166] The Tcplp shall support the client part of the Dynamic Host Configuration Protocol for IPv6 (DHCPv6) which enables DHCP servers to pass configuration parameters such as IPv6 network addresses to IPv6 nodes as defined in IETF RFC 3315 (Dynamic Host Configuration Protocol for IPv6 (DHCPv6)). Due to the fact that only the client functionality shall be supported, the following sections shall not be supported:

- Relay Agent Behavior
- Server Behavior
- Section 12. Management of Temporary Addresses
- Section 21. Authentication of DHCP Messages
- Section 22.5. Identity Association for Temporary Addresses Option
- Section 22.11. Authentication Option
- Section 22.14. Rapid Commit Option

] ()

[SWS_TCPIP_00167] The Tcplp shall support the Fully Qualified Domain Name Option for Dynamic Host Configuration Protocol for IPv6 Client requirements as defined in IETF RFC 4704 (The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option). No DNS shall be supported. Only section 4 DHCPv6 Client FQDN Option and section 5 DHCPv6

Client Behavior shall be supported. Sub-Section 5.1, 5.2, 5.4 shall not be supported.
] ()

7.5 Message Reception

[SWS_TCPIP_00169] The TcpIp IP-layer shall map received IP datagrams to an entry in the local address table (TcpIpAddrId).

The local address table mapping is successful if ALL of the following conditions are fulfilled:

- a) The receiving interface matches the interface assigned to the local address table entry (EthIfCtrl).
- b) The destination IP address contained in the IP header matches the currently assigned IP address of the local address table entry.

All IP datagrams which cannot be mapped to an entry in the local address table shall be silently discarded.

All successfully mapped IP datagrams shall be forwarded to the upper layer protocol.

] ()

[SWS_TCPIP_00170] The TcpIp UDP-layer shall map received UDP datagrams to sockets based on the destination port as contained in the UDP protocol header and the local address (TcpIpAddrId). The local address (TcpIpAddrId) matches if ANY of the following conditions is fulfilled:

- a) The socket is bound to the local address (TcpIpAddrId)
- b) The socket local address uses the wildcard "ANY" AND the socket EthIfCtrl is identical to the EthIfCtrl used in the local address (TcpIpAddrId)
- c) The socket is bound to TCPIP_LOCALADDRID_ANY

The socket is bound to a local address and the EthIfCtrl is identical to the EthIfCtrl used in the local address (TcpIpAddrId) and the received local address (TcpIpAddrId) is a broadcast address.] ()

[SWS_TCPIP_00171] For received UDP datagrams where the local address (TcpIpAddrId) is a broadcast or multicast address, all matching sockets shall receive the incoming message.] ()

Note: A socket may either be explicitly bound to a local IP address by using TcpIp_Bind() or implicitly as part of TcpIp_UdpTransmit() (if it is called without a previous call of TcpIp_Bind()).

[SWS_TCPIP_00172] The TcpIp TCP-layer shall map received TCP datagrams to sockets based on the destination port as contained in the TCP protocol header and the local address (TcpIpAddrId). The local address (TcpIpAddrId) matches if ANY of the following conditions is fulfilled:

- a) The socket is bound to a unicast local address (TcplpAddrId)
- b) The socket local address uses the wildcard “ANY” AND the socket EthIfCtrl is identical to the EthIfCtrl used in the local address (TcplpAddrId)
- c) The socket is bound to TCPIP_LOCALADDRID_ANY

] ()

[SWS_TCPIP_00173] Sockets with established TCP connections shall match source port, source IP address, destination port and destination IP address as contained in the protocol headers additionally to the generic TCP mapping criteria described in [SWS_TCPIP_00172].] ()

[SWS_TCPIP_00174] Received TCP datagrams where the local address (TcplpAddrId) is a broadcast or multicast address, shall be silently discarded.] ()

[SWS_TCPIP_00203] For receptions the Tcplp Module shall ignore the protocol checksum fields of frames with respect to the configuration of the Ethernet Controller according to the following list:

- a) for IPv4 frames if IPv4 checksum verification in hardware is enabled, i.e. EthCtrlEnableOffloadChecksumIPv4 is set to TRUE
- b) for ICMP frames if ICMP checksum verification in hardware is enabled, i.e. EthCtrlEnableOffloadChecksumICMP is set to TRUE
- c) for TCP frames if TCP checksum verification in hardware is enabled, i.e. EthCtrlEnableOffloadChecksumTCP is set to TRUE
- d) for UDP frames if UDP checksum verification in hardware is enabled, i.e. EthCtrlEnableOffloadChecksumUDP is set to TRUE.

In all other cases, the Tcplp module shall treat frames with mismatching checksums according the related protocol specification.] ()

7.6 Message Transmission

[SWS_TCPIP_00175] If data is transmitted using a socket which is bound to an IPv4 Unicast local address (TcplpAddrId) the Tcplp shall use the IP address assigned to the local address (TcplpAddrId) as source IP address in the IP datagram header. The IP datagram shall be transmitted using the EthIfCtrl the local address (TcplpAddrId) is mapped to.] ()

[SWS_TCPIP_00176] If data is transmitted using an IPv4 socket which is bound to a local address (TcplpAddrId) using the wildcard “ANY”, then the Tcplp shall use the IP address of the configured local address (TcplpAddrId), which is of type IPv4 Unicast and assigned to the same EthIfCtrl, as the bound local address (TcplpAddrId) as source IP address in the IP datagram header.] ()

[SWS_TCPIP_00177] If data is transmitted using an IPv4 socket which is bound to TCPIP_LOCALADDRID_ANY, then the Tcplp shall use the IP address of the configured local address (TcplpAddrId), which is of type IPv4 Unicast and assigned

to the EthIfCtrl in the same subnet as the destination IPv4 address as source IP address in the IP datagram header. If no matching subnet is found the IPv4 Unicast local address (TcplpAddrId) of EthIfCtrl = 0 is selected.] ()

[SWS_TCPIP_00178] If data is transmitted using an IPv4 UDP socket which is bound to a local address (TcplpAddrId) of type Multicast, then the Tcplp shall use the IP address of the configured local address (TcplpAddrId), which is of type IPv4 Unicast and assigned to the same EthIfCtrl, as the bound local address (TcplpAddrId) as source IP address in the IP datagram header.] ()

[SWS_TCPIP_00179] If data is transmitted using an IPv4 UDP socket which is bound to a local address (TcplpAddrId) of type Broadcast, then the Tcplp shall use the IP address of the configured local address (TcplpAddrId), which is of type IPv4 Unicast and assigned to the same EthIfCtrl, as the bound local address (TcplpAddrId) as source IP address in the IP datagram header.] ()

[SWS_TCPIP_00180] If data is transmitted using an IPv4 UDP socket which is not bound, then the Tcplp uses the IP address of the configured local address (TcplpAddrId), which is of type IPv4 Unicast and assigned to the EthIfCtrl in the same subnet as the destination IPv4 address as source IP address in the IP datagram header. If no matching subnet is found the IPv4 Unicast local address (TcplpAddrId) of EthIfCtrl = 0 is selected.] ()

[SWS_TCPIP_00181] If data is transmitted using a socket which is bound to an IPv6 Unicast local address (TcplpAddrId) the Tcplp shall use the IP address assigned to local address (TcplpAddrId) as source IP address in the IP datagram header. The IP datagram shall be transmitted using the EthIfCtrl the local address (TcplpAddrId) is mapped to.] ()

[SWS_TCPIP_00182] If data is transmitted using an IPv6 socket which is bound to a local address (TcplpAddrId) using the wildcard "ANY", the Tcplp shall select the source IP address of the IPv6 header according to the source address selection algorithm specified in section 5 of IETF RFC 6724 (Default Address Selection for IPv6). The selection shall be limited to the configured local addresses (TcplpAddrId) on the same EthIfCtrl as the bound local address (TcplpAddrId) only.] ()

[SWS_TCPIP_00183] If data is transmitted using an IPv6 socket which is bound to TCPIP_LOCALADDRID_ANY, the Tcplp shall select the interface that has a local address (TcplpAddrId) which uses the same network prefix as the destination address. If no matching interface is found EthIfCtrl = 0 is selected. The Tcplp shall select the source IP address of the IPv6 header according to the source address selection algorithm specified in section 5 of IETF RFC 6724 (Default Address Selection for IPv6).] ()

[SWS_TCPIP_00184] If data is transmitted using an IPv6 UDP socket which is bound to a local address (TcplpAddrId) of type Multicast, the Tcplp - shall select the source IP address of the IPv6 header according to the source address selection algorithm specified in section 5 of IETF RFC 6724 (Default Address Selection for

IPv6). The selection shall be limited to the configured local addresses (TcplpAddrId) on the same EthIfCtrl as the bound local address (TcplpAddrId) only.] ()

[SWS_TCPIP_00185] If data is transmitted using an IPv6 UDP socket which is not bound, the Tcplp shall select the interface that has a local address (TcplpAddrId) which uses the same network prefix as the destination address. If no matching interface is found EthIfCtrl = 0 is selected. The Tcplp shall select the source IP address of the IPv6 header according to the source address selection algorithm specified in section 5 of IETF RFC 6724 (Default Address Selection for IPv6).] ()

[SWS_TCPIP_00101] The Tcplp shall choose the correct next hop for each datagram it sends according to IETF RFC 1122, section 3.3.1.1. (IPv4) and IETF RFC4861 section 5.2. Conceptual Sending Algorithm (IPv6).] ()

[SWS_TCPIP_00131] Tcplp shall always call `EthIf_Transmit()` with parameter TxConfirmation set to FALSE.] ()

[SWS_TCPIP_00191] If the parameter TcplpARPPaketQueueEnabled is set to TRUE and an IPv4 packet shall be transmitted to a remote host but the related link layer address does not exist in the ARP table, the Tcplp shall start the address resolution and queue this packet according to IETF RFC 1122, section 2.3.2.2.] ()

[SWS_TCPIP_00192] If the parameter TcplpARPPaketQueueEnabled is set to FALSE and an IPv4 packet shall be transmitted to a remote host but the related link layer address does not exist in the ARP table, the Tcplp shall start the address resolution but reject the transmission request with E_NOT_OK.] ()

[SWS_TCPIP_00193] If the parameter TcplpNdpPacketQueueEnabled is set to TRUE and an IPv6 packet shall be transmitted to a remote host but the related link layer address does not exist in the Neighbor Cache, the Tcplp shall start the address resolution and queue this packet according to IETF RFC 4861, section 7.2.2.] ()

[SWS_TCPIP_00194] If the parameter TcplpNdpPacketQueueEnabled is set to FALSE and an IPv6 packet shall be transmitted to a remote host but the related link layer address does not exist in the Neighbor Cache, the Tcplp shall start the address resolution but reject the transmission request with E_NOT_OK.] ()

[SWS_TCPIP_00202] After the maximum retries configured via ECUC_Tcplp_00069 are transmitted, the timer according to TcplpTcpRetransmissionTimeout shall be restarted the last time before the TCP connection is closed.] ()

[SWS_TCPIP_00204] For transmissions the Tcplp Module shall skip the calculation of the protocol checksums and fill the field with the value 0 for frames with respect to the configuration of the Ethernet Controller according the following list:

- a) for IPv4 frames if IPv4 checksum calculation in hardware is enabled, i.e. EthCtrlEnableOffloadChecksumIPv4 is set to TRUE

- b) for not fragmented ICMP frames if ICMP checksum calculation in hardware is enabled, `EthCtrlEnableOffloadChecksumICMP` is set to TRUE
- c) for TCP frames if TCP checksum calculation in hardware is enabled, `EthCtrlEnableOffloadChecksumTCP` is set to TRUE
- d) for not fragmented UDP frames if UDP checksum calculation in hardware is enabled, `EthCtrlEnableOffloadChecksumUDP` is set to TRUE.

In all other cases, the `Tcplp` module shall calculate the checksum according the related protocol specification.] ()

7.7 TCP/IP Stack state handling

[SWS_TCPIP_00083] The `Tcplp` module shall maintain a separate state for each `EthIf` controller used by the `Tcplp` module, store the latest state request and distinguish at least the following states: `TCPIP_STATE_OFFLINE`, `TCPIP_STATE_STARTUP`, `TCPIP_STATE_ONLINE`, `TCPIP_STATE_ONHOLD`, and `TCPIP_STATE_SHUTDOWN`.] ()

[SWS_TCPIP_00136] The `Tcplp` module shall initiate according actions to achieve the requested state if the stored state request is not the active state.] ()

[SWS_TCPIP_00084] After each transition the `Tcplp` module shall report the new state to `EthSM` via `EthSM_TcpIpModeIndication()`.] ()

[SWS_TCPIP_00075] If `TCPIP_STATE_ONLINE` is requested for an `EthIf` controller and the current state is `TCPIP_STATE_OFFLINE` for that `EthIf` controller, the `Tcplp` module shall

- (a) enable all IP address assignments according to the configured assignment methods (`TcplpAssignmentMethod`) and triggers (`TcplpAssignmentTrigger`) for that `EthIf` controller. (Note: If the assignment trigger is configured to `TCPIP_MANUAL` no assignment is actually performed but initiation by the upper layer enabled) and
- (b) enter the state `TCPIP_STATE_STARTUP` for the `EthIf` controller.] ()

[SWS_TCPIP_00127] In case multiple IP address assignment methods are configured and a new address from an assignment method with a higher priority (1 is highest) becomes available, `Tcplp` shall use the new IP address and release the IP address previously assigned by an assignment method with a lower priority.] ()

[SWS_TCPIP_00088] If `TCPIP_STATE_OFFLINE` is requested for an `EthIf` controller and the current state is `TCPIP_STATE_STARTUP` for that `EthIf` controller, the `Tcplp` module shall

- (a) abort all ongoing IP address assignment actions appropriate and
- (b) enter the state `TCPIP_STATE_OFFLINE` for the `EthIf` controller.] ()

[SWS_TCPIP_00085] If at least one IP address has been successfully assigned to an `EthIf` controller and the current state is `TCPIP_STATE_STARTUP` for that `EthIf` controller, the `Tcplp` module shall enter the state `TCPIP_STATE_ONLINE` for the `EthIf` controller.] ()

Note: After successfully assignment of an IP address to the EthIf controller the upper layer module will be notified via `Up_LocalIpAddressAssignmentChg()` with State `TCPIP_IPADDR_STATE_ASSIGNED`.

[SWS_TCPIP_00076] If `TCPIP_STATE_ONHOLD` is requested for an EthIf controller and the current state is `TCPIP_STATE_ONLINE` for that EthIf controller, the Tcplp module shall

- (a) notify the upper layer via `Up_LocalIpAddressAssignmentChg()` with State `TCPIP_IPADDR_STATE_ONHOLD` for all assigned IP addresses of the related EthIf controller, and
- (b) deactivate the communication within the Tcplp module for the related EthIf controller, and
- (c) enter the state `TCPIP_STATE_ONHOLD` for the EthIf controller.] ()

[SWS_TCPIP_00086] If `TCPIP_STATE_ONLINE` is requested for an EthIf controller and the current state is `TCPIP_STATE_ONHOLD` for that EthIf controller, the Tcplp module shall

- (a) reactivate the communication within the Tcplp module for the related EthIf controller,
- (b) call `Up_LocalIpAddressAssignmentChg()` with State `TCPIP_IPADDR_STATE_ASSIGNED` for all assigned IP addresses of the related EthIf controller, and
- (c) enter the state `TCPIP_STATE_ONLINE` for the EthIf controller.] ()

[SWS_TCPIP_00077] If `TCPIP_STATE_OFFLINE` is requested or all assigned IP address have been released for an EthIf controller and the current state is `TCPIP_STATE_ONLINE` or `TCPIP_STATE_ONHOLD` for that EthIf controller, the Tcplp module shall

- (a) call `Up_LocalIpAddressAssignmentChg()` with State `TCPIP_IPADDR_STATE_UNASSIGNED` for all assigned IP addresses of the related EthIf controller,
- (b) deactivate the communication within the Tcplp module for the related EthIf controller,
- (c) release related resources, i.e. any socket using the EthIf controller shall be closed and thereafter any IP address assigned to the EthIf controller shall be unassigned,
- (d) in case the no EthIf controller is assigned any more, all unbound sockets shall be released as well, and
- (e) enter the state `TCPIP_STATE_SHUTDOWN` for the EthIf controller.] ()

[SWS_TCPIP_00087] If the current state of an EthIf controller is `TCPIP_STATE_SHUTDOWN` and all related resources have been released, the Tcplp module shall enter the state `TCPIP_STATE_OFFLINE` for the EthIf controller.] ()

[SWS_TCPIP_00094] The Tcplp module shall only accept new TCP connections if the related EthIf controller is in state `TCPIP_STATE_ONLINE`.] ()

[SWS_TCPIP_00144] The Tcplp module shall indicate events related to sockets to the upper layer module by using the Up_TcplpEvent API and the following events: TCPIP_TCP_RESET, TCPIP_TCP_CLOSED, TCPIP_TCP_FIN_RECEIVED and TCPIP_UDP_CLOSED.] ()

7.8 Error classification

This section describes how the Tcplp module has to manage the error classes that may occur during the life cycle of this basic software.

7.8.1 Development Errors

[SWS_TCPIP_00042] The following table lists development errors that shall be distinguished by the Tcplp module:

| Type or error | Relevance | Related error code | Value [hex] |
|---|-------------|-----------------------|-------------|
| API service called before initializing the module | Development | TCPIP_E_NOTINIT | 0x01 |
| API service called with NULL pointer | Development | TCPIP_E_PARAM_POINTER | 0x02 |
| Invalid argument | Development | TCPIP_E_INV_ARG | 0x03 |
| No buffer space available | Development | TCPIP_E_NOBUFS | 0x04 |
| Message too long | Development | TCPIP_E_MSGSIZE | 0x07 |
| Protocol wrong type for socket | Development | TCPIP_E_PROTOTYPE | 0x08 |
| Address already in use | Development | TCPIP_E_ADDRINUSE | 0x09 |
| Can't assign requested address | Development | TCPIP_E_ADDRNOTAVAIL | 0x0A |
| Socket is already connected | Development | TCPIP_E_ISCONN | 0x0B |
| Socket is not connected | Development | TCPIP_E_NOTCONN | 0x0C |
| Protocol not available | Development | TCPIP_E_NOPROTOOPT | 0x0D |
| Address family not supported by protocol family | Development | TCPIP_E_AFNOSUPPORT | 0x0E |
| Invalid configuration set selection | Development | TCPIP_E_INIT_FAILED | 0x0F |

/ ()

7.8.2 Production Errors

There are no production errors.

7.8.3 Extended Production Errors

[SWS_TCPIP_00043] The following table lists production errors that shall be distinguished by the Tcplp module:

| <i>Type or error</i> | <i>Relevance</i> | <i>Related error code</i> | <i>Value [hex]</i> |
|-----------------------------------|---------------------|---------------------------|--------------------|
| Operation timed out | Extended Production | TCPIP_E_TIMEDOUT | Assigned by DEM |
| Connection refused | Extended Production | TCPIP_E_CONNREFUSED | Assigned by DEM |
| No route to host | Extended Production | TCPIP_E_HOSTUNREACH | Assigned by DEM |
| Path does not support frame size. | Extended Production | TCPIP_E_PACKETTOBIG | Assigned by DEM |

] ()

Note: Values for production code Event Ids are assigned externally by the configuration of the Dem. They are published in the file Dem_IntErrId.h and included via Dem.h.

7.8.3.1 TCPIP_E_TIMEDOUT

[SWS_TCPIP_00207]

| | | |
|------------------------------|--|---|
| Error Name: | TCPIP_E_TIMEDOUT | |
| Short Description: | Timeout of TCP connection | |
| Long Description: | The Tcplp module has sent a SYN to establish a connection but did not receive any response or an established TCP connection is closed because the peer no longer responds, i.e. keep-alive timer runs out and peer does not respond to keep-alive probes according to IETF RFC 1122 chapter 4.2.3.6 TCP Keep-Alives. | |
| Detection Criteria: | Fail | Timeout during TCP connection establishment or timeout after keep-alive probes. |
| | Pass | N/A |
| Secondary Parameters: | N/A | |
| Time Required: | Configuration dependent | |
| Monitor Frequency | Continuous | |

] ()

[SWS_TCPIP_00211] If the optional configuration parameter TCPIP_E_TIMEDOUT (ECUC_Tcplp_00048) exists, the Tcplp shall report the extended production error by calling Dem_ReportErrorStatus (TCPIP_E_TIMEDOUT, DEM_EVENT_STATUS_FAILED) if one of the following conditions applies:

- a) Tcplp module has sent a SYN to establish a connection but did not receive any response.

- b) An established idle TCP connection is closed because the peer is no longer present, i.e. keep-alive timer runs out and peer does not respond to keep-alive probes according to IETF RFC 1122 chapter 4.2.3.6 TCP Keep-Alives.
- c) An established TCP connection is closed because the peer does not respond, i.e. the maximum number of retransmissions has been sent without acknowledgement, according to [SWS_TCPIP_00202].] ()

7.8.3.2 TCPIP_E_CONNREFUSED

[SWS_TCPIP_00208][

| | | |
|------------------------------|---|--|
| Error Name: | TCPIP_E_CONNREFUSED | |
| Short Description: | Connection refused by the peer. | |
| Long Description: | The peer doesn't provide a service at the requested protocol or port and responds with ICMP message Destination Unreachable/Protocol Unreachable or Destination Unreachable/Port Unreachable. | |
| Detection Criteria: | Fail | Connection has been refused by ICMP message. |
| | Pass | N/A |
| Secondary Parameters: | N/A | |
| Time Required: | Configuration dependent | |
| Monitor Frequency | Continuous | |

] ()

[SWS_TCPIP_00212][If the optional configuration parameter TCPIP_E_CONNREFUSED (ECUC_Tcplp_00049) exists, the Tcplp shall report the extended production error by calling Dem_ReportErrorStatus (TCPIP_E_CONNREFUSED, DEM_EVENT_STATUS_FAILED) if one of the following conditions applies:

- a) An ICMP message Destination Unreachable/Protocol Unreachable is received because the peer doesn't provide a service at the requested protocol.
- b) An ICMP message Destination Unreachable/Port Unreachable is received because the peer doesn't provide a service at the requested port.] ()

7.8.3.3 TCPIP_E_HOSTUNREACH

[SWS_TCPIP_00209][

| | | |
|------------------------------|--|--|
| Error Name: | TCPIP_E_HOSTUNREACH | |
| Short Description: | The peer host is unreachable. | |
| Long Description: | A frame can't be forwarded to the peer because the network or host is unreachable or there is no route to the destination. | |
| Detection Criteria: | Fail | Reception of ICMP message signalling that the host is unreachable. |
| | Pass | N/A |
| Secondary Parameters: | N/A | |

| | |
|--------------------------|-------------------------|
| Time Required: | Configuration dependent |
| Monitor Frequency | Continuous |

] ()

[SWS_TCPIP_00213] If the optional configuration parameter TCPIP_E_HOSTUNREACH (ECUC_Tcplp_00051) exists, the Tcplp shall report the extended production error by calling Dem_ReportErrorStatus (TCPIP_E_HOSTUNREACH, DEM_EVENT_STATUS_FAILED) if one of the following conditions applies:

- a) An ICMP message Destination Unreachable is received because the network or host is unreachable or there is no route to the destination.] ()

7.8.3.4 TCPIP_E_PACKETTOBIG

[SWS_TCPIP_00210]

| | | |
|------------------------------|--|---|
| Error Name: | TCPIP_E_PACKETTOBIG | |
| Short Description: | Path does not support frame size. | |
| Long Description: | The network can't forward an oversized frame because of the DF Flag (don't fragment) | |
| Detection Criteria: | Fail | Reception of ICMP message signalling the packet is too big and fragmentation is needed but not allowed. |
| | Pass | N/A |
| Secondary Parameters: | N/A | |
| Time Required: | Configuration dependent | |
| Monitor Frequency | Continuous | |

] ()

[SWS_TCPIP_00214] If the optional configuration parameter TCPIP_E_PACKETTOBIG (ECUC_Tcplp_00196) exists, the Tcplp shall report the extended production error by calling Dem_ReportErrorStatus (TCPIP_E_PACKETTOBIG, DEM_EVENT_STATUS_FAILED) if one of the following conditions applies:

- a) An ICMP message Destination Unreachable/ Fragmentation needed but DF bit set is received because the network can't forward an oversized frame since the DF (don't fragment) Flag is set.] ()

7.9 Application notes

7.10 Debugging Concept

For details refer to the chapter 7.1.17 “Debugging support” in *SWS_BSWGeneral*.

7.11 Version checking

For details refer to the chapter 5.1.8 “Version Check” in *SWS_BSWGeneral*.

8 API specification

8.1 Imported types

The following types shall be imported by the Tcplp from the modules given:

[SWS_TCPIP_00008] [

| Module | Imported Type |
|------------------|----------------------|
| ComStack_Types | BufReq_ReturnType |
| Dem | Dem_EventIdType |
| | Dem_EventStatusType |
| Eth_GeneralTypes | Eth_BufIdxType |
| | Eth_FilterActionType |
| | Eth_FrameType |
| Std_Types | Std_ReturnType |
| | Std_VersionInfoType |

] ()

8.2 Type definitions

[SWS_TCPIP_00067][

| | | | |
|---------------------|---|----|---|
| Name: | TcpIp_ConfigType | | |
| Type: | -- | | |
| Range: | implementation specific | -- | The content of the configuration data structure is implementation specific. |
| Description: | Configuration data structure of the Tcplp module. | | |

] ()

[SWS_TCPIP_00009] [

| | | | |
|---------------------|-------------------------|------|----------|
| Name: | TcpIp_DomainType | | |
| Type: | uint16 | | |
| Range: | TCPIP_AF_INET | 0x02 | Use IPv4 |
| | TCPIP_AF_INET6 | 0x1c | Use IPv6 |
| Description: | TcpIp address families. | | |

] ()

[SWS_TCPIP_00010] [

| | | | |
|---------------------|---------------------------------|---------------|--|
| Name: | TcpIp_ProtocolType | | |
| Type: | Enumeration | | |
| Range: | TCPIP_IPPROTO_TCP | 0x06: Use TCP | |
| | TCPIP_IPPROTO_UDP | 0x11: Use UDP | |
| Description: | Protocol type used by a socket. | | |

] ()

[SWS_TCPIP_00012][

| | | | |
|--------------|--------------------|--|--|
| Name: | TcpIp_SockAddrType | | |
| Type: | Structure | | |

| | | | |
|---------------------|---|--------|---|
| Element: | TcpIp_DomainType | domain | This is the code for the address format of this address |
| Description: | Generic structure used by APIs to specify an IP address. (A specific address type can be derived from this structure via a cast to the specific struct type.) | | |

] ()

[SWS_TCPIP_00013]

| | | | |
|---------------------|---|--------|---|
| Name: | TcpIp_SockAddrInetType | | |
| Type: | Structure | | |
| Element: | TcpIp_DomainType | domain | This is the code for the address format of this address |
| | uint16 | port | port number |
| | uint32[1] | addr | IPv4 address in network byte order |
| Description: | This structure defines an IPv4 address type which can be derived from the generic address structure via cast. | | |

] ()

[SWS_TCPIP_00014]

| | | | |
|---------------------|--|--------|---|
| Name: | TcpIp_SockAddrInet6Type | | |
| Type: | Structure | | |
| Element: | TcpIp_DomainType | domain | This is the code for the address format of this address |
| | uint16 | port | port number |
| | uint32[4] | addr | IPv6 address in network byte order |
| Description: | This structure defines a IPv6 address type which can be derived from the generic address structure via cast. | | |

] ()

[SWS_TCPIP_00030]

| | | | |
|---------------------|--|--|--|
| Name: | TcpIp_LocalAddrIdType | | |
| Type: | uint8 | | |
| Description: | Address identification type for unique identification of a local IP address and EthIf Controller configured in the Tcplp module. | | |

] ()

[SWS_TCPIP_00038]

| | | | |
|---------------------|---|--|--|
| Name: | TcpIp_SocketIdType | | |
| Type: | uint8, uint16 | | |
| Description: | socket identifier type for unique identification of a Tcplp stack socket. TCPIP_SOCKETID_INVALID shall specify an invalid socket handle | | |

] ()

[SWS_TCPIP_00073]

| | | | |
|---------------|--------------------|---|--|
| Name: | TcpIp_StateType | | |
| Type: | Enumeration | | |
| Range: | TCPIP_STATE_ONLINE | TCP/IP stack state for a specific EthIf controller is ONLINE, i.e. communication via at least one IP address is possible. | |

| | | |
|---------------------|--|---|
| | TCPIP_STATE_ONHOLD | TCP/IP stack state for a specific Ethlf controller is ONHOLD, i.e. no communication is currently possible (e.g. link down). |
| | TCPIP_STATE_OFFLINE | TCP/IP stack state for a specific Ethlf controller is OFFLINE, i.e. no communication is possible. |
| | TCPIP_STATE_STARTUP | TCP/IP stack state for a specific Ethlf controller is STARTUP, i.e. IP address assignment in progress or ready for manual start, communication is currently not possible. |
| | TCPIP_STATE_SHUTDOWN | TCP/IP stack state for a specific Ethlf controller is SHUTDOWN, i.e. release of resources using the Ethlf controller, release of IP address assignment. |
| Description: | Specifies the Tcplp state for a specific Ethlf controller. | |

] ()

[SWS_TCPIP_00082]

| | | |
|---------------------|--|---|
| Name: | TcpIp_IpAddrStateType | |
| Type: | Enumeration | |
| Range: | TCPIP_IPADDR_STATE_ASSIGNED | local IP address is assigned |
| | TCPIP_IPADDR_STATE_ONHOLD | local IP address is assigned, but cannot be used as the network is not active |
| | TCPIP_IPADDR_STATE_UNASSIGNED | local IP address is unassigned |
| Description: | Specifies the state of local IP address assignment | |

] ()

[SWS_TCPIP_00031]

| | | |
|---------------------|---------------------------|--|
| Name: | TcpIp_EventType | |
| Type: | Enumeration | |
| Range: | TCPIP_TCP_RESET | TCP connection was reset, TCP socket and all related resources have been released. |
| | TCPIP_TCP_CLOSED | TCP connection was closed successfully, TCP socket and all related resources have been released. |
| | TCPIP_TCP_FIN_RECEIVED | A FIN signal was received on the TCP connection, TCP socket is still valid. |
| | TCPIP_UDP_CLOSED | UDP socket and all related resources have been released. |
| Description: | Events reported by Tcplp. | |

] ()

[SWS_TCPIP_00065]

| | | |
|---------------|--|---|
| Name: | TcpIp_IpAddrAssignmentType | |
| Type: | Enumeration | |
| Range: | TCPIP_IPADDR_ASSIGNMENT_STATIC | Static configured IPv4/IPv6 address. |
| | TCPIP_IPADDR_ASSIGNMENT_LINKLOCAL_DOIP | Linklocal IPv4/IPv6 address assignment using DoIP parameters. |
| | TCPIP_IPADDR_ASSIGNMENT_DHCP | Dynamic configured IPv4/IPv6 address by DHCP. |
| | TCPIP_IPADDR_ASSIGNMENT_LINKLOCAL | Linklocal IPv4/IPv6 address assignment. |
| | TCPIP_IPADDR_ASSIGNMENT_IPV6_ROUTER | Dynamic configured IPv4/IPv6 address by |

| | |
|---------------------|---|
| | Router Advertisement. |
| Description: | Specification of IPv4/IPv6 address assignment policy. |

] ()

[SWS_TCPIP_00066]

| | | |
|---------------------|-----------------------------|--|
| Name: | TcpIp_ReturnType | |
| Type: | Enumeration | |
| Range: | TCPIP_OK | operation completed successfully. |
| | TCPIP_E_NOT_OK | operation failed. |
| | TCPIP_E_PHYS_ADDR_MISS | operation failed because of an ARP/NDP cache miss. |
| Description: | Tcplp specific return type. | |

] ()

[SWS_TCPIP_00126]

| | | | |
|---------------|--|------|--|
| Name: | TcpIp_ParamIdType | | |
| Type: | uint8 | | |
| Range: | TCPIP_PARAMID_TCP_RXWND_MAX | 0x00 | Specifies the maximum TCP receive window for the socket. |
| | TCPIP_PARAMID_FRAMEPRIO | 0x01 | Specifies the frame priority for outgoing frames on the socket. |
| | TCPIP_PARAMID_TCP_NAGLE | 0x02 | Specifies if the Nagle Algorithm according to IETF RFC 896 is enabled or not. |
| | TCPIP_PARAMID_TCP_KEEPALIVE | 0x03 | Specifies if TCP Keep Alive Probes are sent on the socket connection. |
| | TCPIP_PARAMID_TTL | 0x04 | Specifies the time to live value for outgoing frames on the socket. For IPv6 this parameter specifies the value of the HopLimit field used in the IPv6 header. |
| | TCPIP_PARAMID_TCP_KEEPALIVE_TIME | 0x05 | Specifies the time in [s] between the last data packet sent (simple ACKs are not considered data) and the first keepalive probe. |
| | TCPIP_PARAMID_TCP_KEEPALIVE_PROBES_MAX | 0x06 | Specifies the maximum number of times that a keepalive probe is retransmitted. |
| | TCPIP_PARAMID_TCP_KEEPALIVE_INTERVAL | 0x07 | Specifies the interval in [s] between subsequent keepalive probes. |
| | TCPIP_PARAMID_VENDOR_SPECIFIC | 0x80 | Start of vendor specific range of |

| | | | |
|---------------------|--|--|----------------|
| | | | parameter IDs. |
| Description: | Type for the specification of all supported Parameter IDs. | | |

] ()

[SWS_TCPIP_00133]

| | | | |
|---------------------|-------------------------|-------------------------|------------------------------------|
| Name: | TcpIpIpAddrWildcardType | | |
| Type: | uint32 | | |
| Range: | TCPIP_IPADDR_ANY | implementation specific | defines the value used as wildcard |
| Description: | IP address wildcard. | | |

] ()

[SWS_TCPIP_00132]

| | | | |
|---------------------|--------------------------|-------------------------|--|
| Name: | TcpIpIp6AddrWildcardType | | |
| Type: | uint32 | | |
| Range: | TCPIP_IP6ADDR_ANY | implementation specific | defines the value used as wildcard for all IP6 address parts |
| Description: | IP6 address wildcard. | | |

] ()

[SWS_TCPIP_00134]

| | | | |
|---------------------|-----------------------|-------------------------|------------------------------------|
| Name: | TcpIpPortWildcardType | | |
| Type: | uint16 | | |
| Range: | TCPIP_PORT_ANY | implementation specific | defines the value used as wildcard |
| Description: | Port wildcard. | | |

] ()

[SWS_TCPIP_00135]

| | | | |
|---------------------|------------------------------|-------------------------|------------------------------------|
| Name: | TcpIpLocalAddrIdWildcardType | | |
| Type: | TcpIp_LocalAddrIdType | | |
| Range: | TCPIP_LOCALADDRID_ANY | implementation specific | defines the value used as wildcard |
| Description: | LocalAddrId wildcard. | | |

] ()

8.3 Function definitions

This is a list of functions provided for upper layer modules.

8.3.1 General

8.3.1.1 Tcplp_Init

[SWS_TCPIP_00002]

| | | | |
|----------------------|---|--|--|
| Service name: | Tcplp_Init | | |
| Syntax: | <pre>void Tcplp_Init(const TcpIp_ConfigType* ConfigPtr)</pre> | | |

| | |
|----------------------------|--|
| Service ID[hex]: | 0x01 |
| Sync/Async: | Synchronous |
| Reentrancy: | Non Reentrant |
| Parameters (in): | ConfigPtr Pointer to the configuration data of the Tcplp module |
| Parameters (inout): | None |
| Parameters (out): | None |
| Return value: | void None |
| Description: | This service initializes the TCP/IP Stack. Tcplp_Init may not block the start-up process for an indefinite amount of time. Caveats: The call of this service is mandatory before using the Tcplp instance for further processing. |

] ()

8.3.1.2 Tcplp_GetVersionInfo

[SWS_TCPIP_00004] [

| | |
|----------------------------|---|
| Service name: | Tcplp_GetVersionInfo |
| Syntax: | <pre>void TcpIp_GetVersionInfo (Std_VersionInfoType* versioninfo)</pre> |
| Service ID[hex]: | 0x02 |
| Sync/Async: | Synchronous |
| Reentrancy: | Reentrant |
| Parameters (in): | None |
| Parameters (inout): | None |
| Parameters (out): | versioninfo Pointer to where to store the version information of this module. |
| Return value: | None |
| Description: | Returns the version information. |

] ()

[SWS_TCPIP_00005][[The function Tcplp_GetVersionInfo shall return the version information of this module. The version information includes:

- Module Id
- Vendor Id
- Vendor specific version numbers (BSW00407).

] ()

[SWS_TCPIP_00006][The function Tcplp_GetVersionInfo shall be pre compile time configurable On/Off by the configuration parameter:

TCPIP_VERSION_INFO_API] ()

8.3.2 Core Communication Control

8.3.2.1 Tcplp_Close

[SWS_TCPIP_00017][

| | |
|----------------------|------------------------------|
| Service name: | Tcplp_Close |
| Syntax: | Std_ReturnType TcpIp_Close (|

| | | |
|----------------------------|--|--|
| | TcpIp_SocketIdType SocketId, boolean Abort) | |
| Service ID[hex]: | 0x04 | |
| Sync/Async: | Asynchronous | |
| Reentrancy: | Reentrant for different SocketIds. Non reentrant for the same SocketId. | |
| Parameters (in): | SocketId | Socket handle identifying the local socket resource. |
| | Abort | TRUE: connection will immediately be terminated by sending a RST-Segment and releasing all related resources. FALSE: connection will be terminated after performing a regular connection termination handshake and releasing all related resources. |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | Std_ReturnType | E_OK: The request has been accepted E_NOT_OK: The request has not been accepted. |
| Description: | By this API service the TCP/IP stack is requested to close the socket and release all related resources. | |

] ()

[SWS_TCPIP_00109] The service Tcplp_Close() shall perform the following actions for the socket specified by SocketId in case it is a TCP socket:

- (a) if the connection is active and
 - (a1) abort = FALSE: the connection shall be terminated after performing a regular connection termination handshake and releasing all related resources.
 - (a2) abort = TRUE: connection shall immediately be terminated by sending a RST-Segment and releasing all related resources.
- (b) if the socket is in the Listen state, the Listen state shall be left immediately and related resources shall be released.] ()

[SWS_TCPIP_00110] The service Tcplp_Close() shall release all related resources immediately for the socket specified by SocketId in case it is a UDP socket .] ()

Note: The upper layer will be notified via Up_TcplpEvent(TCPIP_TCP_CLOSED, TCPIP_TCP_RESET or TCPIP_UDP_CLOSED) after the socket and all related resources have been released. After this call the SocketId is invalid until allocated again with Tcplp_GetSocket().

8.3.2.2 Tcplp_Bind

[SWS_TCPIP_00015]

| | | |
|-------------------------|--|--|
| Service name: | Tcplp_Bind | |
| Syntax: | Std_ReturnType Tcplp_Bind(TcpIp_SocketIdType SocketId, TcpIp_LocalAddrIdType LocalAddrId, uint16* PortPtr) | |
| Service ID[hex]: | 0x05 | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Reentrant for different SocketIds. Non reentrant for the same SocketId. | |

| | | |
|----------------------------|--|--|
| Parameters (in): | SocketId | Socket identifier of the related local socket resource. |
| | LocalAddrId | IP address identifier representing the local IP address and EthIf controller to bind the socket to. Note: to listen to all EthIf controller, TCPIP_LOCALADDRID_ANY has to be specified as LocalAddrId. Note: to listen on any IP addresss of a EthIf controller, the configuration parameter TcplpStaticIpAddress referenced by LocalAddrId must be set to "ANY". The remote IP address of an incoming packet has no effect then. In case the socket shall be used as client socket, the IP address and EthIf controller represented by LocalAddrId is used for transmission. Note: for an automatic selection of the Local IP address and EthIf Controller, TCPIP_LOCALADDRID_ANY has to be specified as LocalAddrId. |
| Parameters (inout): | PortPtr | Pointer to memory where the local port to which the socket shall be bound is specified. In case the parameter is specified as TCPIP_PORT_ANY, the TCP/IP stack shall choose the local port automatically from the range 49152 to 65535 and shall update the parameter to the chosen value. |
| Parameters (out): | None | |
| Return value: | Std_ReturnType | Result of operation E_OK The request has been accepted E_NOT_OK The request has not been accepted (e.g. address in use) |
| Description: | By this API service the TCP/IP stack is requested to bind a UDP or TCP socket to a local resource. | |

] ()

[SWS_TCPIP_00111] The service Tcplp_Bind() shall bind the socket specified by parameter SocketId to the local resource specified by parameters LocalAddrId and PortPtr.] ()

Note: Sockets that shall be switched in a listening state later on must be bound to a local resource. Optionally this API can be used to specify the local IP address and port used by later calls of Tcplp_TcpConnect() or Tcplp_UdpTransmit().

[SWS_TCPIP_00146] TcpIp_Bind() shall check if there is another socket already bound to the same port, protocol and local address and if that is the case refuse the request and return E_NOT_OK. If development error detection is enabled, the service TcpIp_Bind() shall also raise the development error code TCPIP_E_ADDRINUSE.] ()

[SWS_TCPIP_00147] TcpIp_Bind() shall check if the local address specified by LocalAddrId is assigned and valid and if that is not the case refuse the request and return E_NOT_OK. If development error detection is enabled, the service TcpIp_Bind() shall also raise the development error code TCPIP_E_ADDRNOTAVAIL.] ()

8.3.2.3 Tcplp_TcpConnect

[SWS_TCPIP_00022]

| | | |
|----------------------------|---|---|
| Service name: | Tcplp_TcpConnect | |
| Syntax: | Std_ReturnType TcpIp_TcpConnect (TcpIp_SocketIdType SocketId, const TcpIp_SockAddrType* RemoteAddrPtr) | |
| Service ID[hex]: | 0x06 | |
| Sync/Async: | Asynchronous | |
| Reentrancy: | Reentrant for different SocketIds. Non reentrant for the same SocketId. | |
| Parameters (in): | SocketId | Socket identifier of the related local socket resource. |
| | RemoteAddrPtr | IP address and port of the remote host to connect to. |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | Std_ReturnType | E_OK: The request has been accepted E_NOT_OK: The request has not been accepted, e.g. connection is already established or no route to destination specified by remoteAddrPtr found. |
| Description: | By this API service the TCP/IP stack is requested to establish a TCP connection to the configured peer. | |

] ()

[SWS_TCPIP_00112] The service Tcplp_TcpConnect() shall establish a TCP connection between the local socket specified by parameter SocketId and the remote socket specified with parameter RemoteAddrPtr.] ()

[SWS_TCPIP_00129] [If development error detection is enabled and the parameter RemoteAddrPtr equals NULL_PTR, the Tcplp_TcpConnect function shall raise the development error code TCPIP_E_PARAM_POINTER and the Tcplp_TcpConnect function shall return E_NOT_OK.] ()

8.3.2.4 Tcplp_TcpListen

[SWS_TCPIP_00023]

| | | |
|----------------------------|---|---|
| Service name: | Tcplp_TcpListen | |
| Syntax: | Std_ReturnType TcpIp_TcpListen (TcpIp_SocketIdType SocketId, uint16 MaxChannels) | |
| Service ID[hex]: | 0x07 | |
| Sync/Async: | Asynchronous | |
| Reentrancy: | Reentrant for different SocketIds. Non reentrant for the same SocketId. | |
| Parameters (in): | SocketId | Socket identifier of the related local socket resource. |
| | MaxChannels | Maximum number of new parallel connections established on this listen connection. |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | Std_ReturnType | E_OK: The request has been accepted E_NOT_OK: The request has not been accepted, the socket is not configured to be a server socket. |
| Description: | By this API service the TCP/IP stack is requested to listen on the | |

| | |
|--|--|
| | TCP socket specified by the socket identifier. |
|--|--|

] ()

[SWS_TCPIP_00113] The service Tcplp_TcpListen() shall put the socket specified by SocketId to the listen state (i.e. local socket is listening for incoming connections).

] ()

[SWS_TCPIP_00114] Tcplp shall derive a separate socket from the listen socket to establish a new connection from an incoming connection request on the listen socket and limit the number of new parallel connections to the value specified by MaxChannels.] ()

8.3.2.5 Tcplp_TcpReceived

[SWS_TCPIP_00024]

| | | |
|----------------------------|---|--|
| Service name: | Tcplp_TcpReceived | |
| Syntax: | Std_ReturnType Tcplp_TcpReceived(TcpIp_SocketIdType SocketId, uint32 Length) | |
| Service ID[hex]: | 0x08 | |
| Sync/Async: | Asynchronous | |
| Reentrancy: | Reentrant for different SocketIds. Non reentrant for the same SocketId. | |
| Parameters (in): | SocketId | Socket identifier of the related local socket resource. |
| | Length | Number of bytes finally consumed by the upper layer. |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | Std_ReturnType | E_OK: The request has been accepted E_NOT_OK: The request has not been accepted |
| | | |
| Description: | By this API service the reception of socket data is confirmed to the TCP/IP stack. | |

] ()

[SWS_TCPIP_00115] The service Tcplp_TcpReceived() shall increase the TCP receive window of the socket specified by SocketId considering the number of finally consumed bytes specified by Length.] ()

8.3.2.6 Tcplp_RequestComMode

[SWS_TCPIP_00070]

| | | |
|-------------------------|--|--|
| Service name: | Tcplp_RequestComMode | |
| Syntax: | Std_ReturnType Tcplp_RequestComMode(uint8 CtrlIdx, TcpIp_StateType State) | |
| Service ID[hex]: | 0x09 | |
| Sync/Async: | Asynchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | CtrlIdx | EthIf controller index to identify the communication network where the Tcplp state is requested. |
| | | |

| | | |
|----------------------------|--|--|
| | State | Requested Tcplp state. |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | Std_ReturnType | E_OK: Service accepted E_NOT_OK: Service denied |
| Description: | By this API service the TCP/IP stack is requested to change the Tcplp state of the communication network identified by EthIf controller index. | |

] ()

[SWS_TCPIP_00071] If TCPIP_STATE_ONLINE is requested, the Tcplp module shall initiate activation of the Tcplp communication on the related EthIf controller (e.g. start IP-Address assignment according to the configured IP address assignment policy for the EthIf controller).] ()

[SWS_TCPIP_00072] If TCPIP_STATE_OFFLINE is requested, the Tcplp module shall initiate deactivation of the Tcplp communication on the related EthIf controller (e.g. close all sockets using the specified EthIf controller).] ()

[SWS_TCPIP_00074] If TCPIP_STATE_ONHOLD is requested, the Tcplp module shall set the Tcplp communication to on hold, i.e. new transmit requests shall not be accepted, but sockets and assigned IP addresses shall be kept.] ()

[SWS_TCPIP_00089] If TCPIP_STATE_STARTUP or TCPIP_STATE_SHUTDOWN is requested as state the function Tcplp_RequestComMode shall abort with E_NOT_OK and report TCPIP_E_INV_ARG to DET if DET is enabled.] ()

Note: According to [SWS_TCPIP_00075] and [SWS_TCPIP_00077] TCPIP_STATE_STARTUP or TCPIP_STATE_SHUTDOWN are intermediate states arising from requesting TCPIP_STATE_OFFLINE or TCPIP_STATE_ONLINE. Requesting these intermediate states is not useful.

8.3.3 Extended Communication Control and Information

8.3.3.1 Tcplp_RequestIpAddrAssignment

[SWS_TCPIP_00037]

| | | |
|-------------------------|---|--|
| Service name: | Tcplp_RequestIpAddrAssignment | |
| Syntax: | <pre>Std_ReturnType Tcplp_RequestIpAddrAssignment (TcpIp_LocalAddrIdType LocalAddrId, TcpIp_IpAddrAssignmentType Type, const TcpIp_SockAddrType* LocalIpAddrPtr, uint8 Netmask, const TcpIp_SockAddrType* DefaultRouterPtr)</pre> | |
| Service ID[hex]: | 0x0A | |
| Sync/Async: | Asynchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | LocalAddrId | IP address index specifying the IP address for which an assignment shall be initiated. |

| | | |
|----------------------------|---|--|
| | Type | Type of IP address assignment which shall be initiated |
| | LocalIpAddrPtr | Pointer to structure containing the IP address which shall be assigned to the EthIf controller indirectly specified via LocalAddrId. Note: This parameter is only used in case the parameter Type is set to TCPIP_IPADDR_ASSIGNMENT_STATIC, can be set to NULL_PTR otherwise. |
| | Netmask | Network mask of IPv4 address or address prefix of IPv6 address in CIDR Notation. Note: This parameter is only used in case the parameter Type is set to TCPIP_IPADDR_ASSIGNMENT_STATIC. |
| | DefaultRouterPtr | Pointer to structure containing the IP address of the default router (gateway) which shall be assigned. Note: This parameter is only used in case the parameter Type is set to TCPIP_IPADDR_ASSIGNMENT_STATIC, can be set to NULL_PTR otherwise. |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | Std_ReturnType | E_OK: The request has been accepted E_NOT_OK: The request has not been accepted |
| Description: | By this API service the local IP address assignment for the IP address specified by LocalAddrId shall be initiated. | |

] ()

[SWS_TCPIP_00116] The service `TcpIp_RequestIpAddrAssignment()` shall initiate the local IP address assignment according to the IP address table entry specified by `LocalAddrId` using the method specified by `Type`.] ()

[SWS_TCPIP_00079] In case `Tcplp_RequestIpAddrAssignment()` is called with parameter `Type` set to `TCPIP_IPADDR_ASSIGNMENT_STATIC` and no `TcplpStaticIpAddrConfig` container is configured for the `LocalAddr` specified by parameter `LocalAddrId`, `Tcplp` shall assign the IP address, netmask and default router specified by parameter `LocalIpAddrPtr`, `Netmask` and `DefaultRouterPtr` as soon as `TCPIP_STATE_ONLINE` is requested or immediately if already requested.] ()

[SWS_TCPIP_00080] In case a multicast address is assigned, `Tcplp` shall derive the related physical address from the multicast IP address and add the derived address to the Eth MAC address filter by calling `EthIf_UpdatePhysAddrFilter()` with action set to `ETH_ADD_TO_FILTER`.] ()

[SWS_TCPIP_00195] If `TcpIp_RequestIpAddrAssignment` is called for a `LocalAddrId` configured with `TcplpAssignmentTrigger` set to `TCPIP_MANUAL`, `Tcplp` shall consider the related assignment as available.] ()

[SWS_TCPIP_00196] If `TcpIp_ReleaseIpAddrAssignment` is called for a `LocalAddrId` configured with `TcplpAssignmentTrigger` set to `TCPIP_MANUAL`, `Tcplp` shall consider the related assignment as unavailable.] ()

[SWS_TCPIP_00197] TcplpAddrAssignments configured with TcplpAssignmentTrigger set to TCPIP_AUTOMATIC shall always be available.] ()

[SWS_TCPIP_00198] If Tcplp_RequestIpAddrAssignment is called for a LocalAddrId configured with TcplpAssignmentTrigger set to TCPIP_AUTOMATIC, Tcplp shall reject the request and return E_NOT_OK.] ()

[SWS_TCPIP_00199] If Tcplp_ReleaseIpAddrAssignment is called for a LocalAddrId configured with TcplpAssignmentTrigger set to TCPIP_AUTOMATIC, Tcplp shall reject the request and return E_NOT_OK.] ()

8.3.3.2 Tcplp_ReleaseIpAddrAssignment

[SWS_TCPIP_00078]

| | | |
|----------------------------|--|---|
| Service name: | Tcplp_ReleaseIpAddrAssignment | |
| Syntax: | Std_ReturnType TcpIp_ReleaseIpAddrAssignment (TcpIp_LocalAddrIdType LocalAddrId) | |
| Service ID[hex]: | 0x0B | |
| Sync/Async: | Asynchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | LocalAddrId | IP address index specifying the IP address for which an assignment shall be released. |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | Std_ReturnType | E_OK: The request has been accepted E_NOT_OK: The request has not been accepted |
| Description: | By this API service the local IP address assignment for the IP address specified by LocalAddrId shall be released. | |

] ()

[SWS_TCPIP_00117] The service TcpIp_ReleaseIpAddrAssignment() shall release the local IP address assignment related to the IP address table entry specified by LocalAddrId.] ()

8.3.3.3 Tcplp_ResetIpAssignment

[SWS_TCPIP_00215]

| | | |
|----------------------------|---|--|
| Service name: | Tcplp_ResetIpAssignment | |
| Syntax: | Std_ReturnType TcpIp_ResetIpAssignment (void) | |
| Service ID[hex]: | 0x1b | |
| Sync/Async: | Synchronous /Asynchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | None | |
| Parameters (inout): | None | |
| Parameters (out): | None | |

| | | |
|----------------------|--|---|
| Return value: | Std_ReturnType | E_OK: success E_NOT_OK: switch port could not be initialized |
| Description: | Resets all learned IP-addresses to invalid values. | |

] ()

[SWS_TCPIP_00216] The service `Tcplp_ResetIpAssignment()` shall reset all persistently stored IP addresses in the `NvMBlock` (see **ECUC_Tcplp_00184** :) to invalid values (e.g. to 0.0.0.0 for IPv4 addresses).] (*SRS_Eth_00087*)

Note: The next time the `TcplpAddrAssignments` configured with `TCPIP_STORE` are started, the related address assignment method are started to obtain new IP addresses.

[SWS_TCPIP_00217] The service `TcpIp_ResetIpAssignment()` shall be pre compile time configurable On/Off by the configuration parameter: `TcpIpResetIPAssignmentApi` (see **ECUC_Tcplp_00182** :).] (*SRS_Eth_00087*)

8.3.3.4 Tcplp_IcmpTransmit

[SWS_TCPIP_00039]

| | | |
|----------------------------|--|--|
| Service name: | Tcplp_IcmpTransmit | |
| Syntax: | <pre>Std_ReturnType TcpIp_IcmpTransmit(TcpIp_LocalAddrIdType LocalIpAddrId, const TcpIp_SockAddrType* RemoteAddrPtr, uint8 Ttl, uint8 Type, uint8 Code, uint16 DataLength, const uint8* DataPtr)</pre> | |
| Service ID[hex]: | 0x0C | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | LocalIpAddrId | IP address identifier representing the local IP address and EthIf controller which shall be used for transmission of the ICMP message. |
| | RemoteAddrPtr | pointer to struct representing the remote address |
| | Ttl | Time to live value to be used for the ICMP message. If 0 is specified the default value shall be used. |
| | Type | type field value to be used in the ICMP message (Note: the value of the type field determines the format of the remaining ICMP message data) |
| | Code | code field value to be used in the ICMP message |
| | DataLength | length of ICMP message |
| | DataPtr | Pointer to data which shall be sent as ICMP message data |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | Std_ReturnType | Result of operation E_OK The ICMP message has been sent successfully E_NOT_OK The ICMP message was not sent. |
| Description: | By this API service the TCP/IP stack sends an ICMP message according to the specified parameters. | |

] ()

[SWS_TCPIP_00118] The service `TcpIp_IcmpTransmit()` shall (a) construct an ICMP message according to the parameters `Type`, `Code`, `DataLength` and `DataPtr` and (b) transmit the ICMP message using the local IP address and `EthIf` controller specified by `LocalIpAddrId` to the destination specified by `RemoteAddrPtr` using a time to live value according to the parameter `Ttl`.] ()

8.3.3.5 `Tcplp_IcmpV6Transmit`

[SWS_TCPIP_00187]

| | | |
|----------------------------|---|---|
| Service name: | <code>Tcplp_IcmpV6Transmit</code> | |
| Syntax: | <pre>Std_ReturnType Tcplp_IcmpV6Transmit(TcpIp_LocalAddrIdType LocalIpAddrId, const TcpIp_SockAddrType* RemoteAddrPtr, uint8 HopLimit, uint8 Type, uint8 Code, uint16 DataLength, const uint8* DataPtr)</pre> | |
| Service ID[hex]: | 0x18 | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | <code>LocalIpAddrId</code> | IP address identifier representing the local IP address and <code>EthIf</code> controller which shall be used for transmission of the ICMPv6 message. |
| | <code>RemoteAddrPtr</code> | pointer to struct representing the remote address |
| | <code>HopLimit</code> | Hop Limit value to be used for the ICMPv6 message. If 0 is specified the default value shall be used. |
| | <code>Type</code> | type field value to be used in the ICMPv6 message. (Note: the value of the type field determines the format of the remaining ICMPv6 message data) |
| | <code>Code</code> | code field value to be used in the ICMPv6 message |
| | <code>DataLength</code> | length of ICMPv6 message |
| | <code>DataPtr</code> | Pointer to data which shall be sent as ICMPv6 message data |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | <code>Std_ReturnType</code> | Result of operation E_OK: The ICMPv6 message has been sent successfully E_NOT_OK: The ICMPv6 message was not sent. |
| Description: | By this API service the TCP/IP stack sends an ICMPv6 message according to the specified parameters. | |

] ()

[SWS_TCPIP_00230] The service `Tcplp_IcmpV6Transmit()` shall (a) construct an ICMPv6 message according to the parameters `Type`, `Code`, `DataLength` and `DataPtr` and (b) transmit the ICMPv6 message using the local IP address and `EthIf` controller specified by `LocalIpAddrId` to the destination specified by `RemoteAddrPtr` using a Hop Limit value according to the parameter `HopLimit`.] ()

8.3.3.6 `Tcplp_DhcpReadOption`

[SWS_TCPIP_00040]

| | |
|----------------------|-----------------------------------|
| Service name: | <code>Tcplp_DhcpReadOption</code> |
|----------------------|-----------------------------------|

| | | |
|----------------------------|--|---|
| Syntax: | <pre>Std_ReturnType TcpIp_DhcpReadOption(TcpIp_LocalAddrIdType LocalIpAddrId, uint8 Option, uint8* DataLength, uint8* DataPtr)</pre> | |
| Service ID[hex]: | 0x0D | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | LocalIpAddrId | IP address identifier representing the local IP address and EthIf controller for which the DHCP option shall be read. |
| | Option | DHCP option according to IETF RFC 2132, e.g. hostname |
| | DataPtr | Pointer to memory containing DHCP option data |
| Parameters (inout): | DataLength | As input parameter, contains the length of the provided data buffer. Will be overwritten with the length of the actual data. |
| Parameters (out): | None | |
| Return value: | Std_ReturnType | Result of operation E_OK requested data retrieved successfully. E_NOT_OK requested data could not be retrieved. |
| Description: | By this API service the TCP/IP stack retrieves DHCP option data identified by parameter option for already received DHCP options. | |

] (SRS_Eth_00066)

[SWS_TCPIP_00233] If development error detection is enabled:
TcpIp_DhcpReadOption() shall check if the parameter LocalIpAddrId is valid. If the check fails, TcpIp_DhcpReadOption() shall raise the development error TCPIP_E_INV_ARG and return E_NOT_OK.] (SRS_Eth_00066)

[SWS_TCPIP_00234] If development error detection is enabled:
TcpIp_DhcpReadOption() shall check if the parameter Option is valid. If the check fails, TcpIp_DhcpReadOption() shall raise the development error TCPIP_E_INV_ARG and return E_NOT_OK.] (SRS_Eth_00066)

[SWS_TCPIP_00235] If development error detection is enabled:
TcpIp_DhcpReadOption() shall check if the parameter DataLength is valid (i.e. the buffer is large enough for the requested option). If the check fails, TcpIp_DhcpReadOption() shall raise the development error TCPIP_E_INV_ARG and return E_NOT_OK.] (SRS_Eth_00066)

[SWS_TCPIP_00236] If the requested option has been set for the address specified by LocalIpAddrId, TcpIp_DhcpReadOption() shall copy this option into the buffer provided by DataPtr, set the parameter DataLength to the length of the option and return E_OK.] (SRS_Eth_00066)

[SWS_TCPIP_00237] If the requested option has not been set for the address specified by LocalIpAddrId, TcpIp_DhcpReadOption() shall set the parameter DataLength to zero, leave the buffer provided by DataPtr unchanged and return E_OK.] (SRS_Eth_00066)

8.3.3.7 Tcplp_DhcpV6ReadOption

[SWS_TCPIP_00189]

| | | |
|----------------------------|--|---|
| Service name: | Tcplp_DhcpV6ReadOption | |
| Syntax: | <pre>Std_ReturnType Tcplp_DhcpV6ReadOption(TcpIp_LocalAddrIdType LocalIpAddrId, uint16 Option, uint16* DataLength, uint8* DataPtr)</pre> | |
| Service ID[hex]: | 0x19 | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | LocalIpAddrId | IP address identifier representing the local IP address and EthIf controller for which the DHCPv6 option shall be read. |
| | Option | DHCP option according to IETF RFC 3315, e.g. hostname |
| | DataPtr | Pointer to memory containing DHCPv6 option data |
| Parameters (inout): | DataLength | As input parameter, contains the length of the provided data buffer. Will be overwritten with the length of the actual data. |
| Parameters (out): | None | |
| Return value: | Std_ReturnType | Result of operation E_OK: requested data retrieved successfully. E_NOT_OK: requested data could not be retrieved. |
| Description: | By this API service the TCP/IP stack retrieves DHCPv6 option data identified by parameter option for already received DHCPv6 options. | |

] (SRS_Eth_00066)

[SWS_TCPIP_00238] If development error detection is enabled:

Tcplp_DhcpV6ReadOption() shall check if the parameter LocalIpAddrId is valid. If the check fails, Tcplp_DhcpV6ReadOption() shall raise the development error TCPIP_E_INV_ARG and return E_NOT_OK.] (SRS_Eth_00066)

[SWS_TCPIP_00239] If development error detection is enabled:

Tcplp_DhcpV6ReadOption() shall check if the parameter Option is valid. If the check fails, Tcplp_DhcpV6ReadOption() shall raise the development error TCPIP_E_INV_ARG and return E_NOT_OK.] (SRS_Eth_00066)

[SWS_TCPIP_00240] If development error detection is enabled:

Tcplp_DhcpV6ReadOption() shall check if the parameter DataLength is valid (i.e. the buffer is large enough for the requested option). If the check fails, Tcplp_DhcpV6ReadOption() shall raise the development error TCPIP_E_INV_ARG and return E_NOT_OK.] (SRS_Eth_00066)

[SWS_TCPIP_00241] If the requested option has been set for the address specified by LocalIpAddrId, Tcplp_DhcpV6ReadOption() shall copy this option into the buffer provided by DataPtr, set the parameter DataLength to the length of the option and return E_OK.] (SRS_Eth_00066)

[SWS_TCPIP_00242] If the requested option has not been set for the address specified by LocalIpAddrId, Tcplp_DhcpV6ReadOption() shall set the parameter

DataLength to zero, leave the buffer provided by DataPtr unchanged and return E_OK.] (SRS_Eth_00066)

8.3.3.8 Tcplp_DhcpWriteOption

[SWS_TCPIP_00020]

| | | |
|----------------------------|--|--|
| Service name: | Tcplp_DhcpWriteOption | |
| Syntax: | <pre>Std_ReturnType Tcplp_DhcpWriteOption(TcpIp_LocalAddrIdType LocalIpAddrId, uint8 Option, uint8 DataLength, const uint8* DataPtr)</pre> | |
| Service ID[hex]: | 0x0E | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | LocalIpAddrId | IP address identifier representing the local IP address and EthIf controller for which the DHCP option shall be written. |
| | Option | DHCP option according to IETF RfC 2132, e.g. hostname |
| | DataLength | length of DHCP option data |
| | DataPtr | Pointer to memory containing DHCP option data |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | Std_ReturnType | Result of operation E_OK no error occurred. E_NOT_OK DHCP option data could not be written. |
| Description: | By this API service the TCP/IP stack writes the DHCP option data identified by parameter option. | |

] (SRS_Eth_00065)

[SWS_TCPIP_00243] If development error detection is enabled:
Tcplp_DhcpWriteOption() shall check if the parameter LocalIpAddrId is valid. If the check fails, Tcplp_DhcpWriteOption() shall raise the development error TCPIP_E_INV_ARG and return E_NOT_OK.] (SRS_Eth_00065)

[SWS_TCPIP_00244] If development error detection is enabled:
Tcplp_DhcpWriteOption() shall check if the parameter Option is valid. If the check fails, Tcplp_DhcpWriteOption() shall raise the development error TCPIP_E_INV_ARG and return E_NOT_OK.] (SRS_Eth_00065)

[SWS_TCPIP_00245] If development error detection is enabled:
Tcplp_DhcpWriteOption() shall check if the parameter DataLength is valid (i.e. the length of the provided option is not larger than supported by the protocol). If the check fails, Tcplp_DhcpWriteOption() shall raise the development error TCPIP_E_INV_ARG and return E_NOT_OK.] (SRS_Eth_00065)

[SWS_TCPIP_00246] If the length indicated by DataLength is larger than zero Tcplp_DhcpWriteOption() shall set the option identified by Option to the value provided by DataPtr internally for the address specified by LocalIpAddrId and return E_OK.] (SRS_Eth_00065)

[SWS_TCPIP_00247] If the length indicated by DataLength is equal to zero Tcplp_DhcpWriteOption() shall unset the option identified by Option for the address specified by LocalIpAddrId and return E_OK.] (SRS_Eth_00065)

8.3.3.9 Tcplp_DhcpV6WriteOption

[SWS_TCPIP_00190]

| | | |
|----------------------------|--|--|
| Service name: | Tcplp_DhcpV6WriteOption | |
| Syntax: | <pre>Std_ReturnType Tcplp_DhcpV6WriteOption(TcpIp_LocalAddrIdType LocalIpAddrId, uint16 Option, uint16 DataLength, const uint8* DataPtr)</pre> | |
| Service ID[hex]: | 0x1a | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | LocalIpAddrId | IP address identifier representing the local IP address and EthIf controller for which the DHCPv6 option shall be written. |
| | Option | DHCP option according to IETF RFC 3315, e.g. hostname |
| | DataLength | length of DHCPv6 option data |
| | DataPtr | Pointer to memory containing DHCPv6 option data |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | Std_ReturnType | Result of operation E_OK: no error occurred. E_NOT_OK: DHCPv6 option data could not be written. |
| Description: | By this API service the TCP/IP stack writes the DHCPv6 option data identified by parameter option. | |

] (SRS_Eth_00065)

[SWS_TCPIP_00248] If development error detection is enabled: Tcplp_DhcpV6WriteOption() shall check if the parameter LocalIpAddrId is valid. If the check fails, Tcplp_DhcpV6WriteOption() shall raise the development error TCPIP_E_INV_ARG and return E_NOT_OK.] (SRS_Eth_00065)

[SWS_TCPIP_00249] If development error detection is enabled: Tcplp_DhcpV6WriteOption() shall check if the parameter Option is valid. If the check fails, Tcplp_DhcpV6WriteOption() shall raise the development error TCPIP_E_INV_ARG and return E_NOT_OK.] (SRS_Eth_00065)

[SWS_TCPIP_00250] If development error detection is enabled: Tcplp_DhcpV6WriteOption() shall check if the parameter DataLength is valid (i.e. the length of the provided option is not larger than supported by the protocol). If the check fails, Tcplp_DhcpV6WriteOption() shall raise the development error TCPIP_E_INV_ARG and return E_NOT_OK.] (SRS_Eth_00065)

[SWS_TCPIP_00251] If the length indicated by DataLength is larger than zero Tcplp_DhcpV6WriteOption() shall set the option identified by Option to the value

provided by DataPtr internally for the address specified by LocalAddrId and return E_OK.] (SRS_Eth_00065)

[SWS_TCPIP_00252] If the length indicated by DataLength is equal to zero TcpIp_DhcpV6WriteOption() shall unset the option identified by Option for the address specified by LocalAddrId and return E_OK.] (SRS_Eth_00065)

8.3.3.10 TcpIp_ChangeParameter

[SWS_TCPIP_00016]

| | | |
|----------------------------|---|---|
| Service name: | TcpIp_ChangeParameter | |
| Syntax: | <pre>Std_ReturnType TcpIp_ChangeParameter (TcpIp_SocketIdType SocketId, TcpIp_ParamIdType ParameterId, const uint8* ParameterValue)</pre> | |
| Service ID[hex]: | 0x0F | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Reentrant for different SocketIds. Non reentrant for the same SocketId. | |
| Parameters (in): | SocketId | Socket identifier of the related local socket resource. |
| | ParameterId | Identifier of the parameter to be changed |
| | ParameterValue | Pointer to memory containing the new parameter value |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | Std_ReturnType | E_OK: The parameter has been changed successfully. E_NOT_OK: The parameter could not be changed. |
| Description: | By this API service the TCP/IP stack is requested to change a parameter of a socket. E.g. the Nagle algorithm may be controlled by this API. | |

] ()

[SWS_TCPIP_00119] The service TcpIp_ChangeParameter() shall change the parameter specified by ParameterId with the value specified by ParameterValue of the socket specified by SocketId.] ()

8.3.3.11 TcpIp_GetIpAddr

[SWS_TCPIP_00032]

| | | |
|-------------------------|---|---|
| Service name: | TcpIp_GetIpAddr | |
| Syntax: | <pre>Std_ReturnType TcpIp_GetIpAddr (TcpIp_LocalAddrIdType LocalAddrId, TcpIp_SockAddrType* IpAddrPtr, uint8* NetmaskPtr, TcpIp_SockAddrType* DefaultRouterPtr)</pre> | |
| Service ID[hex]: | 0x10 | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Reentrant | |
| Parameters (in): | LocalAddrId | Local address identifier referring to the local IP address which shall be obtained. |
| | IpAddrPtr | Pointer to a struct where the IP address shall be stored. The struct member domain shall be set to the desired |

| | | |
|--------------------------|---|---|
| | | Tcplp_DomainType and it shall be ensured that the struct is large enough to store an address of the selected type (INET or INET6). Struct members not related to the IP address are of arbitrary value and shall not be used. |
| | DefaultRouterPtr | Pointer to struct where the IP address of the default router (gateway) is stored (struct member "port" is not used and of arbitrary value). The struct must be of the same type and size as IpAddrPtr. |
| Parameters (out): | NetmaskPtr | Pointer to memory where Network mask of IPv4 address or address prefix of IPv6 address in CIDR Notation is stored |
| Return value: | Std_ReturnType | Result of operation E_OK: The request was successful E_NOT_OK: The request was not successful, e.g. domain in IpAddrPtr and the local domain type do not match |
| Description: | Obtains the local IP address actually used by LocalAddrId, the netmask and default router | |

] ()

[SWS_TCPIP_00205] Tcplp_GetIpAddr() shall refuse the request if the domain set in IpAddrPtr does not match the Tcplp_DomainType of the selected local address and return E_NOT_OK. If development error detection is enabled, the service Tcplp_GetIpAddr() shall also raise the development error TCPIP_E_INV_ARG.] ()

[SWS_TCPIP_00206] Tcplp_GetIpAddr() shall refuse the request if the domain set in IpAddrPtr does not match the domain set in DefaultRouterPtr and return E_NOT_OK. If development error detection is enabled, the service Tcplp_GetIpAddr() shall also raise the development error TCPIP_E_INV_ARG.] ()

8.3.3.12 Tcplp_GetPhysAddr

[SWS_TCPIP_00033]

| | | |
|----------------------------|--|--|
| Service name: | Tcplp_GetPhysAddr | |
| Syntax: | Std_ReturnType Tcplp_GetPhysAddr(Tcplp_LocalAddrIdType LocalAddrId, uint8* PhysAddrPtr) | |
| Service ID[hex]: | 0x11 | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | LocalAddrId | Local address identifier implicitly specifying the EthIf controller for which the physical address shall be obtained. |
| Parameters (inout): | None | |
| Parameters (out): | PhysAddrPtr | Pointer to the memory where the physical source address (MAC address) in network byte order is stored |
| Return value: | Std_ReturnType | Result of operation E_OK The request was successful E_NOT_OK The request was not successful, e.g. no unique Ctrl specified via IpAddrId. |
| Description: | Obtains the physical source address used by the EthIf controller implicitly specified via LocalAddrId. | |

] ()

8.3.3.13 Tcplp_GetRemotePhysAddr

[SWS_TCPIP_00137]

| | | |
|----------------------------|--|--|
| Service name: | Tcplp_GetRemotePhysAddr | |
| Syntax: | <pre>TcpIp_ReturnType Tcplp_GetRemotePhysAddr (uint8 CtrlIdx, const TcpIp_SockAddrType* IpAddrPtr, TcpIp_ReturnType PhysAddrPtr, boolean initRes)</pre> | |
| Service ID[hex]: | 0x16 | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | CtrlIdx | EthIf controller index to identify the related ARP/NDP table. |
| | IpAddrPtr | specifies the IP address for which the physical address shall be retrieved |
| | initRes | specifies if the address resolution shall be initiated (TRUE) or not (FALSE) in case the physical address related to the specified IP address is currently unknown. |
| Parameters (inout): | None | |
| Parameters (out): | PhysAddrPtr | Pointer to the memory where the physical address (MAC address) related to the specified IP address is stored in network byte order. |
| Return value: | Tcplp_ReturnType | TCPIP_E_OK: specified IP address resolved, physical address provided via PhysAddrPtr TCPIP_E_PHYS_ADDR_MISS: physical address currently unknown (address resolution initiated if initRes set to TRUE) |
| Description: | Tcplp_GetRemotePhysAddr queries the IP/physical address translation table specified by CtrlIdx and returns the physical address related to the IP address specified by IpAddrPtr. In case no physical address can be retrieved and parameter initRes is TRUE, address resolution for the specified IP address is initiated on the local network. | |

()

[SWS_TCPIP_00138] Tcplp_GetRemotePhysAddr shall lookup the physical address for the IP address specified by IpAddrPtr at the IP/physical address translation table related to the controller identified by CtrlIdx.

(1) If the physical address is already known, PhysAddrPtr shall be set to the related physical address and the function shall return with TCPIP_E_OK.

(2) Otherwise it shall (a) initiate an address resolution if parameter initRes is set to TRUE and (b) return with TCPIP_E_PHYS_ADDR_MISS. PhysAddrPtr is not updated in this case.] ()

[SWS_TCPIP_00139] Tcplp_GetRemotePhysAddr shall immediately return with TCPIP_E_NOT_OK if it is called with an IP address that is not part of the same sub network as the local address currently assigned to the controller identified by CtrlIdx.

] ()

8.3.3.14 Tcplp_GetCtrlIdx

[SWS_TCPIP_00140]

| | |
|----------------------|-----------------------------------|
| Service name: | Tcplp_GetCtrlIdx |
| Syntax: | Std_ReturnType Tcplp_GetCtrlIdx (|

| | | |
|----------------------------|--|--|
| | TcpIp_LocalAddrIdType LocalAddrId, uint8* CtrlIdxPtr) | |
| Service ID[hex]: | 0x17 | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Reentrant | |
| Parameters (in): | LocalAddrId | Local address identifier implicately specifying the EthIf controller that shall be returned. |
| Parameters (inout): | None | |
| Parameters (out): | CtrlIdxPtr | Pointer to the memory where the index of the controller related to LocalAddrId is stored |
| Return value: | Std_ReturnType | Result of operation E_OK the request was successful E_NOT_OK the request was not successful. |
| Description: | Tcplp_GetCtrlIdx returns the index of the controller related to LocalAddrId. | |

] ()

[SWS_TCPIP_00141][[Tcplp_GetCtrlIdx shall return the index of the controller related to LocalAddrId.] ()

8.3.4 Transmission

8.3.4.1 Tcplp_UdpTransmit

[SWS_TCPIP_00025][

| | | |
|----------------------------|---|---|
| Service name: | Tcplp_UdpTransmit | |
| Syntax: | Std_ReturnType Tcplp_UdpTransmit(TcpIp_SocketIdType SocketId, const uint8* DataPtr, const TcpIp_SockAddrType* RemoteAddrPtr, uint16 TotalLength) | |
| Service ID[hex]: | 0x12 | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Reentrant for different SocketIds. Non reentrant for the same SocketId. | |
| Parameters (in): | SocketId | Socket identifier of the related local socket resource. |
| | DataPtr | Pointer to a linear buffer of TotalLength bytes containing the data to be transmitted. In case DataPtr is a NULL_PTR, Tcplp shall retrieve data from upper layer via callback <Up>_CopyTxData(). |
| | RemoteAddrPtr | IP address and port of the remote host to transmit to. |
| | TotalLength | indicates the payload size of the UDP datagram. |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | Std_ReturnType | E_OK: UDP message has been forwarded to EthIf for transmission. E_NOT_OK: UDP message could not be sent because of a permanent error, e.g. message is too long. |
| Description: | This service transmits data via UDP to a remote node. The transmission of the data is immediately performed with this function call by forwarding it to EthIf. | |

] ()

[SWS_TCPIP_00120] The service `TcpIp_UdpTransmit()` shall immediately transmit `TotalLength` data bytes via UDP and the socket specified by `SocketId` to a remote socket specified by `RemoteAddrPtr` according to the sequence diagram specified in section 9.5.] ()

[SWS_TCPIP_00121] `DataPtr` shall either point to a linear buffer of `TotalLength` bytes containing the data for transmission or be a `NULL_PTR`. For data transmission the service `TcpIp_UdpTransmit()` shall either use all data from the linear buffer if `DataPtr` is not a `NULL_PTR`, or retrieve `TotalLength` data bytes from the upper layer by calling `Up_CopyTxData()` one or multiple times in the context of this service otherwise.] ()

[SWS_TCPIP_00122] The service `TcpIp_UdpTransmit()` shall select the local IP address and port for transmission if the socket specified by `SocketId` has not been bound to a local resource via a previous call to `TcpIp_Bind()`.] ()

8.3.4.2 Tcplp_TcpTransmit

[SWS_TCPIP_00050]

| | | |
|----------------------------|---|--|
| Service name: | Tcplp_TcpTransmit | |
| Syntax: | <pre>Std_ReturnType TcpIp_TcpTransmit(TcpIp_SocketIdType SocketId, const uint8* DataPtr, uint32 AvailableLength, boolean ForceRetrieve)</pre> | |
| Service ID[hex]: | 0x13 | |
| Sync/Async: | Asynchronous | |
| Reentrancy: | Reentrant for different SocketIds. Non reentrant for the same SocketId. | |
| Parameters (in): | SocketId | Socket identifier of the related local socket resource. |
| | DataPtr | Pointer to a linear buffer of <code>AvailableLength</code> bytes containing the data to be transmitted. In case <code>DataPtr</code> is a <code>NULL_PTR</code> , <code>Tcplp</code> shall retrieve data from upper layer via callback <code><Up>_CopyTxData()</code> . |
| | AvailableLength | Available data for transmission in bytes. |
| | ForceRetrieve | This parameter is only valid if <code>DataPtr</code> is a <code>NULL_PTR</code> . Indicates how the TCP/IP stack retrieves data from upper layer if <code>DataPtr</code> is a <code>NULL_PTR</code> . TRUE: the whole data indicated by <code>availableLength</code> shall be retrieved from the upper layer via one or multiple <code><Up>_CopyTxData()</code> calls within the context of this transmit function. FALSE: The TCP/IP stack may retrieve up to <code>availableLength</code> data from the upper layer. It is allowed to retrieve less than <code>availableLength</code> bytes. Note: Not retrieved data will be provided by upper layer with the next call to <code>Tcplp_TcpTransmit</code> (along with new data if available). |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | Std_ReturnType | E_OK: The request has been accepted E_NOT_OK: The request has not been accepted, e.g. due to a |

| | |
|---------------------|--|
| | lack of buffer space or the socket is not connected. |
| Description: | This service requests transmission of data via TCP to a remote node. The transmission of the data is decoupled. Note: The TCP segment(s) are sent dependent on runtime factors (e.g. receive window) and configuration parameter (e.g. Nagle algorithm) . |

] ()

[SWS_TCPIP_00123] The service `TcpIp_TcpTransmit()` shall transmit data via TCP and the socket specified by `SocketId` to the connected remote socket according to the sequence diagram specified in section 9.4.] ()

[SWS_TCPIP_00124] `DataPtr` shall either point to a linear buffer of `AvailableLength` bytes containing the data for transmission or be a `NULL_PTR`. For data transmission the service `TcpIp_TcpTransmit()` shall either use all data from the linear buffer if `DataPtr` is not a `NULL_PTR`, or retrieve up to `AvailableLength` data bytes from the upper layer by calling `Up_CopyTxData()` one or multiple times in the context of this service otherwise.] ()

[SWS_TCPIP_00125] The service `TcpIp_TcpTransmit()` shall retrieve exactly `AvailableLength` bytes from the upper layer if the parameter `DataPtr` is a `NULL_PTR` and `ForceRetrieve` is `TRUE`. (If `DataPtr` is a `NULL_PTR` and `ForceRetrieve` is `FALSE`, `TcpIp` may retrieve less data than available).] ()

Note: The TCP segment(s) are sent dependent on runtime factors (e.g. receive window) and configuration parameter (e.g. Nagle algorithm).

8.4 Call-back notifications

This is a list of functions provided for other modules. The function prototypes of the callback functions shall be provided in the file `Tcplp_Cbk.h`.

8.4.1 Tcplp_RxIndication

[SWS_TCPIP_00029]

| | | |
|-------------------------|---|---|
| Service name: | Tcplp_RxIndication | |
| Syntax: | <pre>void TcpIp_RxIndication(uint8 CtrlIdx, Eth_FrameType FrameType, boolean IsBroadcast, const uint8* PhysAddrPtr, uint8* DataPtr, uint16 LenByte)</pre> | |
| Service ID[hex]: | 0x14 | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | <code>CtrlIdx</code> | Index of the EthIf controller. |
| | <code>FrameType</code> | frame type of received Ethernet frame |
| | <code>IsBroadcast</code> | parameter to indicate a broadcast frame |

| | | |
|----------------------------|---|---|
| | PhysAddrPtr | pointer to Physical source address (MAC address in network byte order) of received Ethernet frame |
| | DataPtr | Pointer to payload of the received Ethernet frame (i.e. Ethernet header is not provided). |
| | LenByte | Length of received data. |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | None | |
| Description: | By this API service the TCP/IP stack gets an indication and the data of a received frame. | |

] ()

8.5 Scheduled functions

These functions are directly called by Basic Software Scheduler. The following functions shall have no return value and no parameter. All functions shall be non reentrant.

8.5.1 Terms and definitions

For details refer to the chapter 8.5 “Scheduled functions” in *SWS_BSWGeneral*.

8.5.2 Tcplp_MainFunction

[SWS_TCPIP_00026]

| | |
|-------------------------|--|
| Service name: | Tcplp_MainFunction |
| Syntax: | void TcpIp_MainFunction(void) |
| Service ID[hex]: | 0x15 |
| Description: | Schedules the TCP/IP stack. (Entry point for scheduling) |

] ()

8.6 Expected Interfaces

In this chapter all interfaces required from other modules are listed.

8.6.1 Mandatory Interfaces

This chapter defines all interfaces which are required to fulfill the core functionality of the module.

[SWS_TCPIP_00027]

| API function | Description |
|-----------------------|--|
| Dem_ReportErrorStatus | Queues the reported events from the BSW modules (API is only used by BSW modules). The interface has an asynchronous behavior, |

| | |
|----------------------------|--|
| | because the processing of the event is done within the Dem main function. OBD Events Suppression shall be ignored for this computation. |
| EthIf_GetPhysAddr | Obtains the physical source address used by the indexed controller |
| EthIf_ProvideTxBuffer | Provides access to a transmit buffer of the specified Ethernet controller. |
| EthIf_SetPhysAddr | Sets the physical source address used by the indexed controller. |
| EthIf_Transmit | Triggers transmission of a previously filled transmit buffer |
| EthSM_TcplpModelIndication | This service is called by the Tcplp to report the actual Tcplp state (e.g. online, offline). |

] ()

8.6.2 Optional Interfaces

This chapter defines all interfaces which are required to fulfill an optional functionality of the module.

[SWS_TCPIP_00028]

| API function | Description |
|----------------------------|---|
| Det_ReportError | Service to report development errors. |
| EthIf_UpdatePhysAddrFilter | Update the physical source address to/from the indexed controller filter. If the Ethernet Controller is not capable to do the filtering, the software has to do this. |

] ()

8.6.3 Configurable interfaces

In this chapter all interfaces are listed where the target function could be configured. The target function is usually a call-back function. The names of these kind of interfaces is not fixed because they are configurable.

The ServiceID of the functions defined in this chapter are specified at the upper layer module implementing the functions.

8.6.3.1 Tcplp_<Up>GetSocket

[SWS_TCPIP_00018]

| | | |
|----------------------------|---|--|
| Service name: | Tcplp_<Up>GetSocket | |
| Syntax: | Std_ReturnType Tcplp_<Up>GetSocket(TcpIp_DomainType Domain, TcpIp_ProtocolType Protocol, TcpIp_SocketIdType* SocketIdPtr) | |
| Service ID[hex]: | 0x03 | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Reentrant | |
| Parameters (in): | Domain | IP address family. |
| | Protocol | Socket protocol as sub-family of parameter type. |
| Parameters (inout): | None | |

| | | |
|--------------------------|--|--|
| Parameters (out): | SocketIdPtr | Pointer to socket identifier representing the requested socket. This socket identifier must be provided for all further API calls which requires a SocketId. Note: SocketIdPtr is only valid if return value is E_OK. |
| Return value: | Std_ReturnType | Result of operation E_OK The request has been accepted E_NOT_OK The request has not been accepted: no free socket |
| Description: | By this API service the TCP/IP stack is requested to allocate a new socket. Note: Each accepted incoming TCP connection also allocates a socket resource. | |

] (SRS_Eth_00103)

[SWS_TCPIP_00128] If development error detection is enabled, the service `TcpIp_<Up>GetSocket()` shall check the parameter `Domain` for being valid and raise the development error `TCPIP_E_AFNOSUPPORT` if it is invalid.] ()

[SWS_TCPIP_00222] For each configured `TcpIpSocketOwner` `TcpIp` shall provide a separate `TcpIp_<Up>GetSocket` API by replacing the tag `<Up>` with the short name of the `TcpIpSocketOwner` container. Sockets allocated by a dedicated `TcpIp_<Up>GetSocket` API shall be assigned exclusively to the respective upper layer.] (SRS_Eth_00103)

8.6.3.2 <Up_PhysAddrTableChg>

[SWS_TCPIP_00143]

| | | |
|----------------------------|--|---|
| Service name: | <Up_PhysAddrTableChg> | |
| Syntax: | <pre>void <Up_PhysAddrTableChg>(uint8 CtrlIdx, const TcpIp_SockAddrType* IpAddrPtr, const uint8* PhysAddrPtr, boolean valid)</pre> | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | CtrlIdx | EthIf controller index of the related ARP/NDP table. |
| | IpAddrPtr | specifies the IP address of the changed ARP/NDP table entry |
| | PhysAddrPtr | specifies the physical address of the changed ARP/NDP table entry |
| | valid | specifies if the ARP/NDP table entry is added or changed (TRUE) or has been removed (FALSE) |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | None | |
| Description: | This API is called by <code>TcpIp</code> in case of a change in the ARP/NDP table related to the controller specified by <code>CtrlIdx</code> . | |

()

8.6.3.3 SocketOwner functions

[SWS_TCPIP_00220] For sockets related to a TcpIpSocketOwner with TcpIpSocketOwnerUpperLayerType set to 'SOAD', TcpIp shall replace the tag <Up> with 'SoAd' for each of the following configurable interfaces.] (SRS_Eth_00103)

[SWS_TCPIP_00221] For sockets related to a TcpIpSocketOwner with TcpIpSocketOwnerUpperLayerType set to 'CDD', TcpIp shall use the configured API names for each of the following configurable interfaces.] (SRS_Eth_00103)

8.6.3.3.1 <Up_RxIndication>

[SWS_TCPIP_00223]

| | | |
|----------------------------|--|--|
| Service name: | <Up_RxIndication> | |
| Syntax: | <pre>void <Up_RxIndication>(TcpIp_SocketIdType SocketId, const TcpIp_SockAddrType* RemoteAddrPtr, uint8* BufPtr, uint16 Length)</pre> | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Reentrant for different SocketIds. Non reentrant for the same SocketId. | |
| Parameters (in): | SocketId | Socket identifier of the related local socket resource. |
| | RemoteAddrPtr | Pointer to memory containing IP address and port of the remote host which sent the data. |
| | BufPtr | Pointer to the received data. |
| | Length | Data length of the received TCP segment or UDP datagram. |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | None | |
| Description: | The TCP/IP stack calls this primitive after the reception of data on a socket. The socket identifier along with configuration information determines which module is to be called. | |

] (SRS_Eth_00103)

8.6.3.3.2 <Up_TcpIpEvent>

[SWS_TCPIP_00224]

| | | |
|----------------------------|---|--|
| Service name: | <Up_TcpIpEvent> | |
| Syntax: | <pre>void <Up_TcpIpEvent>(TcpIp_SocketIdType SocketId, TcpIp_EventType Event)</pre> | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | SocketId | Socket identifier of the related local socket resource. |
| | Event | This parameter contains a description of the event just encountered. |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | None | |
| Description: | This service gets called if the stack encounters a condition described by the | |

| | |
|--|------------------|
| | values in Event. |
|--|------------------|

] (SRS_Eth_00103)

8.6.3.3.3 <Up_TxConfirmation>

[SWS_TCPIP_00225]

| | | |
|----------------------------|--|---|
| Service name: | <Up_TxConfirmation> | |
| Syntax: | <pre>void <Up_TxConfirmation>(TcpIp_SocketIdType SocketId, uint16 Length)</pre> | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Reentrant for different SocketIds. Non reentrant for the same SocketId. | |
| Parameters (in): | SocketId | Socket identifier of the related local socket resource. |
| | Length | Number of transmitted data bytes. |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | None | |
| Description: | <p>The TCP/IP stack calls this function after the data has been acknowledged by the peer for TCP.</p> <p>Caveats: The upper layer might not be able to determine exactly which data bytes have been confirmed.</p> | |

] (SRS_Eth_00103)

8.6.3.3.4 <Up_TcpAccepted>

[SWS_TCPIP_00226]

| | | |
|----------------------------|---|--|
| Service name: | <Up_TcpAccepted> | |
| Syntax: | <pre>Std_ReturnType <Up_TcpAccepted>(TcpIp_SocketIdType SocketId, TcpIp_SocketIdType SocketIdConnected, const TcpIp_SockAddrType* RemoteAddrPtr)</pre> | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | SocketId | Socket identifier of the related local socket resource which has been used at Tcplp_Bind() |
| | SocketIdConnected | Socket identifier of the local socket resource used for the established connection. |
| | RemoteAddrPtr | IP address and port of the remote host. |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | Std_ReturnType | Result of operation E_OK upper layer accepts the established connection E_NOT_OK upper layer refuses the established connection, Tcplp stack shall close the connection. |
| Description: | <p>This service gets called if the stack put a socket into the listen mode before (as server) and a peer connected to it (as client).</p> <p>In detail: The TCP/IP stack calls this function after a socket was set into the listen state with Tcplp_TcpListen() and a TCP connection is requested by the peer.</p> | |

] (SRS_Eth_00103)

8.6.3.3.5 <Up_TcpConnected> [SWS_TCPIP_00227]

| | | |
|----------------------------|--|---|
| Service name: | <Up_TcpConnected> | |
| Syntax: | void <Up_TcpConnected>(TcpIp_SocketIdType SocketId) | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | SocketId | Socket identifier of the related local socket resource. |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | None | |
| Description: | This service gets called if the stack initiated a TCP connection before (as client) and the peer (the server) acknowledged the connection set up. In detail: The TCP/IP stack calls this function after a socket was requested to connect with Tcplp_TcpConnect() and a TCP connection is confirmed by the peer. The parameter value of SocketId equals the SocketId value of the preceding Tcplp_TcpConnect() call. | |

] (SRS_Eth_00103)

8.6.3.3.6 <Up_CopyTxData> [SWS_TCPIP_00228]

| | | |
|----------------------------|---|---|
| Service name: | <Up_CopyTxData> | |
| Syntax: | BufReq_ReturnType <Up_CopyTxData>(TcpIp_SocketIdType SocketId, uint8* BufPtr, uint16 BufLength) | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Reentrant for different SocketIds. Non reentrant for the same SocketId. | |
| Parameters (in): | SocketId | Socket identifier of the related local socket resource. |
| | BufPtr | Pointer to buffer for transmission data. |
| | BufLength | Length of provided data buffer. |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | BufReq_ReturnType | BUFREQ_OK: Data has been copied to the transmit buffer completely as requested. BUFREQ_E_NOT_OK: Data has not been copied. Request failed. (No further action for Tcplp required. Later the upper layer might either close the socket or retry the transmit request) |
| Description: | This service requests to copy data for transmission to the buffer indicated. This call is triggered by Tcplp_Transmit(). Note: The call to <Up>_CopyTxData() may happen in the context of Tcplp_Transmit(). | |

] (SRS_Eth_00103)

8.6.3.3.7 <Up_LocallpAddrAssignmentChg> [SWS_TCPIP_00229]

| | |
|----------------------|-------------------------------|
| Service name: | <Up_LocallpAddrAssignmentChg> |
|----------------------|-------------------------------|

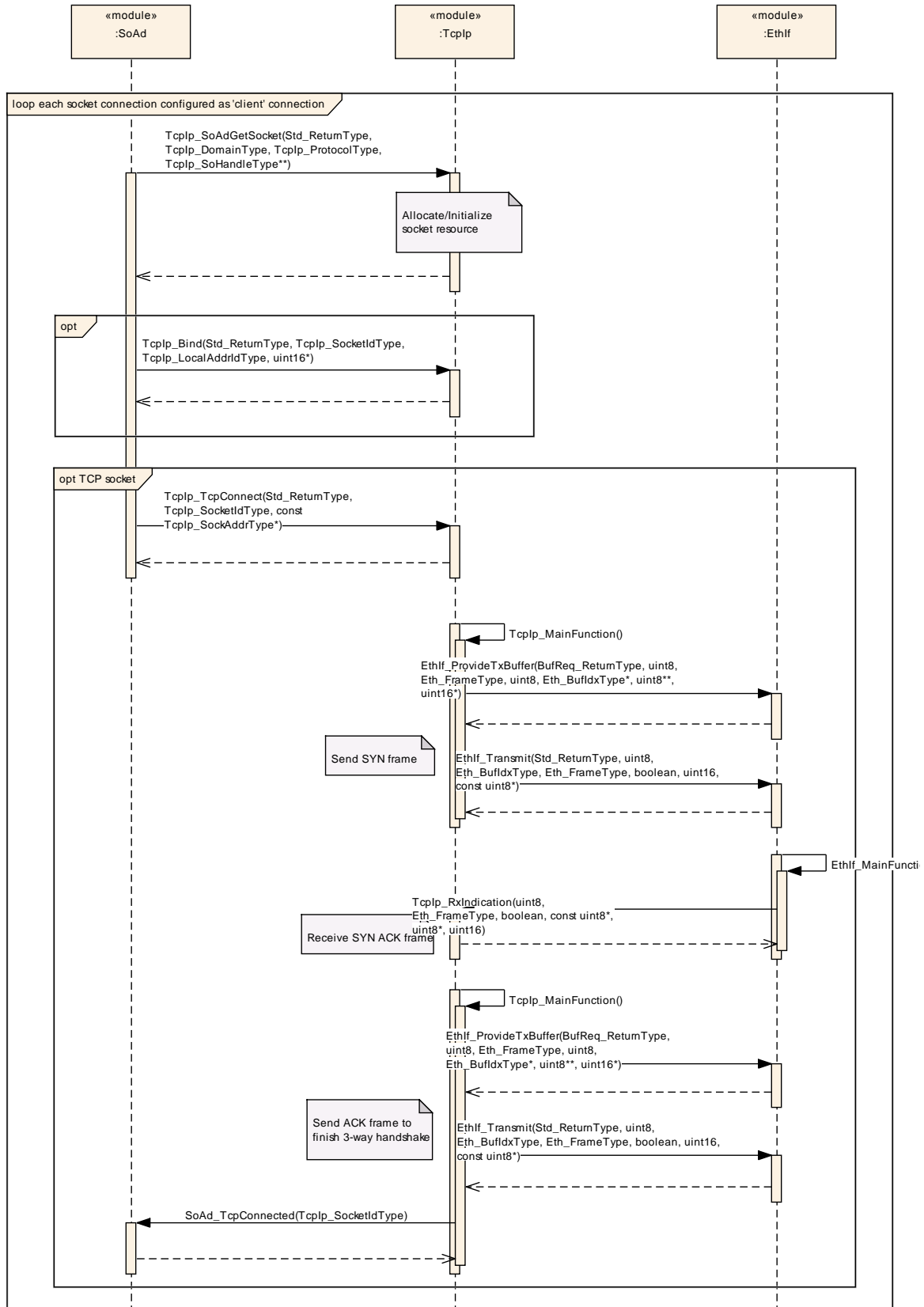
| | | |
|----------------------------|---|---|
| Syntax: | <pre>void <Up_LocalIpAddrAssignmentChg>(TcpIp_LocalAddrIdType IpAddrId, TcpIp_IpAddrStateType State)</pre> | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | IpAddrId | IP address Identifier, representing an IP address specified in the Tcplp module configuraiton (e.g. static IPv4 address on EthIf controller 0). |
| | State | state of IP address assignment |
| Parameters (inout): | None | |
| Parameters (out): | None | |
| Return value: | None | |
| Description: | This service gets called by the TCP/IP stack if an IP address assignment changes (i.e. new address assigned or assigned address becomes invalid). | |

] (SRS_Eth_00103)

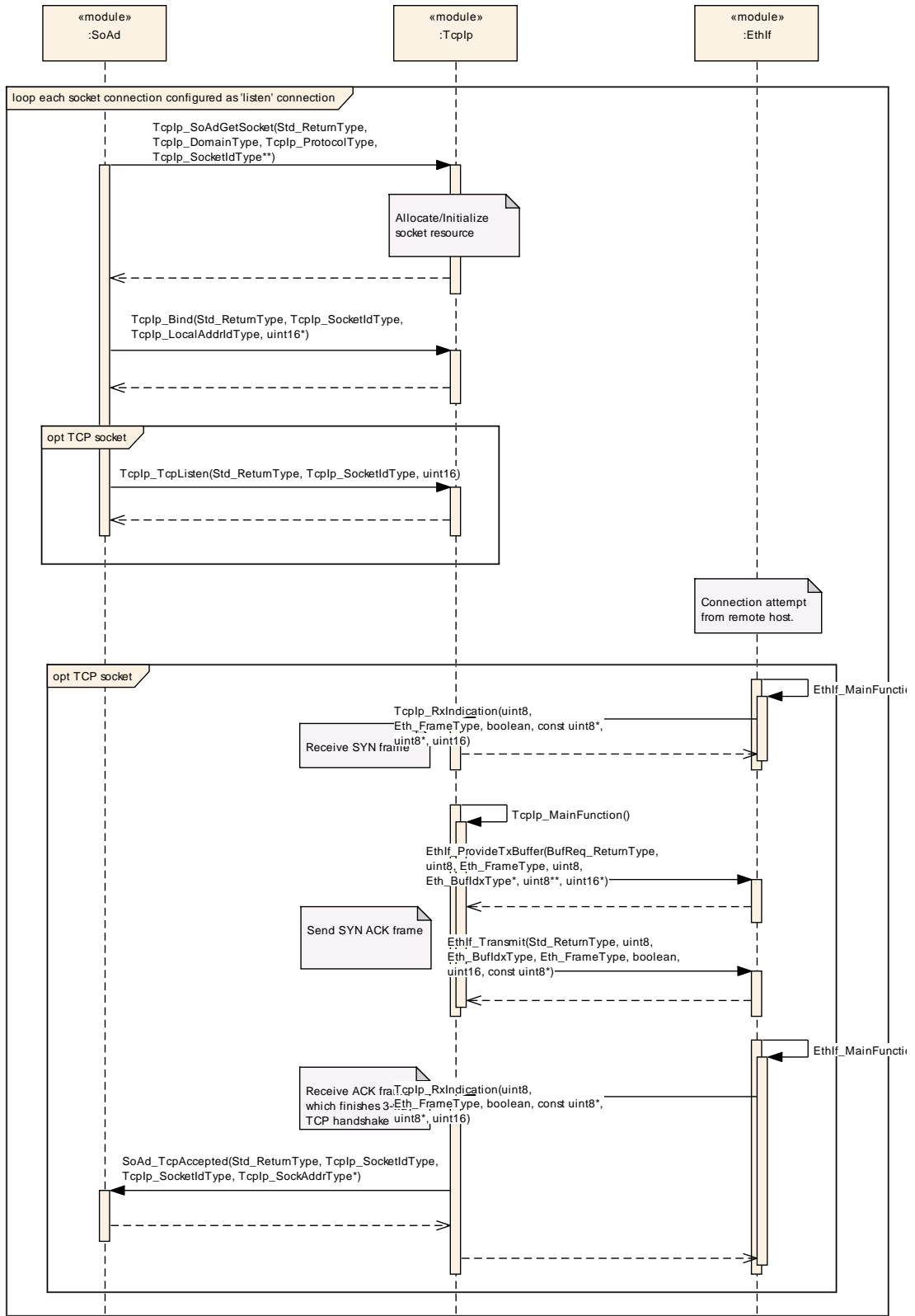
9 Sequence diagrams

Note: The following sequence charts showcase SoAd as upper layer of TcpIp. They shall be understood as example for any other configurable upper layer module.

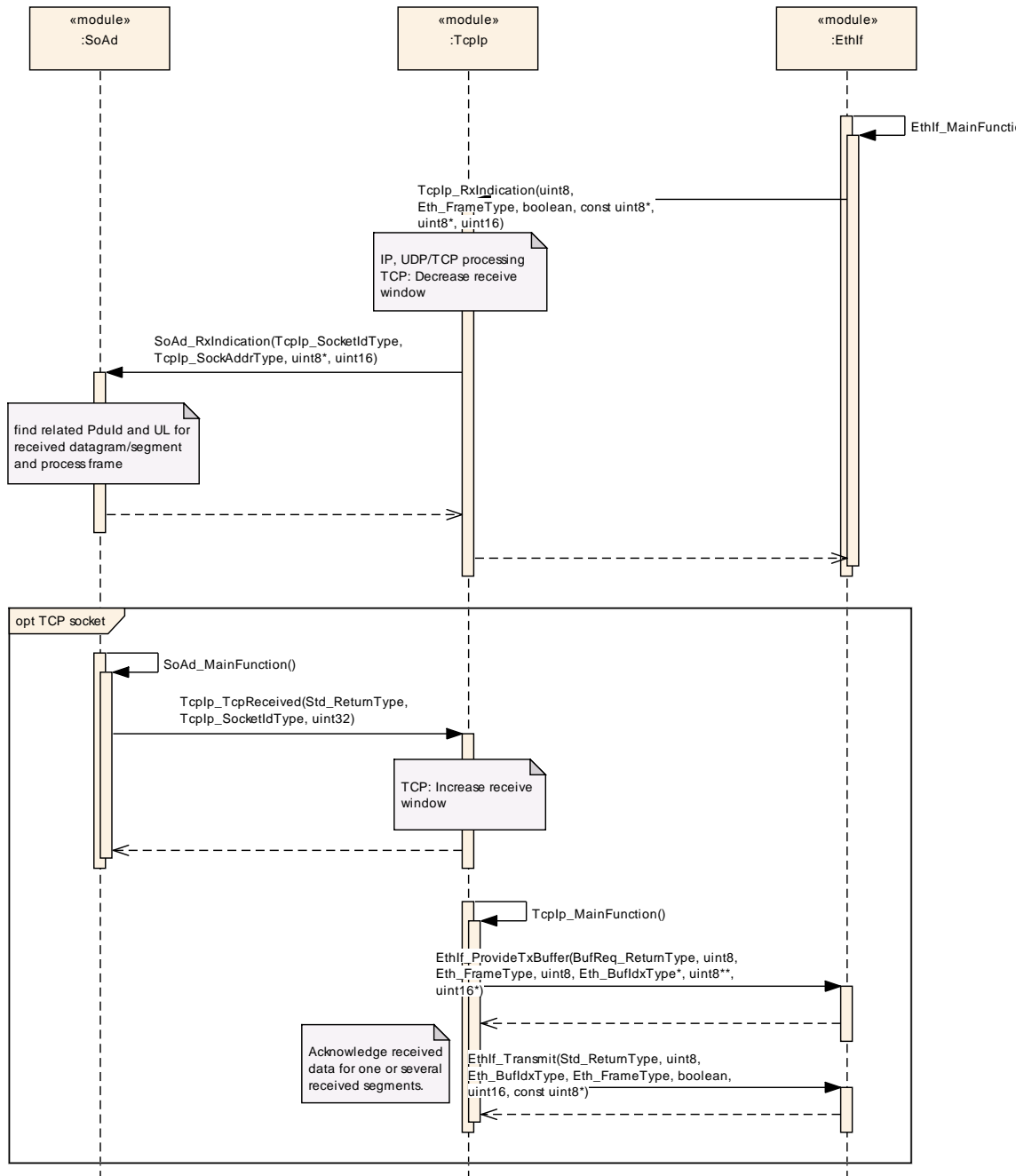
9.1 TCP Connection Setup – Client



9.2 TCP Connection Setup – Server

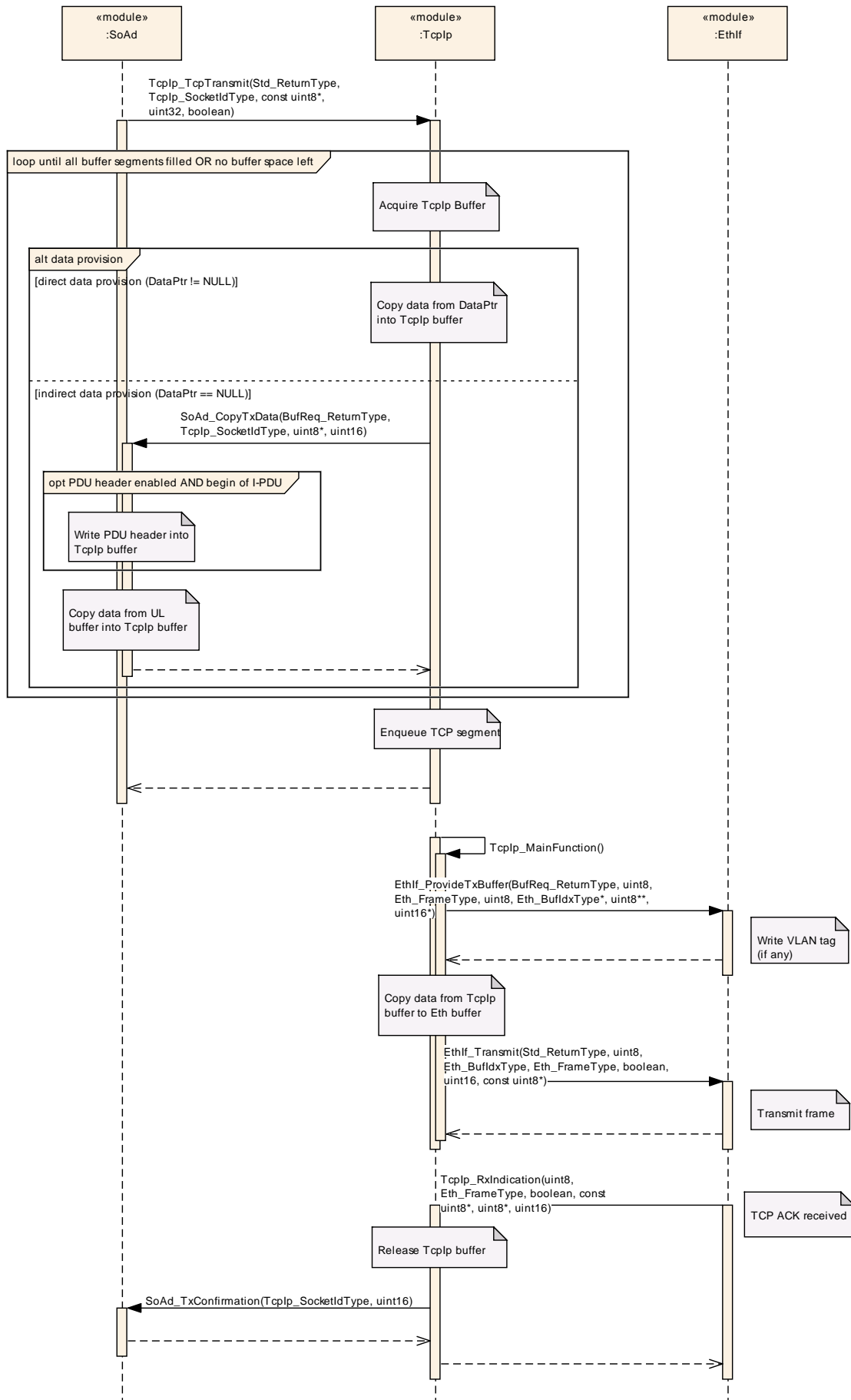


9.3 Reception

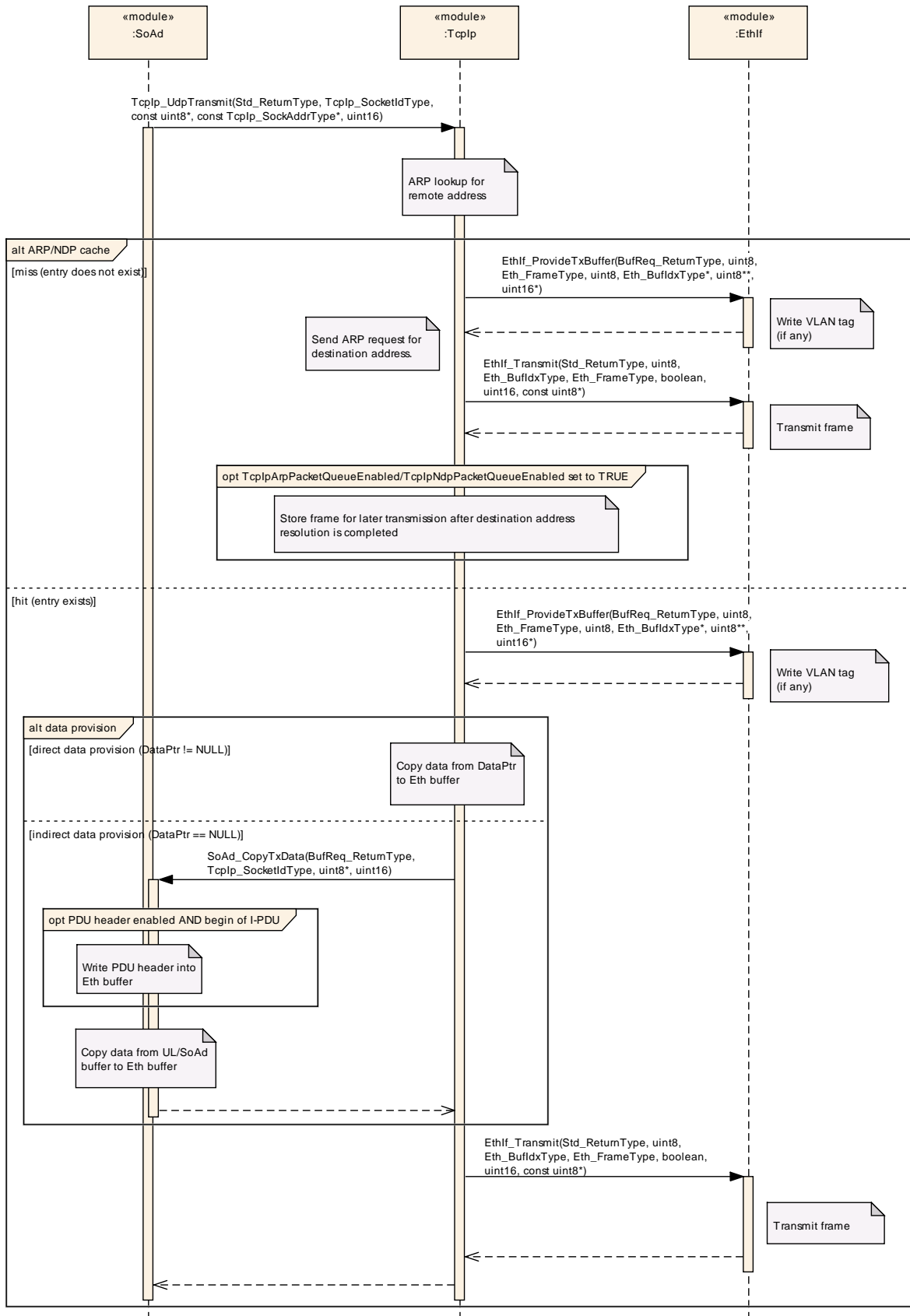


Note: Even it is not shown in the sequence diagram of section 9.3, TcpIp may decouple the data reception if required. E.g. for reassembling of incoming IP datagrams that are fragmented, TcpIp shall copy the received data to a TcpIp buffer and decouple TcpIp_RxIndication() from SoAd_RxIndication().

9.4 Transmission TCP



9.5 Transmission UDP



10 Configuration specification

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

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

Chapter 10.3 specifies published information of the module Tcplp.

10.1 How to read this chapter

For details refer to the chapter 10.1 “Introduction to configuration specification” in *SWS_BSWGeneral*.

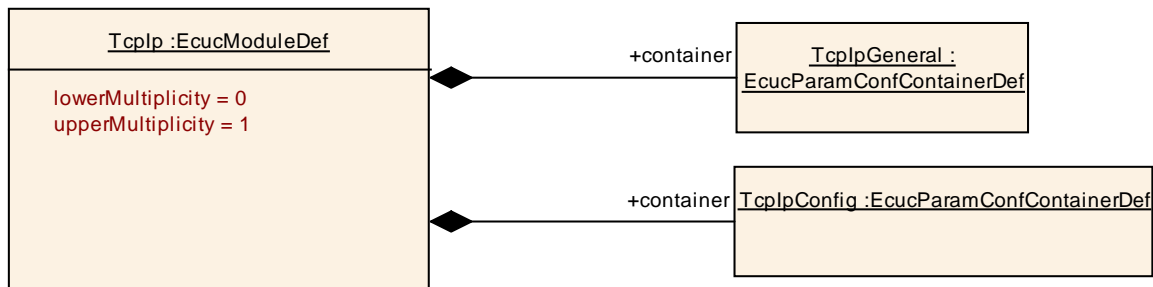
10.2 Containers and configuration parameters

The following chapters summarize all configuration parameters. The detailed meanings of the parameters describe Chapters 7 and Chapter 8.

10.2.1 Variants

VARIANT-PRE-COMPILE only supports pre-compile time configurable parameters.
VARIANT-LINK-TIME includes mainly link-time and some pre-compile configurable parameters

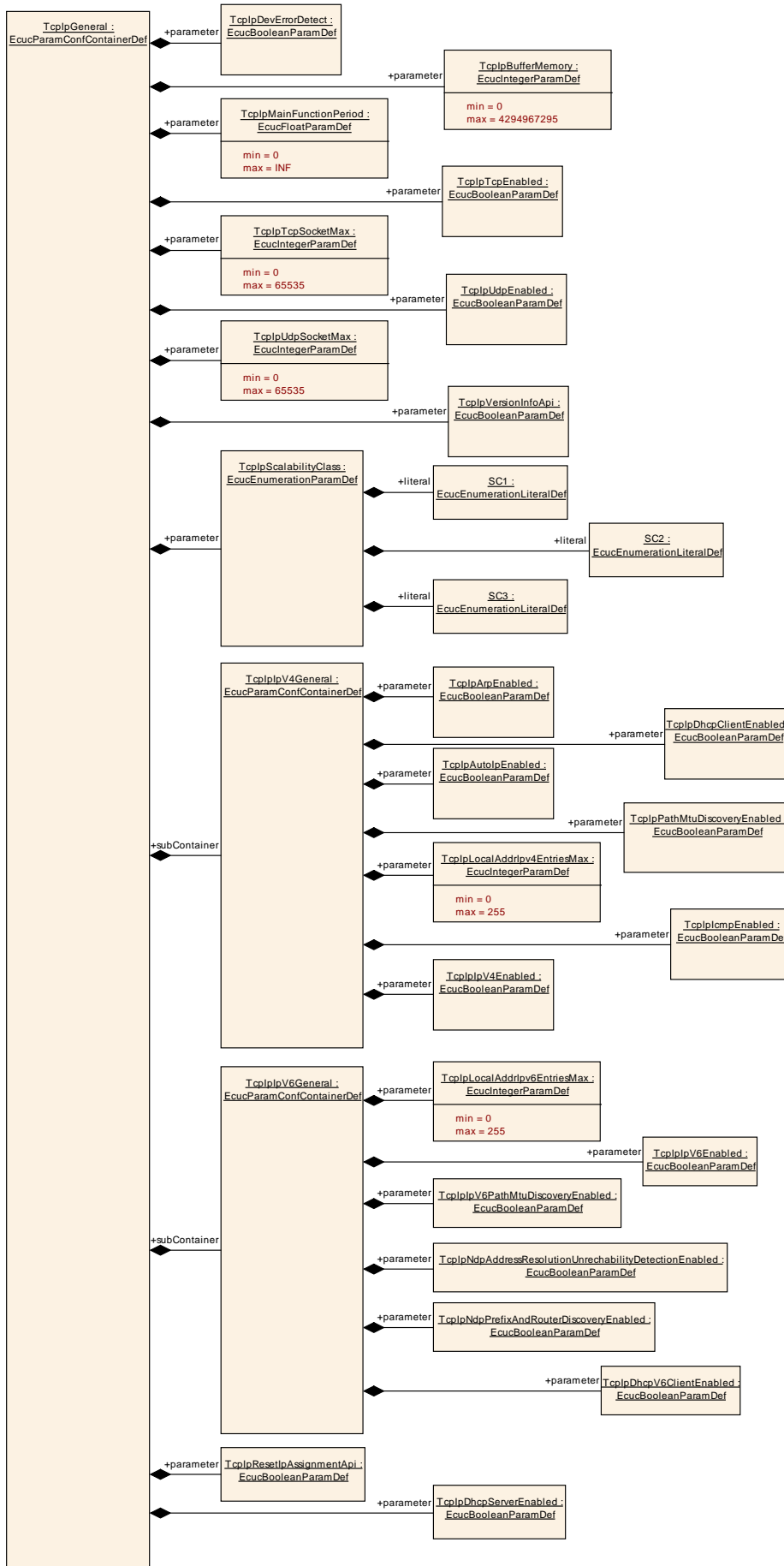
VARIANT-POST-BUILD includes post-build-time, link-time and some pre-compile time configurable parameters.



10.2.2 Tcplp

| | |
|-----------------------------------|---|
| SWS Item | ECUC_Tcplp_00001 : |
| Module Name | <i>Tcplp</i> |
| Module Description | Configuration of the Tcplp (TCP/IP stack) module. |
| Post-Build Variant Support | true |

| Included Containers | | |
|----------------------------|---------------------|---|
| Container Name | Multiplicity | Scope / Dependency |
| TcplpConfig | 1 | This container contains the configuration parameters and sub containers of the AUTOSAR Tcplp module. |
| TcplpGeneral | 1 | This container is a subcontainer of Tcplp and specifies the general configuration parameters of the TCP/IP stack. |



10.2.3 TcplpGeneral

| | | | |
|---------------------------------|---|--|--|
| SWS Item | ECUC_Tcplp_00002 : | | |
| Container Name | TcplpGeneral | | |
| Description | This container is a subcontainer of Tcplp and specifies the general configuration parameters of the TCP/IP stack. | | |
| Configuration Parameters | | | |

| | | | |
|----------------------------------|---|----|---------------------------------------|
| SWS Item | ECUC_Tcplp_00016 : | | |
| Name | TcplpBufferMemory | | |
| Description | Memory size in bytes reserved for TCP/IP buffers. | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 4294967295 | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00004 : | | |
| Name | TcplpDevErrorDetect | | |
| Description | Switches the Default Error Tracer (Det) detection and notification ON or OFF. <ul style="list-style-type: none"> • true: enabled (ON). • false: disabled (OFF). | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00183 : | | |
| Name | TcplpDhcpServerEnabled | | |
| Description | Enables (TRUE) or disables (FALSE) the DHCP (Dynamic Host Configuration Protocol) Server. | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|---------------------|--------------------------------------|--|--|
| SWS Item | ECUC_Tcplp_00013 : | | |
| Name | TcplpMainFunctionPeriod | | |
| Description | Period of Tcplp_MainFunction in [s]. | | |
| Multiplicity | 1 | | |

| | | | |
|----------------------------------|-------------------------|----|---------------------------------------|
| Type | EcucFloatParamDef | | |
| Range | 0 .. INF | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00182 : | | |
| Name | TcplpResetIpAssignmentApi | | |
| Description | Enables/disables the API Tcplp_ResetIpAssignment of a DHCP-client. | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|--|--------------|
| SWS Item | ECUC_Tcplp_00169 : | | |
| Name | TcplpScalabilityClass | | |
| Description | In order to customize the Tcplp Stack to the specific needs of the user it can be scaled according to the scalability classes. | | |
| Multiplicity | 1 | | |
| Type | EcucEnumerationParamDef | | |
| Range | SC1 | IPv4 - In-Vehicle and Diagnostic Communication | |
| | SC2 | IPv6 - In-Vehicle and Diagnostic Communication | |
| | SC3 | IPv4 and IPv6 (Dual Stack) - In-Vehicle and Diagnostic Communication | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00008 : | | |
| Name | TcplpTcpEnabled | | |
| Description | Enables (TRUE) or disabled (FALSE) support of TCP (Transmission Control Protocol). | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|-----------------|---------------------------|--|--|
| SWS Item | ECUC_Tcplp_00014 : | | |
| Name | TcplpTcpSocketMax | | |

| | | | |
|----------------------------------|-------------------------------|----|---------------------------------------|
| Description | Maximum number of TCP sockets | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 65535 | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00009 : | | |
| Name | TcplpUdpEnabled | | |
| Description | Enables (TRUE) or disabled (FALSE) support of UDP (User Datagram Protocol) | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--------------------------------|----|---------------------------------------|
| SWS Item | ECUC_Tcplp_00015 : | | |
| Name | TcplpUdpSocketMax | | |
| Description | Maximum number of UDP sockets. | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 65535 | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00005 : | | |
| Name | TcplpVersionInfoApi | | |
| Description | If true the Tcplp_GetVersionInfo API is available. | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| Included Containers | | |
|----------------------------|---------------------|---|
| Container Name | Multiplicity | Scope / Dependency |
| TcplpV4General | 1 | This container is a subcontainer of Tcplp and specifies the general configuration parameters of the TCP/IP stack for IPv4 |
| TcplpV6General | 1 | This container is a subcontainer of Tcplp and specifies the |

| | |
|--|--|
| | general configuration parameters of the TCP/IP stack for IPv6. |
|--|--|

10.2.4 TcplpV4General

| | |
|---------------------------------|---|
| SWS Item | ECUC_Tcplp_00163 : |
| Container Name | TcplpV4General |
| Description | This container is a subcontainer of Tcplp and specifies the general configuration parameters of the TCP/IP stack for IPv4 |
| Configuration Parameters | |

| | | | | | | | | | | |
|----------------------------------|--|-------------------------|---|--------------|------------------|----|--|------------------------|----|--|
| SWS Item | ECUC_Tcplp_00006 : | | | | | | | | | |
| Name | TcplpArpEnabled | | | | | | | | | |
| Description | Enables (TRUE) or disables (FALSE) support of ARP (Address Resolution Protocol). | | | | | | | | | |
| Multiplicity | 1 | | | | | | | | | |
| Type | EcucBooleanParamDef | | | | | | | | | |
| Default value | -- | | | | | | | | | |
| Post-Build Variant Value | false | | | | | | | | | |
| Value Configuration Class | <table border="1"> <tr> <td>Pre-compile time</td> <td>X</td> <td>All Variants</td> </tr> <tr> <td>Link time</td> <td>--</td> <td></td> </tr> <tr> <td>Post-build time</td> <td>--</td> <td></td> </tr> </table> | Pre-compile time | X | All Variants | Link time | -- | | Post-build time | -- | |
| Pre-compile time | X | All Variants | | | | | | | | |
| Link time | -- | | | | | | | | | |
| Post-build time | -- | | | | | | | | | |
| Scope / Dependency | scope: local | | | | | | | | | |

| | | | | | | | | | | |
|----------------------------------|--|-------------------------|---|--------------|------------------|----|--|------------------------|----|--|
| SWS Item | ECUC_Tcplp_00011 : | | | | | | | | | |
| Name | TcplpAutoIpEnabled | | | | | | | | | |
| Description | Enables (TRUE) or disables (FALSE) the Auto-IP (automatic private IP addressing) sub-module. | | | | | | | | | |
| Multiplicity | 1 | | | | | | | | | |
| Type | EcucBooleanParamDef | | | | | | | | | |
| Default value | -- | | | | | | | | | |
| Post-Build Variant Value | false | | | | | | | | | |
| Value Configuration Class | <table border="1"> <tr> <td>Pre-compile time</td> <td>X</td> <td>All Variants</td> </tr> <tr> <td>Link time</td> <td>--</td> <td></td> </tr> <tr> <td>Post-build time</td> <td>--</td> <td></td> </tr> </table> | Pre-compile time | X | All Variants | Link time | -- | | Post-build time | -- | |
| Pre-compile time | X | All Variants | | | | | | | | |
| Link time | -- | | | | | | | | | |
| Post-build time | -- | | | | | | | | | |
| Scope / Dependency | scope: local | | | | | | | | | |

| | | | | | | | | | | |
|----------------------------------|--|-------------------------|---|--------------|------------------|----|--|------------------------|----|--|
| SWS Item | ECUC_Tcplp_00010 : | | | | | | | | | |
| Name | TcplpDhcpClientEnabled | | | | | | | | | |
| Description | Enables (TRUE) or disables (FALSE) the DHCP (Dynamic Host Configuration Protocol) Client. | | | | | | | | | |
| Multiplicity | 1 | | | | | | | | | |
| Type | EcucBooleanParamDef | | | | | | | | | |
| Default value | -- | | | | | | | | | |
| Post-Build Variant Value | false | | | | | | | | | |
| Value Configuration Class | <table border="1"> <tr> <td>Pre-compile time</td> <td>X</td> <td>All Variants</td> </tr> <tr> <td>Link time</td> <td>--</td> <td></td> </tr> <tr> <td>Post-build time</td> <td>--</td> <td></td> </tr> </table> | Pre-compile time | X | All Variants | Link time | -- | | Post-build time | -- | |
| Pre-compile time | X | All Variants | | | | | | | | |
| Link time | -- | | | | | | | | | |
| Post-build time | -- | | | | | | | | | |
| Scope / Dependency | scope: local | | | | | | | | | |

| | |
|--------------------|---|
| SWS Item | ECUC_Tcplp_00007 : |
| Name | TcplpIcmpEnabled |
| Description | Enables (TRUE) or disabled (FALSE) support of ICMP (Internet Control Message Protocol). |

| | | | |
|----------------------------------|-------------------------|----|--------------|
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00088 : | | |
| Name | TcplpV4Enabled | | |
| Description | Enables (TRUE) or disables (FALSE) support of IPv4 (Internet Protocol version 4). | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|---------------------------------------|
| SWS Item | ECUC_Tcplp_00018 : | | |
| Name | TcplpLocalAddrIpv4EntriesMax | | |
| Description | Maximum number of LocalAddr table entries for IPv4. | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 255 | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00012 : | | |
| Name | TcplpPathMtuDiscoveryEnabled | | |
| Description | Enables (TRUE) or disables (FALSE) the discovery of the maximum transmission unit on a path according to IETF RfC 1191. | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

No Included Containers

10.2.5 TcplpV6General

| | | | |
|---------------------------------|--|--|--|
| SWS Item | ECUC_Tcplp_00164 : | | |
| Container Name | TcplpV6General | | |
| Description | This container is a subcontainer of Tcplp and specifies the general configuration parameters of the TCP/IP stack for IPv6. | | |
| Configuration Parameters | | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00093 : | | |
| Name | TcplpDhcpV6ClientEnabled | | |
| Description | Enables (TRUE) or disables (FALSE) the DHCPv6 (Dynamic Host Configuration Protocol for IPv6) Client. | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00089 : | | |
| Name | TcplpV6Enabled | | |
| Description | Enables (TRUE) or disables (FALSE) support of IPv6 (Internet Protocol version 6). | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00090 : | | |
| Name | TcplpV6PathMtuDiscoveryEnabled | | |
| Description | Enables (TRUE) or disables (FALSE) Path MTU Discovery support for IPv6 according to IETF RFC 1981. | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

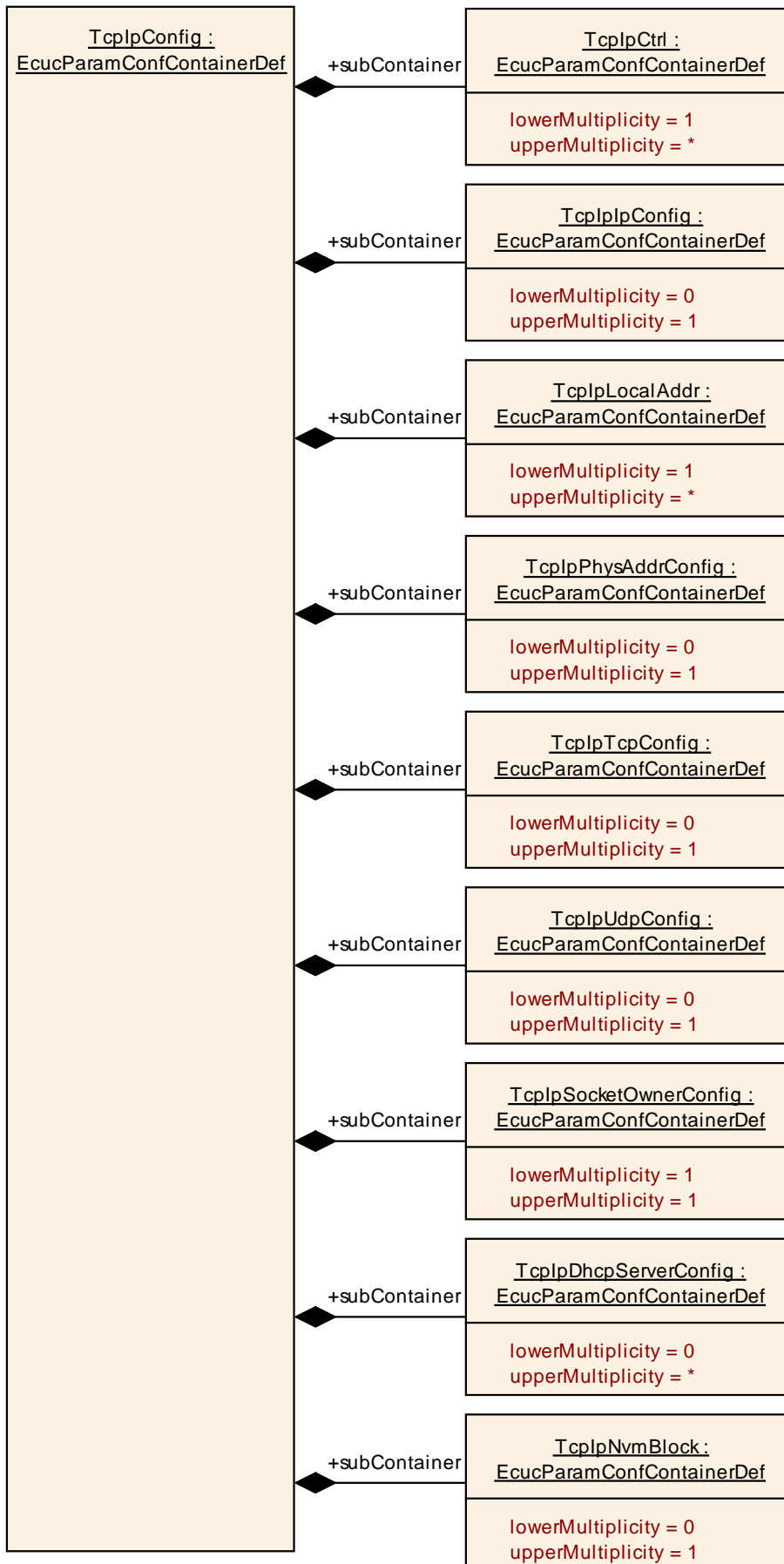
| | | | |
|----------------------------------|---|---|---------------------------------------|
| SWS Item | ECUC_Tcplp_00017 : | | |
| Name | TcplpLocalAddrIpv6EntriesMax | | |
| Description | Maximum number of LocalAddr table entries for IPv6. | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 255 | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |

| | | | |
|---------------------------|------------------------|----|--|
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00091 : | | |
| Name | TcplpNdpAddressResolutionUnreachabilityDetectionEnabled | | |
| Description | Enables (TRUE) or disables (FALSE) support of Address Resoulution and Neighbor Unreachability Detetion via NDP. | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00092 : | | |
| Name | TcplpNdpPrefixAndRouterDiscoveryEnabled | | |
| Description | Enables (TRUE) or disables (FALSE) support of Prefix and Router Discovery via NDP. | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

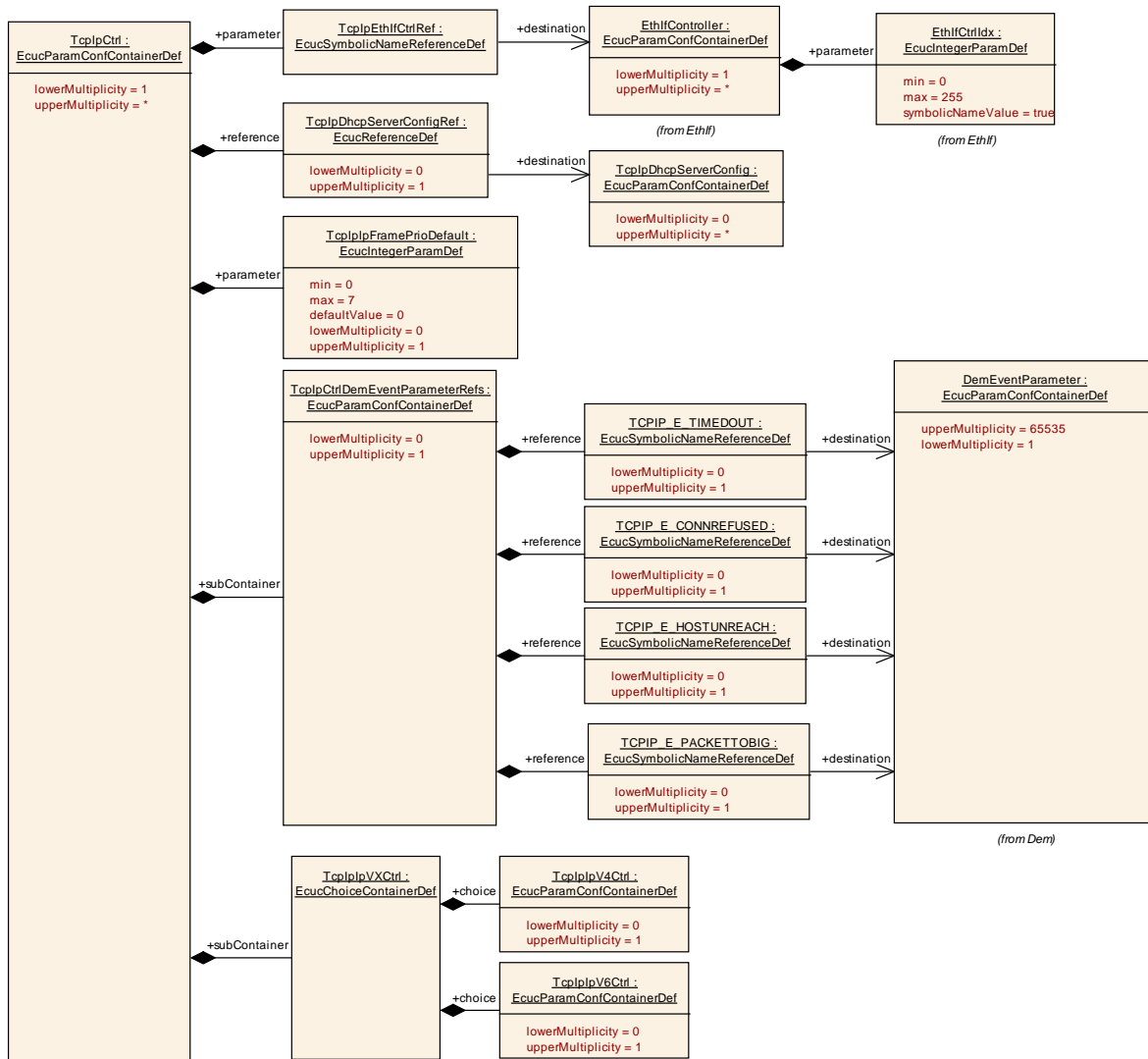
| |
|-------------------------------|
| No Included Containers |
|-------------------------------|



10.2.6 TcplpConfig

| | |
|---------------------------------|--|
| SWS Item | ECUC_Tcplp_00003 : |
| Container Name | TcplpConfig |
| Description | This container contains the configuration parameters and sub containers of the AUTOSAR Tcplp module. |
| Configuration Parameters | |

| Included Containers | | |
|----------------------------|---------------------|---|
| Container Name | Multiplicity | Scope / Dependency |
| TcplpCtrl | 1..* | Specifies the EthIf controller used for IP communication and Tcplp errors that shall be reported to DEM. |
| TcplpDhcpServerConfig | 0..* | Specifies the configuration parameters of the DHCP Server sub-module. |
| TcplpIpConfig | 0..1 | Specifies the configuration parameters of the IP (Internet Protocol) sub-module |
| TcplpLocalAddr | 1..* | Specifies the local IP (Internet Protocol) addresses used for IP communication. |
| TcplpNvmBlock | 0..1 | Configuration of optional usage of Nvm in case the Tcplp module requires non volatile memory in the Ecu to store information (e.g. IP Address received via DHCP and shall be stored). |
| TcplpPhysAddrConfig | 0..1 | Specifies the physical address configuration. |
| TcplpSocketOwnerConfig | 1 | Specifies the upper layer modules of Tcplp using the socket API. |
| TcplpTcpConfig | 0..1 | Specifies the configuration parameters of the TCP (Transmission Control Protocol) sub-module. |
| TcplpUdpConfig | 0..1 | Specifies the configuration parameters of the UDP (User Datagram Protocol) sub-module |



10.2.7 TcpIpCtrl

| | |
|---------------------------------|--|
| SWS Item | ECUC_Tcplp_00021 : |
| Container Name | TcplpCtrl |
| Description | Specifies the EthIf controller used for IP communication and Tcplp errors that shall be reported to DEM. |
| Configuration Parameters | |

| | |
|--|--|
| SWS Item | ECUC_Tcplp_00081 : |
| Name | TcplpFramePrioDefault |
| Description | Specifies the default value for the priority for all outgoing frames. Note: the value can be changed for each socket individually via Tcplp_ChangeParameter() service. If this optional parameter is not available, 0 is used as default priority. |
| Multiplicity | 0..1 |
| Type | EcucIntegerParamDef |
| Range | 0 .. 7 |
| Default value | 0 |
| Post-Build Variant Multiplicity | true |

| | | | |
|---|-------------------------|---|---------------------|
| Post-Build Variant Value | true | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|---|---------------------|
| SWS Item | ECUC_Tcplp_00195 : | | |
| Name | TcplpDhcpServerConfigRef | | |
| Description | Reference to a TcplpDhcpServerConfig which shall be used for this controller setting (VLAN). | | |
| Multiplicity | 0..1 | | |
| Type | Reference to [TcplpDhcpServerConfig] | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00041 : | | |
| Name | TcplpEthIfCtrlRef | | |
| Description | Reference to EthIf controller where the IP address shall be assigned. | | |
| Multiplicity | 1 | | |
| Type | Symbolic name reference to [EthIfController] | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| Included Containers | | |
|--------------------------------|---------------------|---|
| Container Name | Multiplicity | Scope / Dependency |
| TcplpCtrlDemEventParameterRefs | 0..1 | Specifies references to DEMEventParameter elements for Tcplp errors that shall be detected on the EthIf controller and reported to DEM. |
| TcplpVXCtrl | 1 | Specifies whether this controller is an Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) instance. |

10.2.8 TcplpCtrlDemEventParameterRefs

| | |
|---------------------------------|---|
| SWS Item | ECUC_Tcplp_00042 : |
| Container Name | TcplpCtrlDemEventParameterRefs |
| Description | This container is a subcontainer of TcplpCtrl and specifies the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus API in case the corresponding Tcplp error occurs for communication on the EthIf Controller. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references. |
| Configuration Parameters | |

| | | | |
|---|--|---|---------------------|
| SWS Item | ECUC_Tcplp_00049 : | | |
| Name | TCPIP_E_CONNREFUSED | | |
| Description | Reference to the DemEventParameter which shall be issued when the error "Connection refused" has occurred. | | |
| Multiplicity | 0..1 | | |
| Type | Symbolic name reference to [DemEventParameter] | | |
| Post-Build Variant Multiplicity | true | | |
| Post-Build Variant Value | true | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|---|--|---|---------------------|
| SWS Item | ECUC_Tcplp_00051 : | | |
| Name | TCPIP_E_HOSTUNREACH | | |
| Description | Reference to the DemEventParameter which shall be issued when the error "No route to host" has occurred. | | |
| Multiplicity | 0..1 | | |
| Type | Symbolic name reference to [DemEventParameter] | | |
| Post-Build Variant Multiplicity | true | | |
| Post-Build Variant Value | true | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|---|--|---|---------------------|
| SWS Item | ECUC_Tcplp_00196 : | | |
| Name | TCPIP_E_PACKETTOBIG | | |
| Description | Reference to the DemEventParameter which shall be issued when the error "Path does not support frame size" has occurred. | | |
| Multiplicity | 0..1 | | |
| Type | Symbolic name reference to [DemEventParameter] | | |
| Post-Build Variant Multiplicity | true | | |
| Post-Build Variant Value | true | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|--------------------|---|--|--|
| SWS Item | ECUC_Tcplp_00048 : | | |
| Name | TCPIP_E_TIMEDOUT | | |
| Description | Reference to the DemEventParameter which shall be issued when the error "Operation timed out" has occurred. | | |

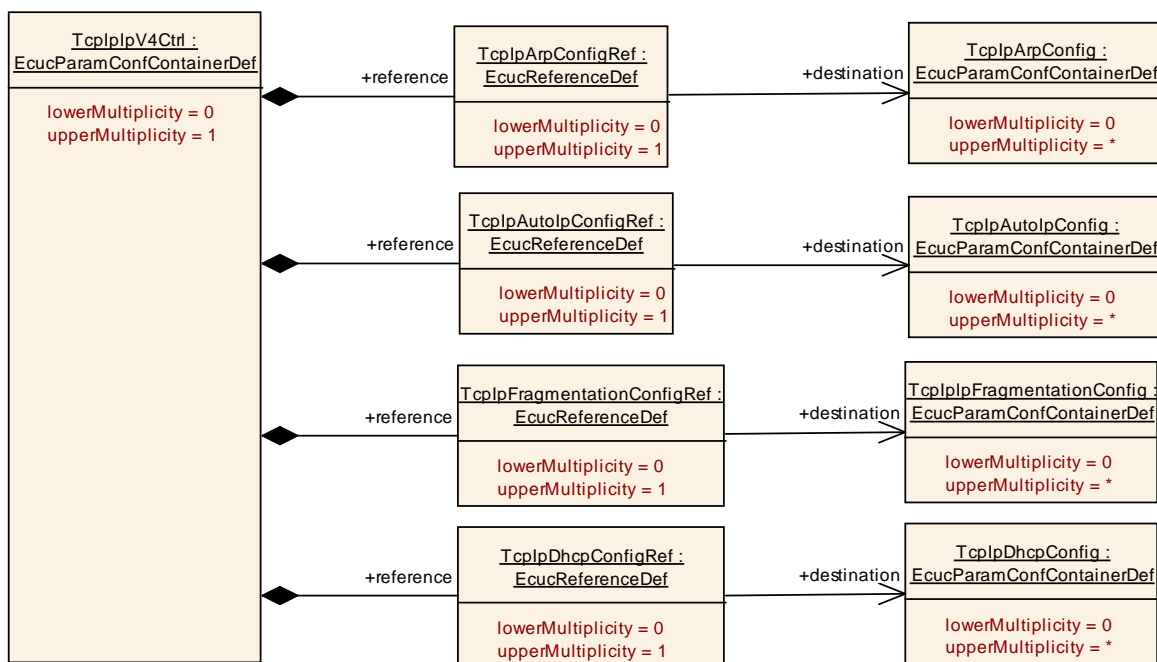
| | | | |
|---|--|---|---------------------|
| Multiplicity | 0..1 | | |
| Type | Symbolic name reference to [DemEventParameter] | | |
| Post-Build Variant Multiplicity | true | | |
| Post-Build Variant Value | true | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

No Included Containers

10.2.9 TcplpVXCtrl

| | |
|------------------------------|--|
| SWS Item | ECUC_Tcplp_00094 : |
| Choice container Name | TcplpVXCtrl |
| Description | Specifies whether this controller is an Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) instance. |

| Container Choices | | |
|--------------------------|---------------------|---|
| Container Name | Multiplicity | Scope / Dependency |
| TcplpV4Ctrl | 0..1 | Specifies an Internet Protocol version 4 (IPv4) instance. |
| TcplpV6Ctrl | 0..1 | Specifies an Internet Protocol version 6 (IPv6) instance. |



10.2.10 TcplpV4Ctrl

| | |
|---------------------------------|---|
| SWS Item | ECUC_Tcplp_00166 : |
| Container Name | TcplpV4Ctrl |
| Description | Specifies an Internet Protocol version 4 (IPv4) instance. |
| Configuration Parameters | |

| | | | |
|---|---|----|--------------|
| SWS Item | ECUC_Tcplp_00097 : | | |
| Name | TcplpArpConfigRef | | |
| Description | Reference to ARP configuration for this IPv4 instance. (Multiple IPv4 instances may use the same configuration container but will operate independently) | | |
| Multiplicity | 0..1 | | |
| Type | Reference to [TcplpArpConfig] | | |
| Post-Build Variant Multiplicity | false | | |
| Post-Build Variant Value | false | | |
| Multiplicity Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

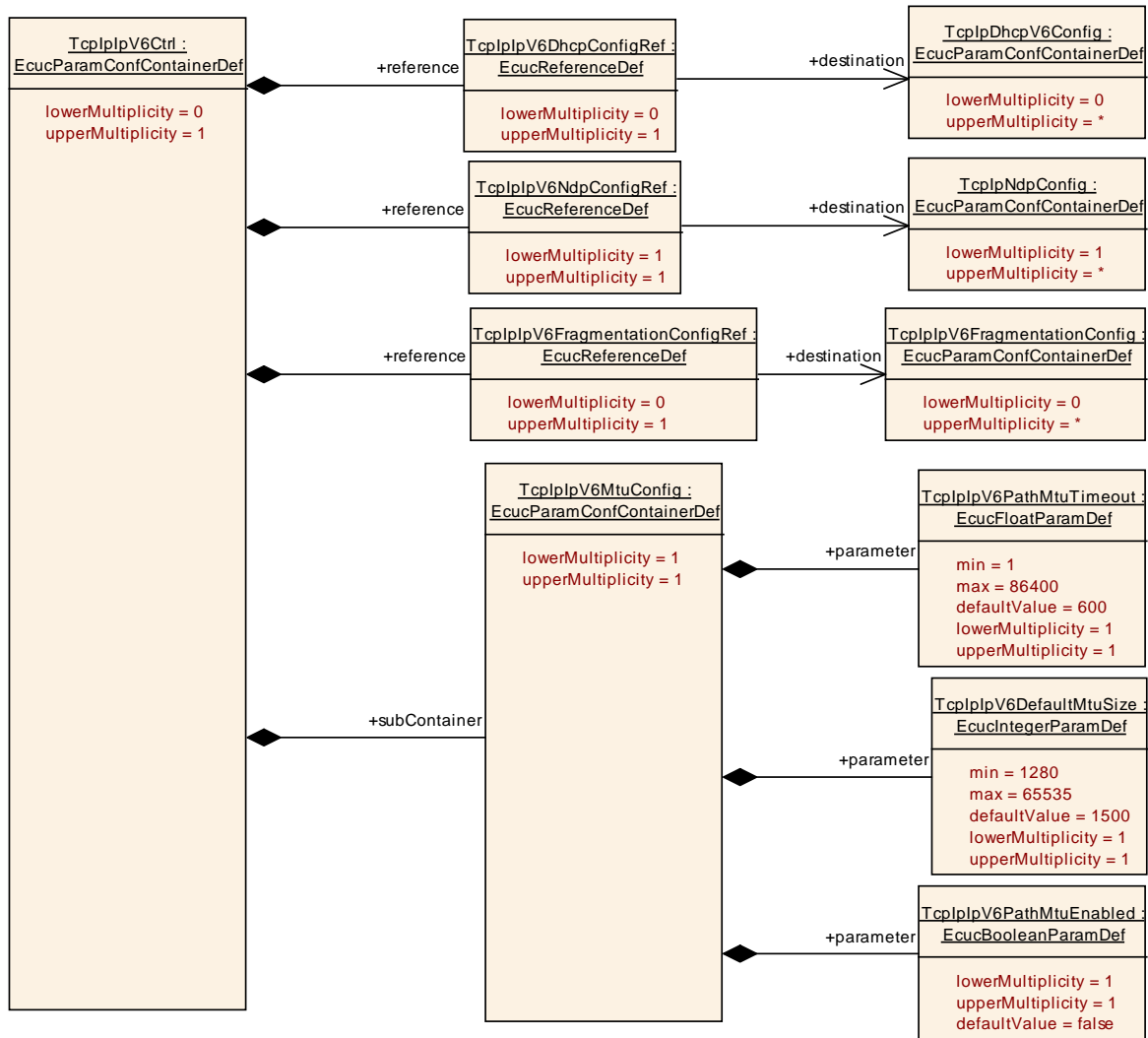
| | | | |
|---|--|----|--------------|
| SWS Item | ECUC_Tcplp_00098 : | | |
| Name | TcplpAutoIpConfigRef | | |
| Description | Reference to AutoIp configuration for this IPv4 instance. (Multiple IPv4 instances may use the same configuration container but will operate independently) | | |
| Multiplicity | 0..1 | | |
| Type | Reference to [TcplpAutoIpConfig] | | |
| Post-Build Variant Multiplicity | false | | |
| Post-Build Variant Value | false | | |
| Multiplicity Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|---|--|----|--------------|
| SWS Item | ECUC_Tcplp_00100 : | | |
| Name | TcplpDhcpConfigRef | | |
| Description | Reference to DHCP configuration for this IPv4 instance. (Multiple IPv4 instances may use the same configuration container but will operate independently) | | |
| Multiplicity | 0..1 | | |
| Type | Reference to [TcplpDhcpConfig] | | |
| Post-Build Variant Multiplicity | false | | |
| Post-Build Variant Value | false | | |
| Multiplicity Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |

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

| | | | |
|---|---|----|--------------|
| SWS Item | ECUC_Tcplp_00099 : | | |
| Name | TcplpFragmentationConfigRef | | |
| Description | Reference to Fragmentation configuration for this IPv4 instance. (Multiple IPv4 instances may use the same configuration container but will operate independently) | | |
| Multiplicity | 0..1 | | |
| Type | Reference to [TcplpFragmentationConfig] | | |
| Post-Build Variant Multiplicity | false | | |
| Post-Build Variant Value | false | | |
| Multiplicity Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

No Included Containers



10.2.11 TcplpV6Ctrl

| | | | |
|---------------------------------|---|--|--|
| SWS Item | ECUC_Tcplp_00096 : | | |
| Container Name | TcplpV6Ctrl | | |
| Description | Specifies an Internet Protocol version 6 (IPv6) instance. | | |
| Configuration Parameters | | | |

| | | | |
|---|---|----|--------------|
| SWS Item | ECUC_Tcplp_00101 : | | |
| Name | TcplpV6DhcpConfigRef | | |
| Description | Reference to DHCPv6 configuration. (Multiple IPv6 instances may use the same configuration container but will operate independently) | | |
| Multiplicity | 0..1 | | |
| Type | Reference to [TcplpDhcpV6Config] | | |
| Post-Build Variant Multiplicity | false | | |
| Post-Build Variant Value | false | | |
| Multiplicity Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Value Configuration Class | Pre-compile time | X | All Variants |

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

| | | | |
|---|---|----|--------------|
| SWS Item | ECUC_Tcplp_00103 : | | |
| Name | TcplpV6FragmentationConfigRef | | |
| Description | Reference to IPv6 Fragmentation Configuration. (Multiple IPv6 instances may use the same configuration container but will operate independently) | | |
| Multiplicity | 0..1 | | |
| Type | Reference to [TcplpV6FragmentationConfig] | | |
| Post-Build Variant Multiplicity | false | | |
| Post-Build Variant Value | false | | |
| Multiplicity Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00102 : | | |
| Name | TcplpV6NdpConfigRef | | |
| Description | Reference to Neighbor Discovery Protocol Configuration. (Multiple IPv6 instances may use the same configuration container but will operate independently) | | |
| Multiplicity | 1 | | |
| Type | Reference to [TcplpNdpConfig] | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| Included Containers | | |
|----------------------------|---------------------|---|
| Container Name | Multiplicity | Scope / Dependency |
| TcplpV6MtuConfig | 1 | This container specifies the Maximum Transmission Unit parameters for this IPv6 instance. |

10.2.12 TcplpV6MtuConfig

| | | | |
|---------------------------------|---|--|--|
| SWS Item | ECUC_Tcplp_00104 : | | |
| Container Name | TcplpV6MtuConfig | | |
| Description | This container specifies the Maximum Transmission Unit parameters for this IPv6 instance. | | |
| Configuration Parameters | | | |

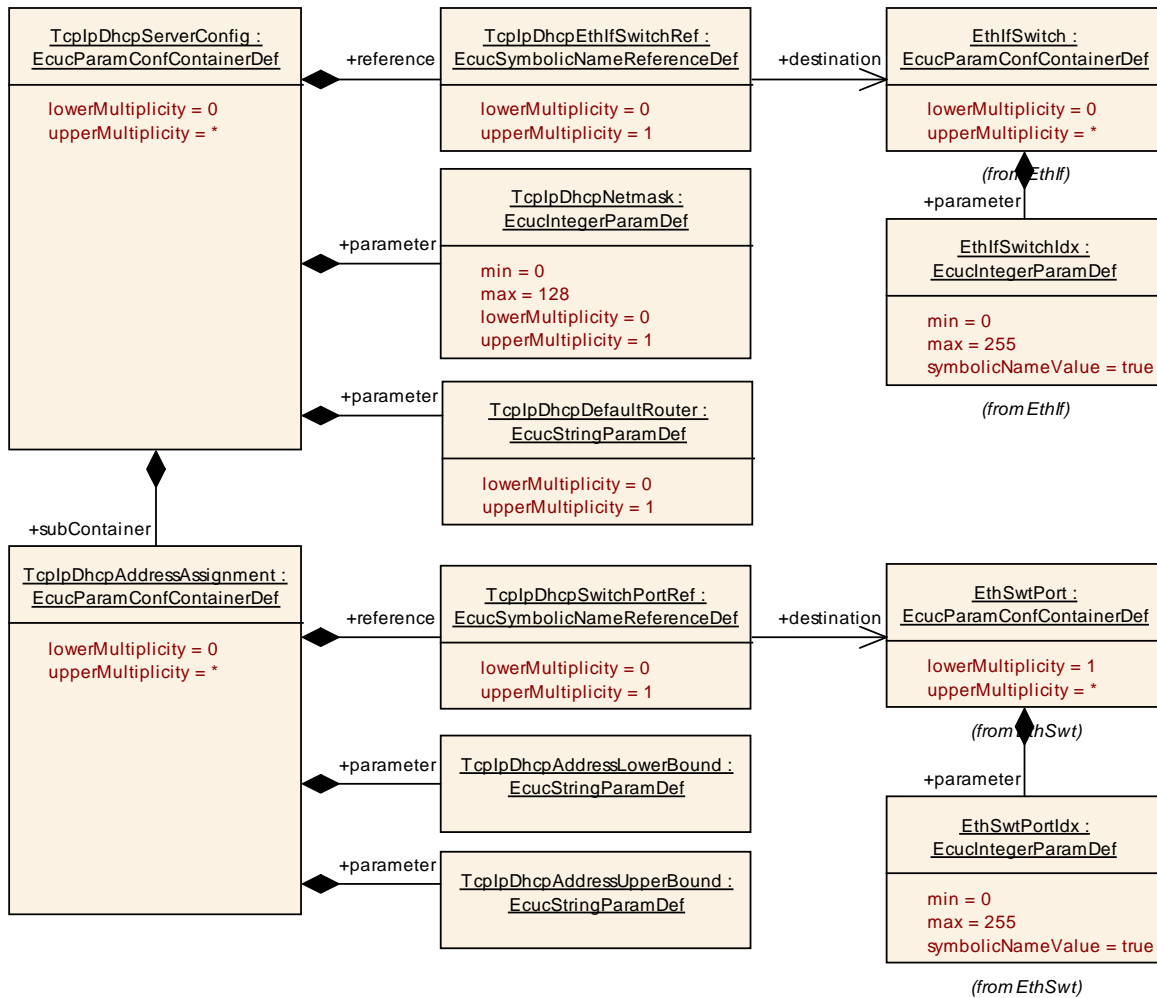
| | | | |
|--------------------|---|--|--|
| SWS Item | ECUC_Tcplp_00106 : | | |
| Name | TcplpV6DefaultMtuSize | | |
| Description | Maximum Transmission Unit (MTU) of the link. [RFC2460 5. Packet Size Issues] "IPv6 requires that every link in the internet have an MTU of 1280 octets or greater. On any link that cannot | | |

| | | | |
|---------------------------------|--|--|--|
| | convey a 1280-octet packet in one piece, link-specific fragmentation and reassembly must be provided at a layer below IPv6. Links that have a configurable MTU (for example, PPP links [RFC-1661]) must be configured to have an MTU of at least 1280 octets; it is recommended that they be configured with an MTU of 1500 octets or greater, to accommodate possible encapsulations (i.e., tunneling) without incurring IPv6-layer fragmentation. From each link to which a node is directly attached, the node must be able to accept packets as large as that link's MTU." | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 1280 .. 65535 | | |
| Default value | 1500 | | |
| Post-Build Variant Value | false | | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00107 : | | |
| Name | TcplpV6PathMtuEnabled | | |
| Description | If enabled the IPv6 processes incoming ICMPv6 "Packet Too Big" messages and stores a MTU value for each destination address. See RFC1981 "Path MTU Discovery for IP version 6" for details about PathMTU. | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | false | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00105 : | | |
| Name | TcplpV6PathMtuTimeout | | |
| Description | If this value is >0 the IPv6 will reset the MTU value stored for each destination after n seconds. see [RFC1981 5.3. Purging stale PMTU information] Default: 600 seconds (10 minutes) | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 1 .. 86400 | | |
| Default value | 600 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

No Included Containers



10.2.13 TcpIpDhcpServerConfig

| | | | |
|---|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00187 : | | |
| Container Name | TcpIpDhcpServerConfig | | |
| Description | Specifies the configuration parameters of the DHCP Server sub-module. | | |
| Post-Build Variant | true | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Configuration Parameters | | | |

| | | | |
|---------------------------|---|--|--|
| SWS Item | ECUC_Tcplp_00190 : | | |
| Name | TcpIpDhcpDefaultRouter | | |
| Description | IP address of default router (gateway). | | |
| Multiplicity | 0..1 | | |
| Type | EcucStringParamDef | | |
| Default value | -- | | |
| maxLength | -- | | |
| minLength | -- | | |
| regularExpression | -- | | |
| Post-Build Variant | true | | |

| | | | |
|---|-------------------------|---|---------------------|
| Multiplicity | | | |
| Post-Build Variant Value | true | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|---|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00189 : | | |
| Name | TcplpDhcpNetmask | | |
| Description | Network mask of IPv4 address or address prefix of IPv6 address in CIDR Notation, i.e. decimal value between 0 and 32 (IPv4) or 0 and 128 (IPv6) that describes the number of significant bits defining the network number or prefix of an IP address. | | |
| Multiplicity | 0..1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 128 | | |
| Default value | -- | | |
| Post-Build Variant Multiplicity | true | | |
| Post-Build Variant Value | true | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|---|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00188 : | | |
| Name | TcplpDhcpEthIfSwitchRef | | |
| Description | Reference to EthIfSwitch representation. Optional in case the Dhcp server is operating without an Ethernet switch. | | |
| Multiplicity | 0..1 | | |
| Type | Symbolic name reference to [EthIfSwitch] | | |
| Post-Build Variant Multiplicity | true | | |
| Post-Build Variant Value | true | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: ECU | | |

| | | |
|----------------------------|---------------------|---|
| Included Containers | | |
| Container Name | Multiplicity | Scope / Dependency |
| TcplpDhcpAddressAssignment | 0..* | Defines a Ethernet Switch port based IP address assignment. |

10.2.14 TcplpDhcpAddressAssignment

| | | | |
|---|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00191 : | | |
| Container Name | TcplpDhcpAddressAssignment | | |
| Description | Defines a Ethernet Switch port based IP address assignment. | | |
| Post-Build Variant Multiplicity | true | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Configuration Parameters | | | |

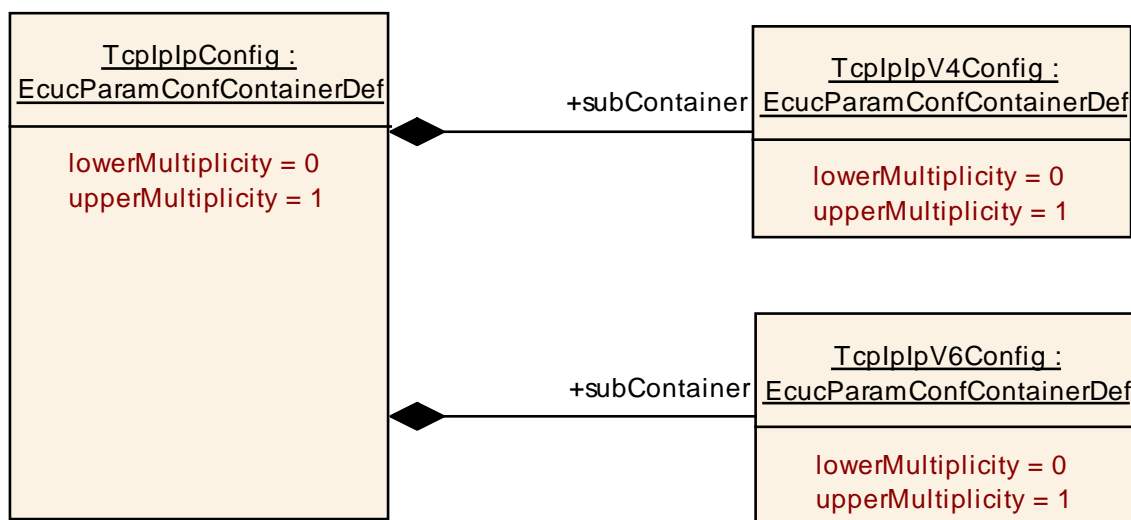
| | | | |
|----------------------------------|--|---|---------------------|
| SWS Item | ECUC_Tcplp_00193 : | | |
| Name | TcplpDhcpAddressLowerBound | | |
| Description | The lower bound IP address which shall be assigned. If lower bound and upper bound are identical exactly this IP address shall be assigned. | | |
| Multiplicity | 1 | | |
| Type | EcucStringParamDef | | |
| Default value | -- | | |
| maxLength | -- | | |
| minLength | -- | | |
| regularExpression | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|---|---------------------|
| SWS Item | ECUC_Tcplp_00194 : | | |
| Name | TcplpDhcpAddressUpperBound | | |
| Description | The upper bound IP address which shall be assigned. If lower bound and upper bound are identical exactly this IP address shall be assigned. | | |
| Multiplicity | 1 | | |
| Type | EcucStringParamDef | | |
| Default value | -- | | |
| maxLength | -- | | |
| minLength | -- | | |
| regularExpression | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|--|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00192 : | | |
| Name | TcplpDhcpSwitchPortRef | | |
| Description | Reference to Ethernet Switch port. Optional in case the Dhcp server is operating without an Ethernet switch. | | |
| Multiplicity | 0..1 | | |
| Type | Symbolic name reference to [EthSwPort] | | |
| Post-Build Variant Multiplicity | true | | |
| Post-Build Variant Value | true | | |
| Multiplicity Configuration | Pre-compile time | X | VARIANT-PRE-COMPILE |

| | | | |
|----------------------------------|-------------------------|---|---------------------|
| Class | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: ECU | | |

No Included Containers



10.2.15 TcpIplpConfig

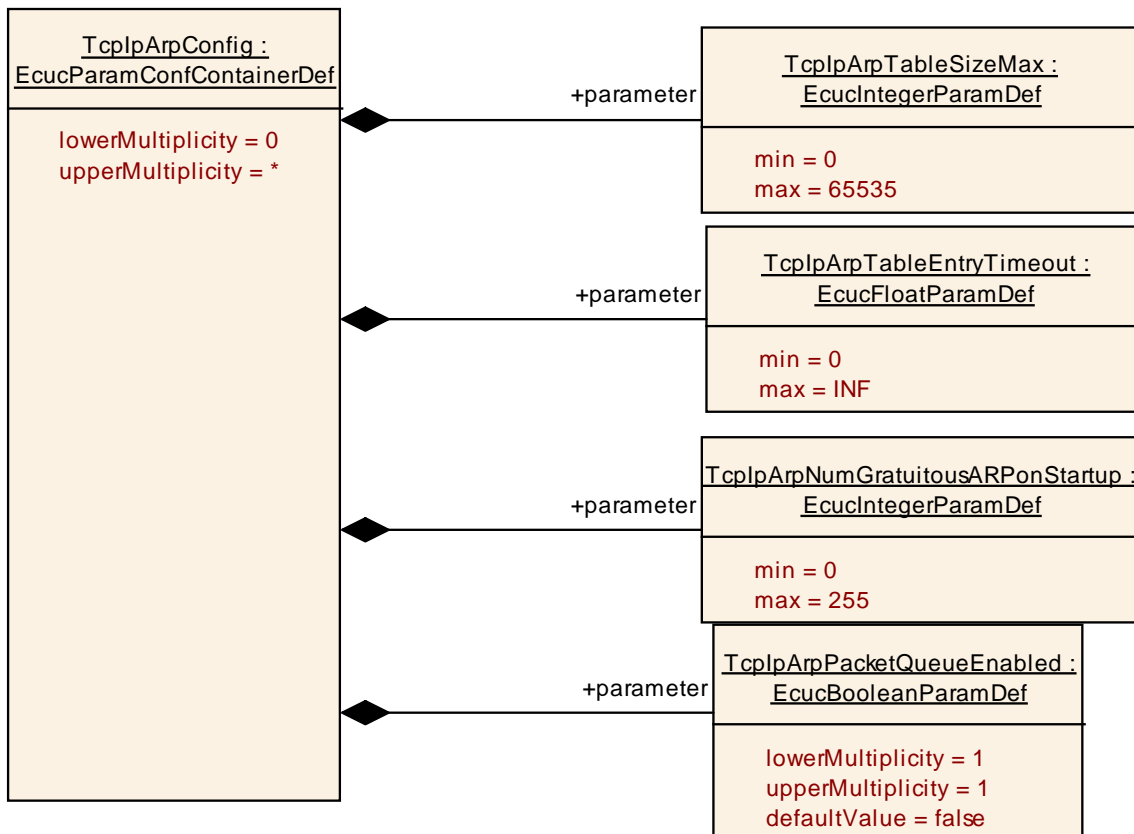
| | |
|---------------------------------|---|
| SWS Item | ECUC_Tcplp_00022 : |
| Container Name | TcplpConfig |
| Description | Specifies the configuration parameters of the IP (Internet Protocol) sub-module |
| Configuration Parameters | |

| | | |
|----------------------------|---------------------|--|
| Included Containers | | |
| Container Name | Multiplicity | Scope / Dependency |
| TcplpV4Config | 0..1 | Specifies the configuration parameters of the IPv4 (Internet Protocol version 4) sub-module. |
| TcplpV6Config | 0..1 | Specifies the configuration parameters of the IPv6 (Internet Protocol version 6) sub-module. |

10.2.16 TcpIplpV4Config

| | |
|---------------------------------|--|
| SWS Item | ECUC_Tcplp_00095 : |
| Container Name | TcplpV4Config |
| Description | Specifies the configuration parameters of the IPv4 (Internet Protocol version 4) sub-module. |
| Configuration Parameters | |

| Included Containers | | |
|----------------------------|---------------------|---|
| Container Name | Multiplicity | Scope / Dependency |
| TcpIpArpConfig | 0..* | Specifies the configuration parameters of the ARP (Address Resolution Protocol) sub-module. |
| TcpIpAutoIpConfig | 0..* | Specifies the configuration parameters of the Auto-IP (automatic private IP addressing) sub-module. |
| TcpIpDhcpConfig | 0..* | Specifies the configuration parameters of the DHCPv4. This container may be referenced by multiple IPv4 instances if they shall use the same configuration. This container may have multiple instances if different configurations are required for different IPv4 instances. |
| TcpIpIcmpConfig | 0..1 | Specifies the configuration parameters of the ICMP (Internet Control Message Protocol) sub-module. |
| TcpIpIpFragmentationConfig | 0..* | Specifies the configuration parameters of IPv4 packet fragmentation/reassembly. This container may be referenced by multiple IPv4 instances if they shall use the same configuration. This container may have multiple instances if different configurations are required for different IPv4 instances. |



10.2.17 TcpIpArpConfig

| | |
|-----------------------|---------------------------|
| SWS Item | ECUC_TcpIp_00023 : |
| Container Name | TcpIpArpConfig |

| | |
|---------------------------------|---|
| Description | Specifies the configuration parameters of the ARP (Address Resolution Protocol) sub-module. |
| Configuration Parameters | |

| | | | |
|----------------------------------|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00054 : | | |
| Name | TcplpArpNumGratuitousARPOnStartup | | |
| Description | Specifies the number of gratuitous ARP replies which shall be sent on assignment of a new IP address. | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 255 | | |
| Default value | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

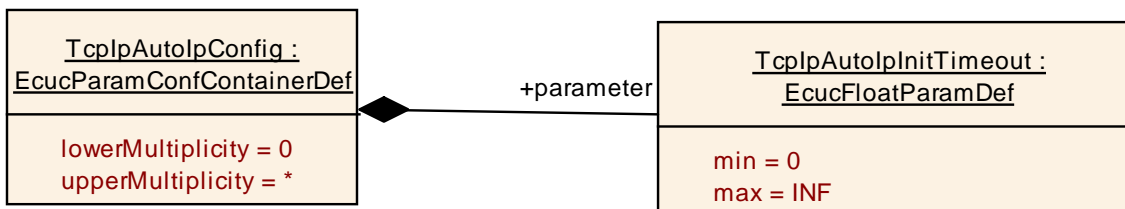
| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00170 : | | |
| Name | TcplpArpPacketQueueEnabled | | |
| Description | Enables (TRUE) or disables (FALSE) support of the ARP Packet Queue according to IETF RFC 1122, section 2.3.2.2. | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | false | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|---|---------------------|
| SWS Item | ECUC_Tcplp_00053 : | | |
| Name | TcplpArpTableEntryTimeout | | |
| Description | Timeout in seconds after which an unused ARP entry is removed. | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0 .. INF | | |
| Default value | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|---------------------------------------|
| SWS Item | ECUC_Tcplp_00052 : | | |
| Name | TcplpArpTableSizeMax | | |
| Description | Maximum number of entries in the ARP table. | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 65535 | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |

| | |
|---------------------------|--------------|
| Scope / Dependency | scope: local |
|---------------------------|--------------|

| |
|-------------------------------|
| No Included Containers |
|-------------------------------|

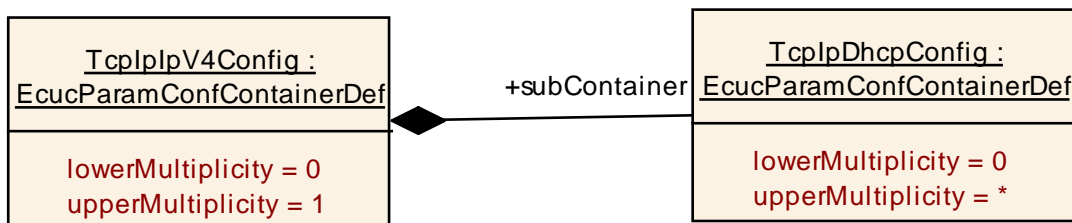


10.2.18 TcplpAutolpConfig

| | | | |
|---------------------------------|---|--|--|
| SWS Item | ECUC_Tcplp_00028 : | | |
| Container Name | TcplpAutolpConfig | | |
| Description | Specifies the configuration parameters of the Auto-IP (automatic private IP addressing) sub-module. | | |
| Configuration Parameters | | | |

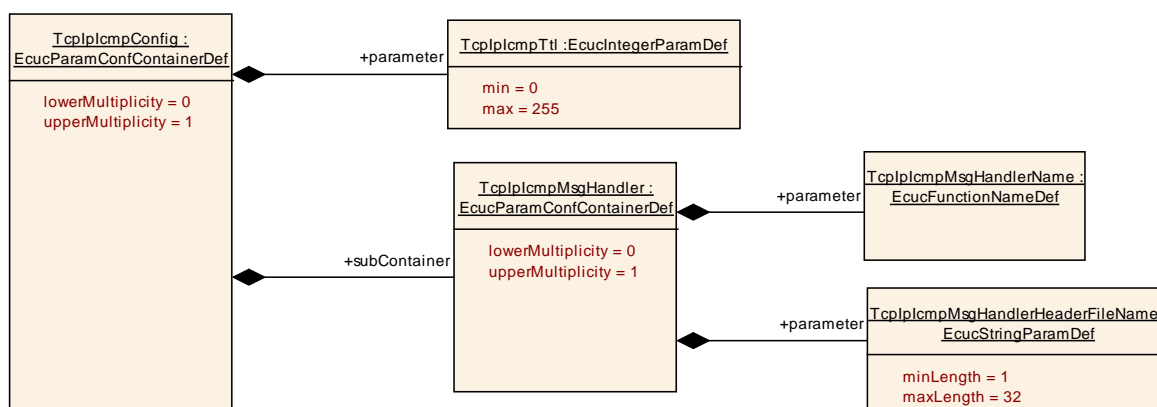
| | | | |
|----------------------------------|--|---|---------------------|
| SWS Item | ECUC_Tcplp_00074 : | | |
| Name | TcplpAutolpInitTimeout | | |
| Description | The time in seconds Auto-IP waits at startup, before beginning with ARP probing. This delay is used to give DHCP time to acquire a lease in case a DHCP server is present. | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0 .. INF | | |
| Default value | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| |
|-------------------------------|
| No Included Containers |
|-------------------------------|



10.2.19 TcplpDhcpConfig

| | |
|---------------------------------|---|
| SWS Item | ECUC_Tcplp_00167 : |
| Container Name | TcplpDhcpConfig |
| Description | Specifies the configuration parameters of the DHCPv4. This container may be referenced by multiple IPv4 instances if they shall use the same configuration. This container may have multiple instances if different configurations are required for different IPv4 instances. |
| Configuration Parameters | |
| No Included Containers | |



10.2.20 TcplpIcmpConfig

| | |
|---------------------------------|--|
| SWS Item | ECUC_Tcplp_00024 : |
| Container Name | TcplpIcmpConfig |
| Description | Specifies the configuration parameters of the ICMP (Internet Control Message Protocol) sub-module. |
| Configuration Parameters | |

| | | | |
|----------------------------------|--|---|---------------------|
| SWS Item | ECUC_Tcplp_00055 : | | |
| Name | TcplpIcmpTtl | | |
| Description | Default Time-to-live value of outgoing ICMP packets. | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 255 | | |
| Default value | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | |
|----------------------------|---------------------|--|
| Included Containers | | |
| Container Name | Multiplicity | Scope / Dependency |
| TcplpIcmpMsgHandler | 0..1 | This container is a subcontainer of TcplpIcmpConfig and specifies the configuration parameters for the ICMP message handler. |

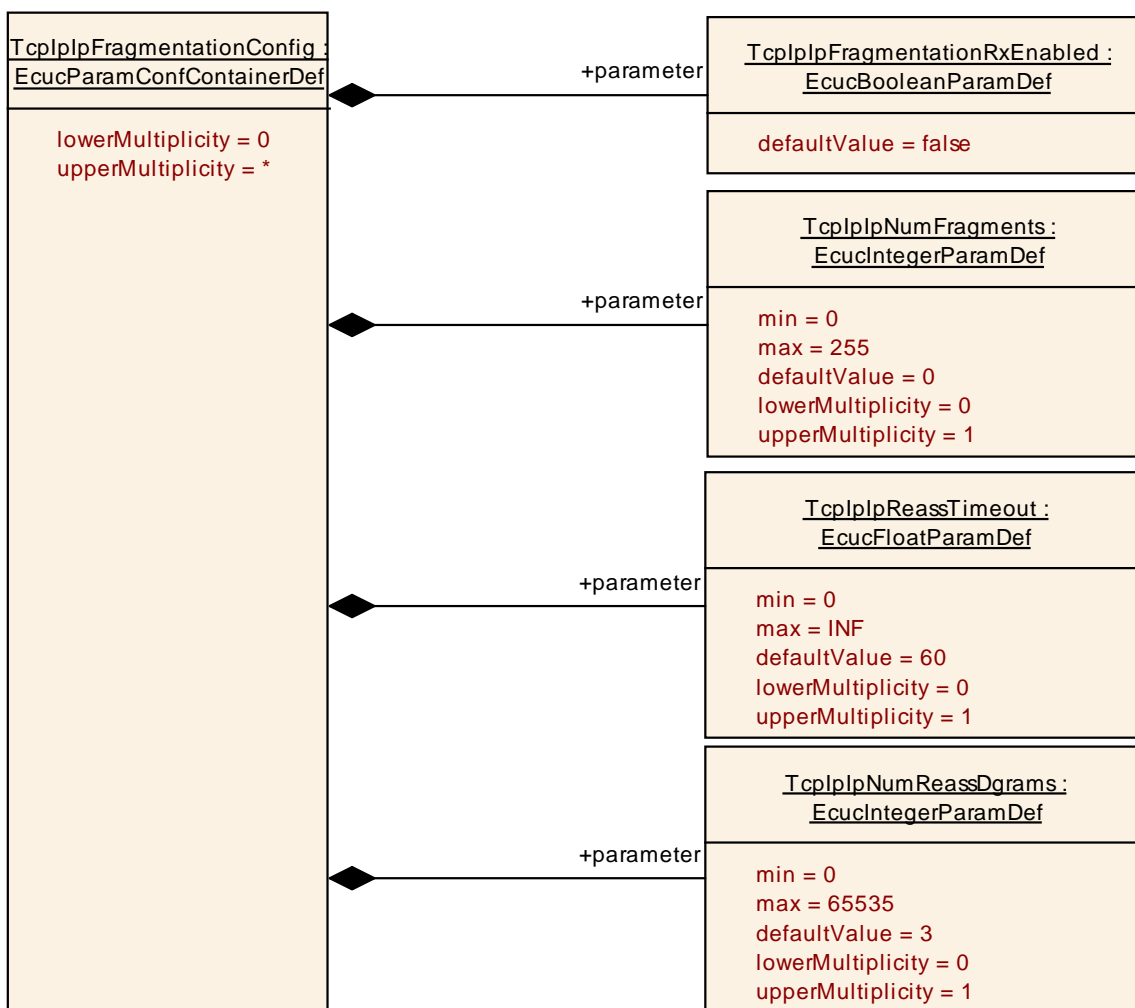
10.2.21 TcplplcmpMsgHandler

| | |
|---------------------------------|--|
| SWS Item | ECUC_Tcplp_00056 : |
| Container Name | TcplplcmpMsgHandler |
| Description | This container is a subcontainer of TcplplcmpConfig and specifies the configuration parameters for the ICMP message handler. |
| Configuration Parameters | |

| | | | |
|----------------------------------|--|----|---------------------------------------|
| SWS Item | ECUC_Tcplp_00058 : | | |
| Name | TcplplcmpMsgHandlerHeaderFileName | | |
| Description | This parameter specifies the name of the header file containing the definition of the ICMP message handler function. | | |
| Multiplicity | 1 | | |
| Type | EcucStringParamDef | | |
| Default value | -- | | |
| maxLength | 32 | | |
| minLength | 1 | | |
| regularExpression | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|---------------------------------------|
| SWS Item | ECUC_Tcplp_00057 : | | |
| Name | TcplplcmpMsgHandlerName | | |
| Description | This parameter defines the name of the ICMP message handler function <User_IcmpMsgHandler>. | | |
| Multiplicity | 1 | | |
| Type | EcucFunctionNameDef | | |
| Default value | -- | | |
| maxLength | -- | | |
| minLength | -- | | |
| regularExpression | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| |
|-------------------------------|
| No Included Containers |
|-------------------------------|



10.2.22 TcplpFragmentationConfig

| | |
|---------------------------------|--|
| SWS Item | ECUC_Tcplp_00108 : |
| Container Name | TcplpFragmentationConfig |
| Description | Specifies the configuration parameters of IPv4 packet fragmentation/reassembly. This container may be referenced by multiple IPv4 instances if they shall use the same configuration. This container may have multiple instances if different configurations are required for different IPv4 instances. |
| Configuration Parameters | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00077 : | | |
| Name | TcplpFragmentationRxEnabled | | |
| Description | Enables (TRUE) or disables (FALSE) support for reassembling of incoming datagrams that are fragmented according to IETF RFC 815 (IP Datagram Reassembly Algorithms). | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | false | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |

| | | | |
|---------------------------|------------------------|----|--|
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|---|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00078 : | | |
| Name | TcplpNumFragments | | |
| Description | Specifies the maximum number of IP fragments per datagram. Note: this parameter is only relevant if TcplpFragmentationRxEnabled is TRUE. | | |
| Multiplicity | 0..1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 255 | | |
| Default value | 0 | | |
| Post-Build Variant Multiplicity | true | | |
| Post-Build Variant Value | true | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local dependency: TcplpFragmentationRxEnabled | | |

| | | | |
|---|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00080 : | | |
| Name | TcplpNumReassDgrams | | |
| Description | Specifies the maximum number of fragmented IP datagrams that can be reassembled in parallel. Note: this parameter is only relevant if TcplpFragmentationRxEnabled is TRUE. | | |
| Multiplicity | 0..1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 65535 | | |
| Default value | 3 | | |
| Post-Build Variant Multiplicity | true | | |
| Post-Build Variant Value | true | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local dependency: TcplpFragmentationRxEnabled | | |

| | | | |
|----------------------|--|--|--|
| SWS Item | ECUC_Tcplp_00079 : | | |
| Name | TcplpReassTimeout | | |
| Description | Specifies the timeout in [s] after which an incomplete datagram gets discarded. Note: this parameter is only relevant if TcplpFragmentationRxEnabled is TRUE. | | |
| Multiplicity | 0..1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0 .. INF | | |
| Default value | 60 | | |

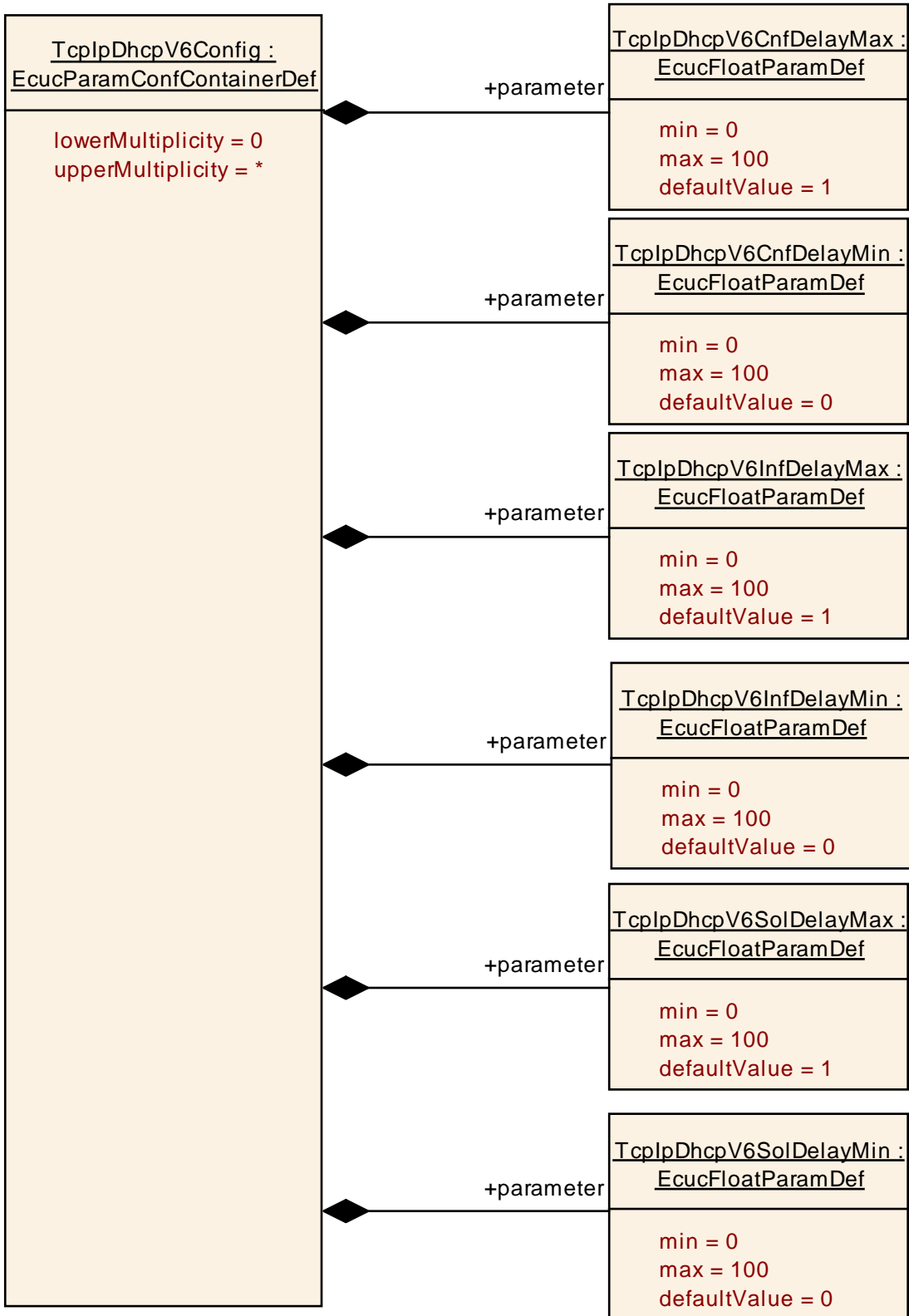
| | | | |
|---|---|---|---------------------|
| Post-Build Variant Multiplicity | true | | |
| Post-Build Variant Value | true | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local dependency: TcplpFragmentationRxEnabled | | |

No Included Containers

10.2.23 TcplpV6Config

| | |
|---------------------------------|--|
| SWS Item | ECUC_Tcplp_00168 : |
| Container Name | TcplpV6Config |
| Description | Specifies the configuration parameters of the IPv6 (Internet Protocol version 6) sub-module. |
| Configuration Parameters | |

| Included Containers | | |
|----------------------------|---------------------|---|
| Container Name | Multiplicity | Scope / Dependency |
| TcplpDhcpV6Config | 0..* | Specifies the configuration parameters of the DHCPv6. This container may be referenced by multiple IPv6 instances if they shall use the same configuration. This container may have multiple instances if different configurations are required for different IPv6 instances. |
| TcplpIcmpV6Config | 1 | Specifies the configuration parameters of the ICMPv6 (Internet Control Message Protocol for IPv6) sub-module. |
| TcplpV6FragmentationConfig | 0..* | Specifies the configuration parameters of IPv6 packet fragmentation/reassembly. This container may be referenced by multiple IPv6 instances if they shall use the same configuration. This container may have multiple instances if different configurations are required for different IPv6 instances. |
| TcplpNdpConfig | 1..* | Specifies the configuration parameters of the Neighbor Discovery Protocol for IPv6. This container may be referenced by multiple IPv6 instances if they shall use the same configuration. This container may have multiple instances if different configurations are required for different IPv6 instances. |



10.2.24 TcpIpDhcpV6Config

| | |
|-----------------|---------------------------|
| SWS Item | ECUC_Tcplp_00110 : |
|-----------------|---------------------------|

| | |
|---------------------------------|--|
| Container Name | TcplpDhcpV6Config |
| Description | Specifies the configuration parameters of the DHCPv6. This container may be referenced by multiple IPv6 instances if they shall use the same configuration. This container may have multiple instances if different configurations are required for different IPv6 instances. |
| Configuration Parameters | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00116 : | | |
| Name | TcplpDhcpV6CnfDelayMax | | |
| Description | Maximum delay (s) before sending the first Confirm message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval. | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0 .. 100 | | |
| Default value | 1 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00117 : | | |
| Name | TcplpDhcpV6CnfDelayMin | | |
| Description | Minimum delay (s) before the first Confirm message will be sent. | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0 .. 100 | | |
| Default value | 0 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00118 : | | |
| Name | TcplpDhcpV6InfDelayMax | | |
| Description | Maximum delay (s) before sending the first Information Request message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval. | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0 .. 100 | | |
| Default value | 1 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

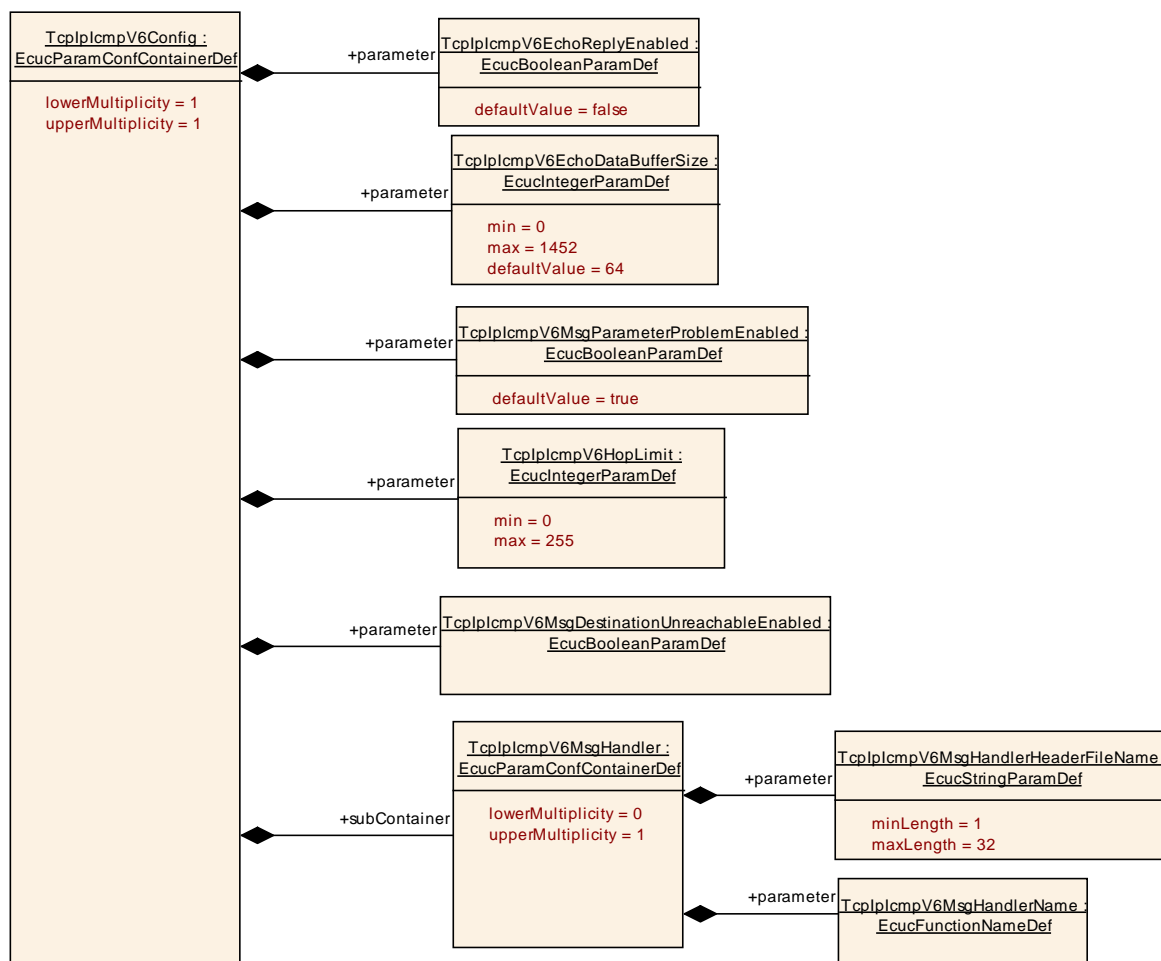
| | | | |
|---------------------|--|--|--|
| SWS Item | ECUC_Tcplp_00119 : | | |
| Name | TcplpDhcpV6InfDelayMin | | |
| Description | Minimum delay (s) before the first Information Request message will be sent. | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |

| | | | |
|----------------------------------|-------------------------|----|--------------|
| Range | 0 .. 100 | | |
| Default value | 0 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00120 : | | |
| Name | TcplpDhcpV6SolDelayMax | | |
| Description | Maximum delay (s) before sending the first Solicit message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval. | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0 .. 100 | | |
| Default value | 1 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00121 : | | |
| Name | TcplpDhcpV6SolDelayMin | | |
| Description | Minimum delay (s) before the first Solicit message will be sent. | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0 .. 100 | | |
| Default value | 0 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

No Included Containers



10.2.25 TcpIplcmpV6Config

| | | | |
|---------------------------------|---|--|--|
| SWS Item | ECUC_Tcplp_00113 : | | |
| Container Name | TcpIplcmpV6Config | | |
| Description | Specifies the configuration parameters of the ICMPv6 (Internet Control Message Protocol for IPv6) sub-module. | | |
| Configuration Parameters | | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00150 : | | |
| Name | TcpIplcmpV6EchoDataBufferSize | | |
| Description | Configure the maximum amount of arbitrary data that will be sent back to the sender of an echo request. If the arbitrary data of an incoming echo request is bigger than configured here, the response payload will be truncated. | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 1452 | | |
| Default value | 64 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00149 : | | |
| Name | TcplplcmpV6EchoReplyEnabled | | |
| Description | If enabled, the stack will respond to incoming ICMPv6 Echo Requests (Pings). | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | false | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00152 : | | |
| Name | TcplplcmpV6HopLimit | | |
| Description | Default Hop-Limit value of outgoing ICMPv6 packets. | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 255 | | |
| Default value | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00153 : | | |
| Name | TcplplcmpV6MsgDestinationUnreachableEnabled | | |
| Description | Dis/Enables transmission of Destination Unreachable Messages | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00151 : | | |
| Name | TcplplcmpV6MsgParameterProblemEnabled | | |
| Description | If enabled an ICMPv6 parameter problem message will be sent if a received packet has been dropped due to unknown options or headers that are found in the packet. [RFC2460 4. IPv6 Extension Headers] | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | true | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | |
|----------------------------|---------------------|---------------------------|
| Included Containers | | |
| Container Name | Multiplicity | Scope / Dependency |

| | | |
|-----------------------|------|--|
| TcplplcmpV6MsgHandler | 0..1 | This container is a subcontainer of TcplplcmpConfig and specifies the configuration parameters for the ICMPv6 message handler. |
|-----------------------|------|--|

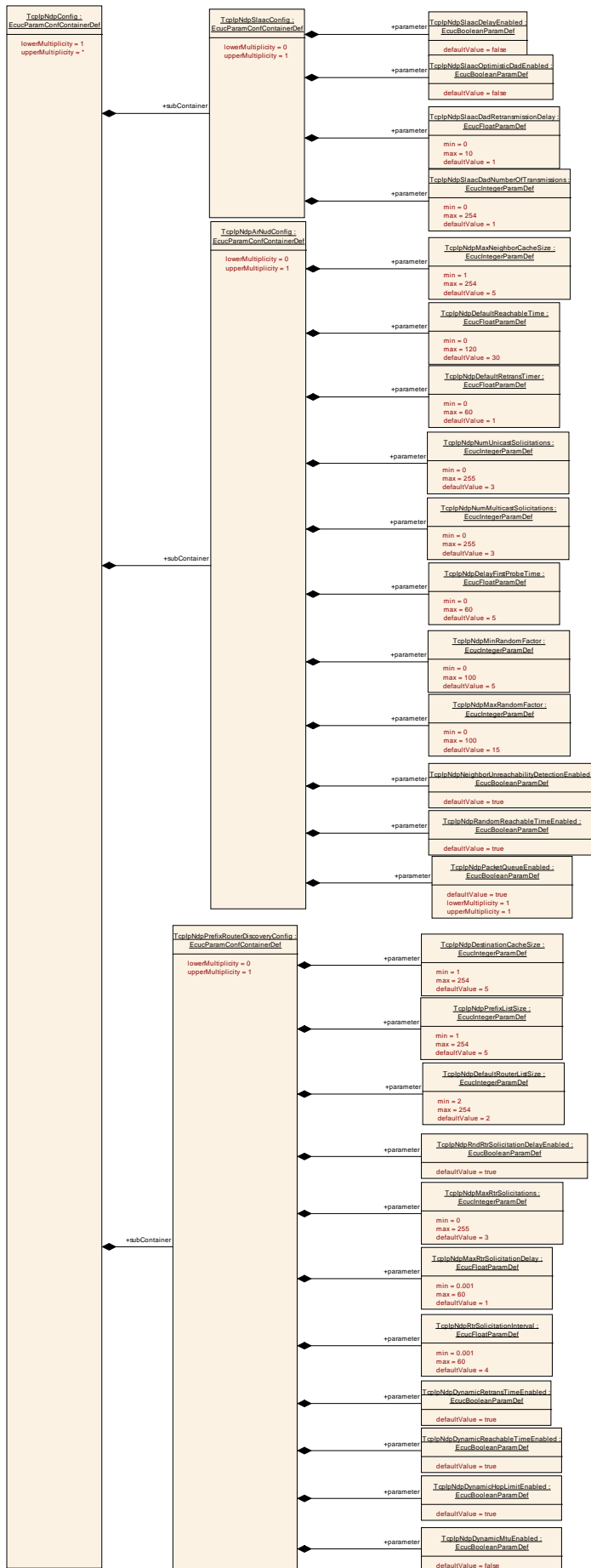
10.2.26 TcplplcmpV6MsgHandler

| | | | |
|---------------------------------|--|--|--|
| SWS Item | ECUC_Tcplp_00154 : | | |
| Container Name | TcplplcmpV6MsgHandler | | |
| Description | This container is a subcontainer of TcplplcmpConfig and specifies the configuration parameters for the ICMPv6 message handler. | | |
| Configuration Parameters | | | |

| | | | |
|----------------------------------|--|----|---------------------------------------|
| SWS Item | ECUC_Tcplp_00155 : | | |
| Name | TcplplcmpV6MsgHandlerHeaderFileName | | |
| Description | This parameter specifies the name of the header file containing the definition of the ICMPv6 message handler function. | | |
| Multiplicity | 1 | | |
| Type | EcucStringParamDef | | |
| Default value | -- | | |
| maxLength | 32 | | |
| minLength | 1 | | |
| regularExpression | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|---------------------------------------|
| SWS Item | ECUC_Tcplp_00156 : | | |
| Name | TcplplcmpV6MsgHandlerName | | |
| Description | This parameter defines the name of the ICMP message handler function <User_IcmpMsgHandler>. | | |
| Multiplicity | 1 | | |
| Type | EcucFunctionNameDef | | |
| Default value | -- | | |
| maxLength | -- | | |
| minLength | -- | | |
| regularExpression | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| |
|-------------------------------|
| No Included Containers |
|-------------------------------|



10.2.27 TcplpNdpConfig

| | |
|---------------------------------|---|
| SWS Item | ECUC_Tcplp_00112 : |
| Container Name | TcplpNdpConfig |
| Description | Specifies the configuration parameters of the Neighbor Discovery Protocol for IPv6 This container may be referenced by multiple IPv6 instances if they shall use the same configuration. This container may have multiple instances if different configurations are required for different IPv6 instances. |
| Configuration Parameters | |

| Included Containers | | |
|-------------------------------------|---------------------|--|
| Container Name | Multiplicity | Scope / Dependency |
| TcplpNdpArNudConfig | 0..1 | Specifies the configuration parameters for NDP Address Resolution and Neighbor Unreachability Detection. |
| TcplpNdpPrefixRouterDiscoveryConfig | 0..1 | Specifies the configuration parameters for NDP Prefix and Router Discovery. |
| TcplpNdpSlaacConfig | 0..1 | Specifies the configuration parameters for StateLess Address AutoConfiguration. |

10.2.28 TcplpNdpArNudConfig

| | |
|---------------------------------|--|
| SWS Item | ECUC_Tcplp_00123 : |
| Container Name | TcplpNdpArNudConfig |
| Description | Specifies the configuration parameters for NDP Address Resolution and Neighbor Unreachability Detection. |
| Configuration Parameters | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00130 : | | |
| Name | TcplpNdpDefaultReachableTime | | |
| Description | Configuration of the ReachableTime (s) specified in [RFC4861 6.3.2. Host Variables]. "The time a neighbor is considered reachable after receiving a reachability confirmation." If "TcplpNdpDynamicReachableTimeEnabled" is checked, this value may be reconfigured based on received Router Advertisements. Default: REACHABLE_TIME = 30 seconds | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0 .. 120 | | |
| Default value | 30 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|--------------------|--|--|--|
| SWS Item | ECUC_Tcplp_00165 : | | |
| Name | TcplpNdpDefaultRetransTimer | | |
| Description | Configures the default value (s) for the RetransTimer variable specified in [RFC4861 6.3.2. Host Variables]. "The time between retransmissions of Neighbor Solicitation messages to a neighbor when resolving the address or when probing the reachability of a | | |

| | | | |
|----------------------------------|---|----|--------------|
| | neighbor." If "TcplpNdpDynamicRetransTimeEnabled" is checked, this value may be reconfigured based on received Router Advertisements. Default: RETRANS_TIMER = 1 second | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0 .. 60 | | |
| Default value | 1 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00133 : | | |
| Name | TcplpNdpDelayFirstProbeTime | | |
| Description | Delay before sending the first NUD probe in (s). [RFC4861 7.3.3. Node Behavior] Default: DELAY_FIRST_PROBE_TIME = 5 seconds | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0 .. 60 | | |
| Default value | 5 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00129 : | | |
| Name | TcplpNdpMaxNeighborCacheSize | | |
| Description | Maximum number of entries in the neighbor cache. [RFC4861 5.1. Conceptual Data Structures] | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 1 .. 254 | | |
| Default value | 5 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00135 : | | |
| Name | TcplpNdpMaxRandomFactor | | |
| Description | Maximum random factor used for randomization [RFC4861 10. Protocol Constants] Default: 15 (MAX_RANDOM_FACTOR = 1.5) | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 100 | | |
| Default value | 15 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |

| | |
|---------------------------|--------------|
| Scope / Dependency | scope: local |
|---------------------------|--------------|

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00134 : | | |
| Name | TcplpNdpMinRandomFactor | | |
| Description | Minimum random factor used for randomization [RFC4861 10. Protocol Constants] Default: 5 (MIN_RANDOM_FACTOR = 0.5) | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 100 | | |
| Default value | 5 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00136 : | | |
| Name | TcplpNdpNeighborUnreachabilityDetectionEnabled | | |
| Description | Neighbor Unreachability Detection is used to remove unused entries from the neighbor cache. This feature is a basic feature of NDP and should be turned on. | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | true | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00132 : | | |
| Name | TcplpNdpNumMulticastSolicitations | | |
| Description | Maximum number of multicast solicitations that will be sent when performing address resolution. [RFC4861 7.2.2. Sending Neighbor Solicitations] Default: MAX_MULTICAST_SOLICIT = 3 | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 255 | | |
| Default value | 3 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|---------------------|--|--|--|
| SWS Item | ECUC_Tcplp_00131 : | | |
| Name | TcplpNdpNumUnicastSolicitations | | |
| Description | Maximum number of unicast solicitations that will be sent when performing Neighbor Unreachability Detection. [RFC4861 7.3.3. Node Behavior] Default: MAX_UNICAST_SOLICIT = 3 | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 255 | | |

| | | | |
|----------------------------------|-------------------------|----|--------------|
| Default value | 3 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00171 : | | |
| Name | TcplpNdpPacketQueueEnabled | | |
| Description | Enables (TRUE) or disables (FALSE) support of a NDP Packet Queue according to IETF RFC 4861, section 7.2.2. | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | true | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00137 : | | |
| Name | TcplpNdpRandomReachableTimeEnabled | | |
| Description | If enabled the value of ReachableTime will be multiplied with a random value between MIN_RANDOM_FACTOR and MAX_RANDOM_FACTOR in order to prevent multiple nodes from transmitting at exactly the same time [RFC4861 6.3.2. Host Variables / ReachableTime] | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | true | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

No Included Containers

10.2.29 TcplpNdpPrefixRouterDiscoveryConfig

| | | | |
|---------------------------------|---|--|--|
| SWS Item | ECUC_Tcplp_00124 : | | |
| Container Name | TcplpNdpPrefixRouterDiscoveryConfig | | |
| Description | Specifies the configuration parameters for NDP Prefix and Router Discovery. | | |
| Configuration Parameters | | | |

| | | | |
|---------------------|--|--|--|
| SWS Item | ECUC_Tcplp_00139 : | | |
| Name | TcplpNdpDefaultRouterListSize | | |
| Description | Maximum number of default router entries. [RFC4861 5.1. Conceptual Data Structures] | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 2 .. 254 | | |

| | | | |
|----------------------------------|-------------------------|----|--------------|
| Default value | 2 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00138 : | | |
| Name | TcplpNdpDestinationCacheSize | | |
| Description | Maximum number of entries in the destination cache. [RFC4861 5.1. Conceptual Data Structures] | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 1 .. 254 | | |
| Default value | 5 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00147 : | | |
| Name | TcplpNdpDynamicHopLimitEnabled | | |
| Description | If enabled the default hop limit may be reconfigured based on received Router Advertisements. [RFC4861 6.3.4. Processing Received Router Advertisements] | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | true | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00148 : | | |
| Name | TcplpNdpDynamicMtuEnabled | | |
| Description | Allow dynamic reconfiguration of link MTU via Router Advertisements. [RFC4861 4.6.4. MTU] | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | false | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|---------------------|--|--|--|
| SWS Item | ECUC_Tcplp_00146 : | | |
| Name | TcplpNdpDynamicReachableTimeEnabled | | |
| Description | If enabled the default Reachable Time value may be reconfigured based on received Router Advertisements. [RFC4861 6.3.4. Processing Received Router Advertisements] Default: Enabled | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |

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

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00145 : | | |
| Name | TcplpNdpDynamicRetransTimeEnabled | | |
| Description | If enabled the default Retransmit Timer value may be reconfigured based on received Router Advertisements. [RFC4861 6.3.4. Processing Received Router Advertisements] Default: Enabled | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | true | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00143 : | | |
| Name | TcplpNdpMaxRtrSolicitationDelay | | |
| Description | Maximum delay before the first Router Solicitation will be sent after interface initialization in (s). [RFC4861 6.3.7. Sending Router Solicitations] Default: MAX_RTR_SOLICITATION_DELAY = 1 second | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0.001 .. 60 | | |
| Default value | 1 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00142 : | | |
| Name | TcplpNdpMaxRtrSolicitations | | |
| Description | Maximum number of Router Solicitations that will be sent before the first Router Advertisement has been received. 0 = No Router Solicitations will be sent. This has no impact on handling Router Advertisements. [RFC4861 6.3.7. Sending Router Solicitations] Default: MAX_RTR_SOLICITATIONS = 3 transmissions | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 255 | | |
| Default value | 3 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00140 : | | |
| Name | TcplpNdpPrefixListSize | | |
| Description | Maximum number of entries in the on-link prefix list. [RFC4861 5.1. Conceptual Data Structures] | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 1 .. 254 | | |
| Default value | 5 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00141 : | | |
| Name | TcplpNdpRndRtrSolicitationDelayEnabled | | |
| Description | If enabled the first router solicitation will be delayed randomly from [0...MAX_RTR_SOLICITATION_DELAY]. Otherwise the first router solicitation will be sent after exactly MAX_RTR_SOLICITATION_DELAY milliseconds. [RFC4861 6.3.7. Sending Router Solicitations] Default: Enabled | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | true | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00144 : | | |
| Name | TcplpNdpRtrSolicitationInterval | | |
| Description | Interval between consecutive Router Solicitations in (s). [RFC4861 6.3.7. Sending Router Solicitations] Default: RTR_SOLICITATION_INTERVAL = 4 seconds | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0.001 .. 60 | | |
| Default value | 4 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

No Included Containers

10.2.30 TcplpNdpSlaacConfig

| | | | |
|-----------------------|--|--|--|
| SWS Item | ECUC_Tcplp_00122 : | | |
| Container Name | TcplpNdpSlaacConfig | | |
| Description | Specifies the configuration parameters for StateLess Address | | |

| | |
|---------------------------------|--------------------|
| | AutoConfiguration. |
| Configuration Parameters | |

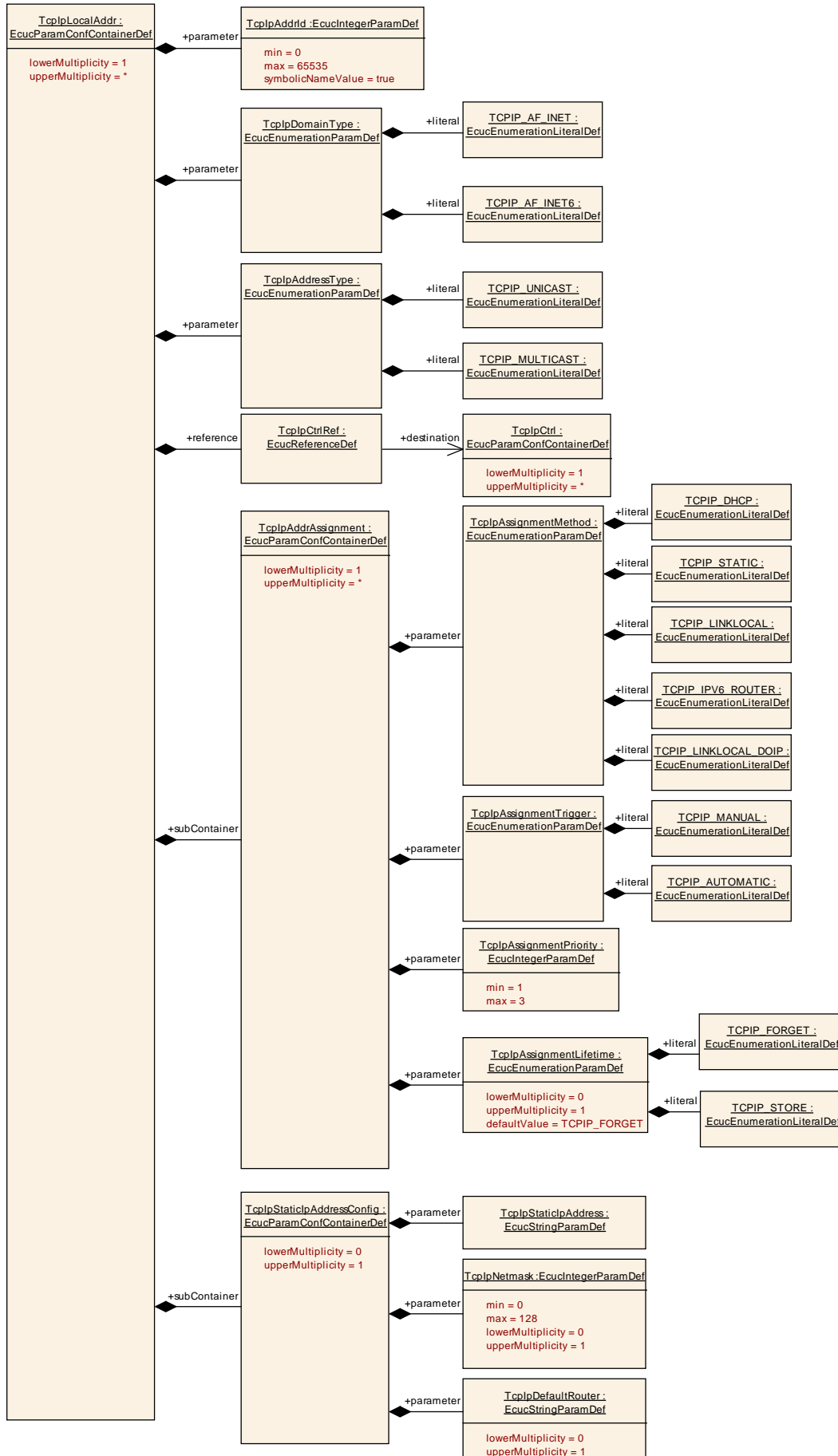
| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00128 : | | |
| Name | TcplpNdpSlaacDadNumberOfTransmissions | | |
| Description | Number of Neighbor Solicitations that have to be unanswered in order to set an autoconfigured address to PREFERRED (usable) state. [RFC4861 5.1. Node Configuration Variables] Default: DupAddrDetectTransmits = 1 Setting this value to 0 turns off DAD. | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 254 | | |
| Default value | 1 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00127 : | | |
| Name | TcplpNdpSlaacDadRetransmissionDelay | | |
| Description | Sets the maximum value for the address configuration delay (s). According to [RFC4861 5.4.2. Sending Neighbor Solicitation Messages] this value should be the same as MAX_RTR_SOLICITATION_DELAY. Default: MAX_RTR_SOLICITATION_DELAY = 1 second | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0 .. 10 | | |
| Default value | 1 | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00125 : | | |
| Name | TcplpNdpSlaacDelayEnabled | | |
| Description | If enabled transmission of the first DAD Neighbor Solicitation will be delayed by a random value from [0...MAX_DAD_DELAY]. "This serves to alleviate congestion when many nodes start up on the link at the same time, such as after a power failure, and may help to avoid race conditions when more than one node is trying to solicit for the same address at the same time." "The delay will avoid similar congestion when multiple nodes are going to configure addresses by receiving the same single multicast router advertisement." [RFC4861 5.4.2. Sending Neighbor Solicitation Messages] Default: True | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | false | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00126 : | | |
| Name | TcplpNdpSlaacOptimisticDadEnabled | | |
| Description | Enable Optimistic Duplicate Address Detection (DAD) according to RFC4429. | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | false | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

No Included Containers



10.2.31 TcplpLocalAddr

| | |
|---------------------------------|---|
| SWS Item | ECUC_Tcplp_00020 : |
| Container Name | TcplpLocalAddr |
| Description | Specifies the local IP (Internet Protocol) addresses used for IP communication. |
| Configuration Parameters | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00029 : | | |
| Name | TcplpAddrId | | |
| Description | IP address table identifier assigned by TCP/IP stack. | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef (Symbolic Name generated for this parameter) | | |
| Range | 0 .. 65535 | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: ECU | | |

| | | | |
|----------------------------------|---------------------------|---|---------------------|
| SWS Item | ECUC_Tcplp_00031 : | | |
| Name | TcplpAddressType | | |
| Description | Address type. | | |
| Multiplicity | 1 | | |
| Type | EcucEnumerationParamDef | | |
| Range | TCPIP_MULTICAST | | Multicast address. |
| | TCPIP_UNICAST | | Unicast address |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---------------------------|---|---------------------|
| SWS Item | ECUC_Tcplp_00030 : | | |
| Name | TcplpDomainType | | |
| Description | Address family. | | |
| Multiplicity | 1 | | |
| Type | EcucEnumerationParamDef | | |
| Range | TCPIP_AF_INET | | IPv4 address |
| | TCPIP_AF_INET6 | | IPv6 address |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|--------------------|---|--|--|
| SWS Item | ECUC_Tcplp_00032 : | | |
| Name | TcplpCtrlRef | | |
| Description | Reference to a TcplpCtrl specifying the EthIf Controller where the IP address shall be assigned and DEM errors that shall be reported in case of an error on this controller. | | |

| | | | |
|----------------------------------|----------------------------|---|---------------------|
| Multiplicity | 1 | | |
| Type | Reference to [TcplpCtrl] | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| Included Containers | | |
|----------------------------|---------------------|---|
| Container Name | Multiplicity | Scope / Dependency |
| TcplpAddrAssignment | 1..* | This container is a subcontainer of TcplpLocalAddr and specifies the assignment policy for the IP address. |
| TcplpStaticIpAddressConfig | 0..1 | This container is a subcontainer of TcplpLocalAddr and specifies a static IP address including directly related parameters. |

10.2.32 TcplpAddrAssignment

| | |
|---------------------------------|--|
| SWS Item | ECUC_Tcplp_00033 : |
| Container Name | TcplpAddrAssignment |
| Description | This container is a subcontainer of TcplpLocalAddr and specifies the assignment policy for the IP address. |
| Configuration Parameters | |

| | | | |
|----------------------------------|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00186 : | | |
| Name | TcplpAssignmentLifetime | | |
| Description | Defines the lifetime of a dynamically fetched IP address. If TcplpAssignmentMethod = TCPIP_STATIC then TcplpAssignmentLifetime shall be omitted. | | |
| Multiplicity | 0..1 | | |
| Type | EcucEnumerationParamDef | | |
| Range | TCPIP_FORGET | After a dynamic IP address has been assigned just use it for this link-up time. | |
| | TCPIP_STORE | After a dynamic IP address has been assigned store the address persistently. | |
| Default value | TCPIP_FORGET | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | |
|---------------------|------------------------------|---|
| SWS Item | ECUC_Tcplp_00035 : | |
| Name | TcplpAssignmentMethod | |
| Description | Method of address assignment | |
| Multiplicity | 1 | |
| Type | EcucEnumerationParamDef | |
| Range | TCPIP_DHCP | Dynamic Assigned IP Address using DHCP |
| | TCPIP_IPV6_ROUTER | Dynamic Configured IPv6 Address by Router Advertisement |

| | | | |
|----------------------------------|-------------------------|--|---------------------|
| | TCPIP_LINKLOCAL | Linklocal IPv4/IPv6 Address Assignment | |
| | TCPIP_LINKLOCAL_DOIP | Linklocal IPv4/IPv6 Address Assignment using DoIP Parameters | |
| | TCPIP_STATIC | Static Assigned IP Address | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|---|---------------------|
| SWS Item | ECUC_Tcplp_00037 : | | |
| Name | TcplpAssignmentPriority | | |
| Description | Priority of assignment (1 is highest). If a new address from an assignment method with a higher priority is available, it overwrites the IP address previously assigned by an assignment method with a lower priority. | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 1 .. 3 | | |
| Default value | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--------------------------------|---|---------------------|
| SWS Item | ECUC_Tcplp_00036 : | | |
| Name | TcplpAssignmentTrigger | | |
| Description | Trigger of address assignment. | | |
| Multiplicity | 1 | | |
| Type | EcucEnumerationParamDef | | |
| Range | TCPIP_AUTOMATIC | Assignment shall be initiated automatically by TCP/IP stack. | |
| | TCPIP_MANUAL | Assignment shall be initiated manually via Tcplp_RequestIpAddrAssignment(). | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

No Included Containers

10.2.33 TcplpStaticIpAddressConfig

| | | | |
|-----------------------|---|--|--|
| SWS Item | ECUC_Tcplp_00034 : | | |
| Container Name | TcplpStaticIpAddressConfig | | |
| Description | This container is a subcontainer of TcplpLocalAddr and specifies a static IP address including directly related parameters. | | |

Configuration Parameters

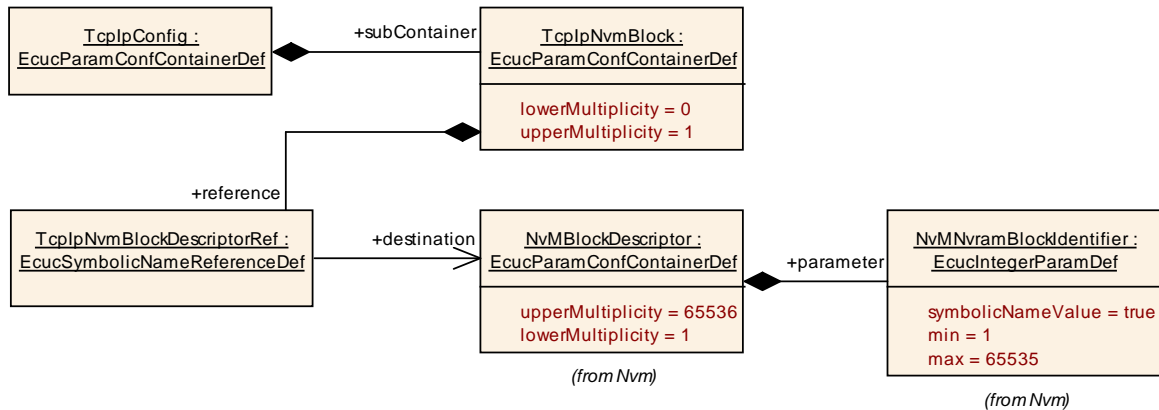
| SWS Item | | ECUC_Tcplp_00040 : | |
|----------------------------------|--|--------------------|---------------------|
| Name | TcplpDefaultRouter | | |
| Description | IP address of default router (gateway) | | |
| Multiplicity | 0..1 | | |
| Type | EcucStringParamDef | | |
| Default value | -- | | |
| maxLength | -- | | |
| minLength | -- | | |
| regularExpression | -- | | |
| Post-Build Variant Multiplicity | true | | |
| Post-Build Variant Value | true | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| SWS Item | | ECUC_Tcplp_00039 : | |
|----------------------------------|---|--------------------|---------------------|
| Name | TcplpNetmask | | |
| Description | Network mask of IPv4 address or address prefix of IPv6 address in CIDR Notation, i.e. decimal value between 0 and 32 (IPv4) or 0 and 128 (IPv6) that describes the number of significant bits defining the network number or prefix of an IP address. | | |
| Multiplicity | 0..1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 128 | | |
| Default value | -- | | |
| Post-Build Variant Multiplicity | true | | |
| Post-Build Variant Value | true | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| SWS Item | | ECUC_Tcplp_00038 : | |
|---------------------------|--|--------------------|---------------------|
| Name | TcplpStaticIpAddress | | |
| Description | Static IP Address. To specify any IP address for a certain EthIfCtrl, "ANY" has to be set as wildcard. See Tcplp_Bind() for more details. | | |
| Multiplicity | 1 | | |
| Type | EcucStringParamDef | | |
| Default value | -- | | |
| maxLength | -- | | |
| minLength | -- | | |
| regularExpression | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |

| | | | |
|---------------------------|------------------------|---|--------------------|
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

No Included Containers

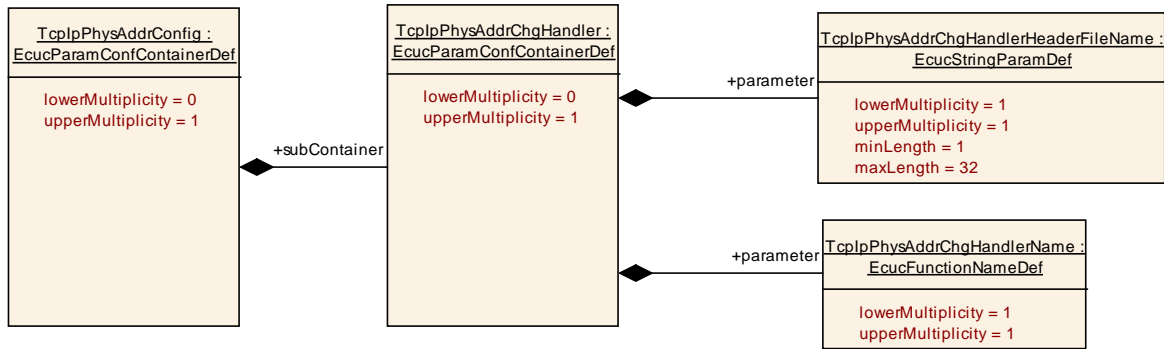


10.2.34 TcpIpNvmBlock

| | | | |
|---|---|----|---------------------------------------|
| SWS Item | ECUC_Tcplp_00184 : | | |
| Container Name | TcpIpNvmBlock | | |
| Description | Configuration of optional usage of Nvm in case the TcpIp module requires non volatile memory in the Ecu to store information (e.g. IP Address received via DHCP and shall be stored). | | |
| Multiplicity Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Configuration Parameters | | | |

| | | | |
|----------------------------------|---|----|---------------------------------------|
| SWS Item | ECUC_Tcplp_00185 : | | |
| Name | TcpIpNvmBlockDescriptorRef | | |
| Description | Reference to the Nvm block description in the Nvm module configuration. | | |
| Multiplicity | 1 | | |
| Type | Symbolic name reference to [NvMBlockDescriptor] | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: ECU | | |

No Included Containers



10.2.35 TcpIpPhysAddrConfig

| | |
|---------------------------------|---|
| SWS Item | ECUC_Tcplp_00083 : |
| Container Name | TcplpPhysAddrConfig |
| Description | Specifies the physical address configuration. |
| Configuration Parameters | |

| Included Containers | | |
|----------------------------|---------------------|---|
| Container Name | Multiplicity | Scope / Dependency |
| TcplpPhysAddrChgHandler | 0..1 | This container is a subcontainer of TcplpPhysAddrConfig and specifies the configuration parameters for physical address change handler. |

10.2.36 TcpIpPhysAddrChgHandler

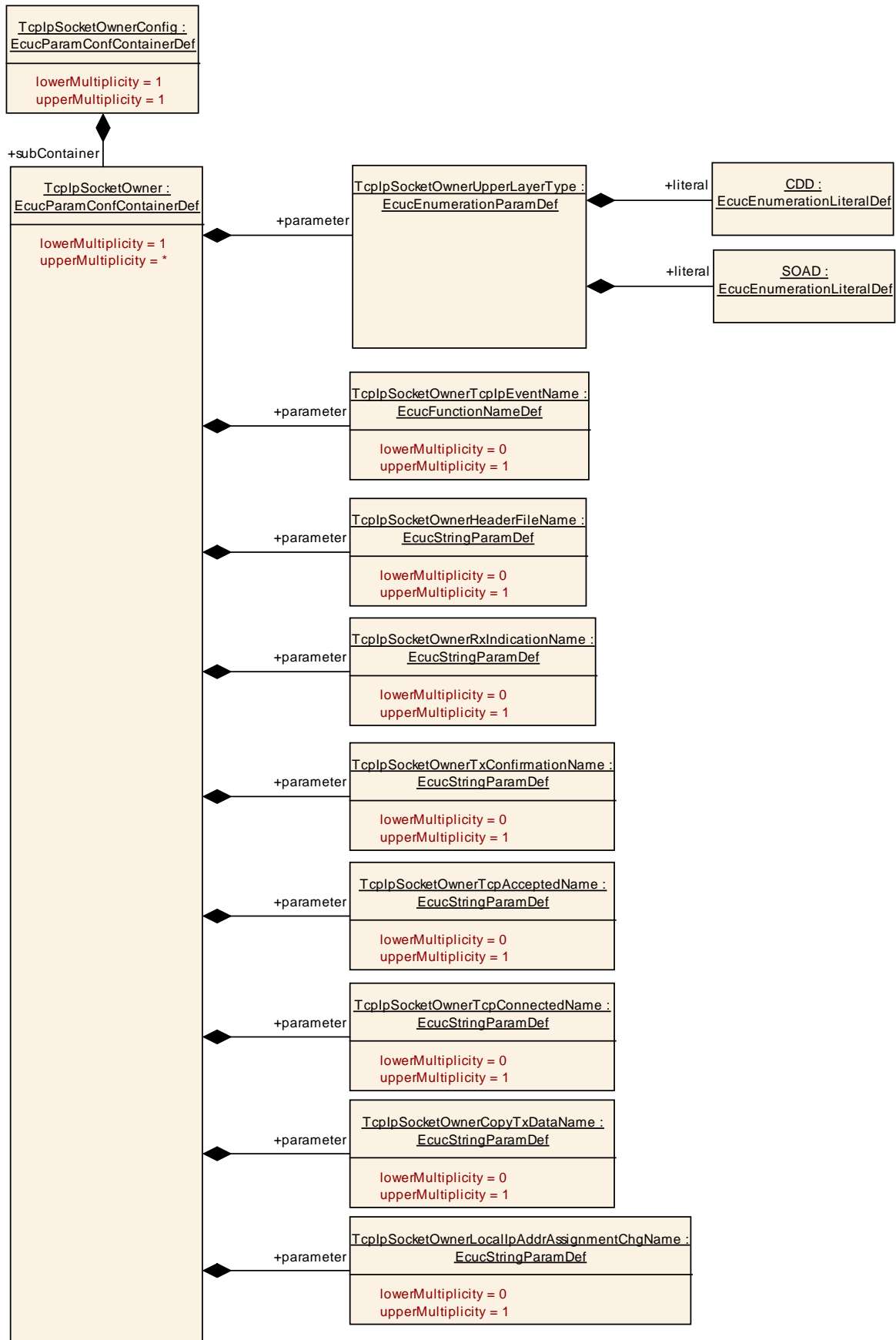
| | |
|---------------------------------|---|
| SWS Item | ECUC_Tcplp_00084 : |
| Container Name | TcplpPhysAddrChgHandler |
| Description | This container is a subcontainer of TcplpPhysAddrConfig and specifies the configuration parameters for physical address change handler. |
| Configuration Parameters | |

| | | | |
|----------------------------------|---|----|---------------------------------------|
| SWS Item | ECUC_Tcplp_00085 : | | |
| Name | TcplpPhysAddrChgHandlerHeaderFileName | | |
| Description | This parameter specifies the name of the header file containing the definition of the physical address change handler function. | | |
| Multiplicity | 1 | | |
| Type | EcucStringParamDef | | |
| Default value | -- | | |
| maxLength | 32 | | |
| minLength | 1 | | |
| regularExpression | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | |
|-----------------|---------------------------|
| SWS Item | ECUC_Tcplp_00086 : |
|-----------------|---------------------------|

| | | | |
|----------------------------------|--|----|--------------|
| Name | TcplpPhysAddrChgHandlerName | | |
| Description | This parameter defines the name of the physical address change function <Up>_PhysAddrTableChg. | | |
| Multiplicity | 1 | | |
| Type | EcucFunctionNameDef | | |
| Default value | -- | | |
| maxLength | -- | | |
| minLength | -- | | |
| regularExpression | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: ECU | | |

No Included Containers



10.2.37 TcplpSocketOwnerConfig

| | |
|---------------------------------|--|
| SWS Item | ECUC_Tcplp_00172 : |
| Container Name | TcplpSocketOwnerConfig |
| Description | Specifies the upper layer modules of Tcplp using the socket API. |
| Configuration Parameters | |

| Included Containers | | |
|----------------------------|---------------------|--|
| Container Name | Multiplicity | Scope / Dependency |
| TcplpSocketOwner | 1..* | This container is a subcontainer of TcplpSocketOwnerConfig and specifies an upper layer of Tcplp that uses the socket API. |

10.2.38 TcplpSocketOwner

| | |
|---------------------------------|--|
| SWS Item | ECUC_Tcplp_00173 : |
| Container Name | TcplpSocketOwner |
| Description | This container is a subcontainer of TcplpSocketOwnerConfig and specifies an upper layer of Tcplp that uses the socket API. |
| Configuration Parameters | |

| | | | |
|----------------------------------|---|----|---------------------------------------|
| SWS Item | ECUC_Tcplp_00180 : | | |
| Name | TcplpSocketOwnerCopyTxDataName | | |
| Description | This parameter defines the name of the <Up_CopyTxData> function of the TcplpSocketOwner module. The function name shall only be configurable if TcplpSocketOwnerUpperLayerType is set to CDD. | | |
| Multiplicity | 0..1 | | |
| Type | EcucStringParamDef | | |
| Default value | -- | | |
| maxLength | -- | | |
| minLength | -- | | |
| regularExpression | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: local dependency: TcplpSocketOwnerUpperLayerType | | |

| | | | |
|----------------------------------|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00175 : | | |
| Name | TcplpSocketOwnerHeaderFileName | | |
| Description | This parameter specifies the name of the header file containing the definition of the TcplpSocketOwner module functions. The header file name shall only be configurable if TcplpSocketOwnerUpperLayerType is set to CDD. | | |
| Multiplicity | 0..1 | | |
| Type | EcucStringParamDef | | |
| Default value | -- | | |
| maxLength | -- | | |
| minLength | -- | | |
| regularExpression | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |

| | | | |
|---------------------------|--|----|---------------------------------------|
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: local dependency: TcplpSocketOwnerUpperLayerType | | |

| | | | |
|----------------------------------|---|----|---------------------------------------|
| SWS Item | ECUC_Tcplp_00181 : | | |
| Name | TcplpSocketOwnerLocalIpAddrAssignmentChgName | | |
| Description | This parameter defines the name of the <Up_LocalIpAddrAssignmentChg> function of the TcplpSocketOwner module. The function name shall only be configurable if TcplpSocketOwnerUpperLayerType is set to CDD. | | |
| Multiplicity | 0..1 | | |
| Type | EcucStringParamDef | | |
| Default value | -- | | |
| maxLength | -- | | |
| minLength | -- | | |
| regularExpression | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: local dependency: TcplpSocketOwnerUpperLayerType | | |

| | | | |
|----------------------------------|---|----|---------------------------------------|
| SWS Item | ECUC_Tcplp_00176 : | | |
| Name | TcplpSocketOwnerRxIndicationName | | |
| Description | This parameter defines the name of the <Up_RxIndication> function of the TcplpSocketOwner module. The function name shall only be configurable if TcplpSocketOwnerUpperLayerType is set to CDD. | | |
| Multiplicity | 0..1 | | |
| Type | EcucStringParamDef | | |
| Default value | -- | | |
| maxLength | -- | | |
| minLength | -- | | |
| regularExpression | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: local dependency: TcplpSocketOwnerUpperLayerType | | |

| | | | |
|---------------------------------|---|--|--|
| SWS Item | ECUC_Tcplp_00178 : | | |
| Name | TcplpSocketOwnerTcpcAcceptedName | | |
| Description | This parameter defines the name of the <Up_TcpcAccepted> function of the TcplpSocketOwner module. The function name shall only be configurable if TcplpSocketOwnerUpperLayerType is set to CDD. | | |
| Multiplicity | 0..1 | | |
| Type | EcucStringParamDef | | |
| Default value | -- | | |
| maxLength | -- | | |
| minLength | -- | | |
| regularExpression | -- | | |
| Post-Build Variant Value | false | | |

| | | | |
|----------------------------------|--|----|---------------------------------------|
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: local dependency: TcplpSocketOwnerUpperLayerType | | |

| | | | |
|----------------------------------|---|----|---------------------------------------|
| SWS Item | ECUC_Tcplp_00179 : | | |
| Name | TcplpSocketOwnerTcpConnectedName | | |
| Description | This parameter defines the name of the <Up_TcpConnected> function of the TcplpSocketOwner module. The function name shall only be configurable if TcplpSocketOwnerUpperLayerType is set to CDD. | | |
| Multiplicity | 0..1 | | |
| Type | EcucStringParamDef | | |
| Default value | -- | | |
| maxLength | -- | | |
| minLength | -- | | |
| regularExpression | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: local dependency: TcplpSocketOwnerUpperLayerType | | |

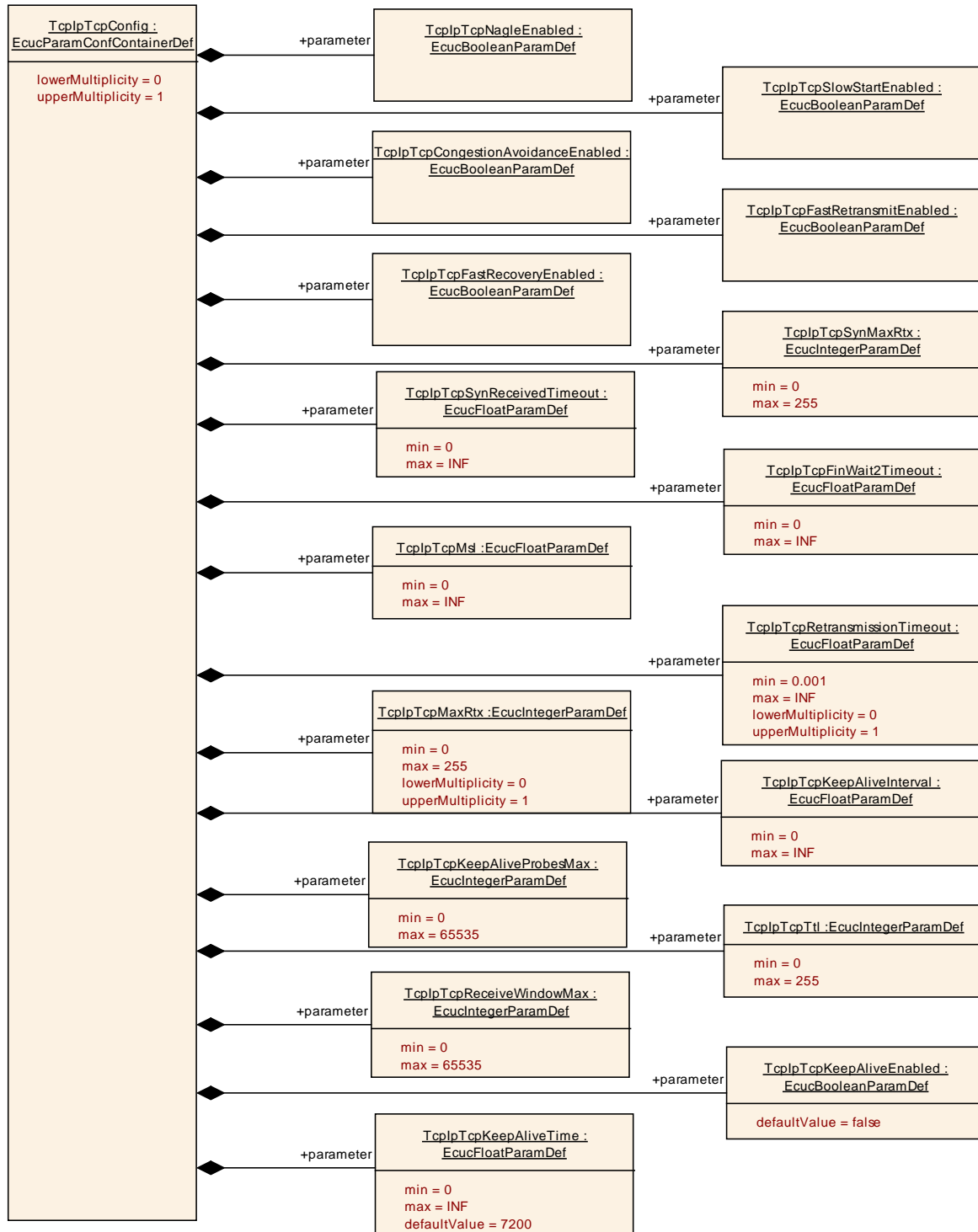
| | | | |
|----------------------------------|---|----|---------------------------------------|
| SWS Item | ECUC_Tcplp_00197 : | | |
| Name | TcplpSocketOwnerTcplpEventName | | |
| Description | This parameter defines the name of the <Up_TcplpEvent> function of the TcplpSocketOwner module. The function name shall only be configurable if TcplpSocketOwnerUpperLayerType is set to CDD. | | |
| Multiplicity | 0..1 | | |
| Type | EcucFunctionNameDef | | |
| Default value | -- | | |
| maxLength | -- | | |
| minLength | -- | | |
| regularExpression | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: local dependency: TcplpSocketOwnerUpperLayerType | | |

| | | | |
|--------------------------|---|--|--|
| SWS Item | ECUC_Tcplp_00177 : | | |
| Name | TcplpSocketOwnerTxConfirmationName | | |
| Description | This parameter defines the name of the <Up_TxConfirmation> function of the TcplpSocketOwner module. The function name shall only be configurable if TcplpSocketOwnerUpperLayerType is set to CDD. | | |
| Multiplicity | 0..1 | | |
| Type | EcucStringParamDef | | |
| Default value | -- | | |
| maxLength | -- | | |
| minLength | -- | | |
| regularExpression | -- | | |

| | | | |
|----------------------------------|--|----|---------------------------------------|
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | Post-build time | -- | |
| Scope / Dependency | scope: local dependency: TcpIpSocketOwnerUpperLayerType | | |

| | | | |
|----------------------------------|--|----------------|---------------------|
| SWS Item | ECUC_Tcplp_00174 : | | |
| Name | TcpIpSocketOwnerUpperLayerType | | |
| Description | This parameter specifies the type of the upper layer module. | | |
| Multiplicity | 1 | | |
| Type | EcucEnumerationParamDef | | |
| Range | CDD | Complex Driver | |
| | SOAD | Socket Adaptor | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| |
|-------------------------------|
| No Included Containers |
|-------------------------------|



10.2.39 TcpIpTcpConfig

| | |
|---------------------------------|---|
| SWS Item | ECUC_Tcplp_00025 : |
| Container Name | TcplpTcpConfig |
| Description | Specifies the configuration parameters of the TCP (Transmission Control Protocol) sub-module. |
| Configuration Parameters | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00061 : | | |
| Name | TcplpTcpCongestionAvoidanceEnabled | | |
| Description | Enables (TRUE) or disables (FALSE) support of TCP congestion avoidance algorithm according to IETF RFC 5681. | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00063 : | | |
| Name | TcplpTcpFastRecoveryEnabled | | |
| Description | Enables (TRUE) or disables (FALSE) support of TCP Fast Recovery according to IETF RFC 5681. | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00062 : | | |
| Name | TcplpTcpFastRetransmitEnabled | | |
| Description | Enables (TRUE) or disables (FALSE) support of TCP Fast Retransmission according to IETF RFC 5681. | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00066 : | | |
| Name | TcplpTcpFinWait2Timeout | | |
| Description | Timeout in [s] to receive a FIN from the remote node (after this node has initiated connection termination), i.e. maximum time waiting in FINWAIT-2 for a connection termination request from the remote TCP. | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0 .. INF | | |
| Default value | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|-----------------|---------------------------|--|--|
| SWS Item | ECUC_Tcplp_00082 : | | |
| Name | TcplpTcpKeepAliveEnabled | | |

| | | | |
|----------------------------------|---|----|--------------|
| Description | Enables (TRUE) or disables (FALSE) TCP Keep Alive Probes according to IETF RFC 1122 chapter 4.2.3.6 | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | false | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|---|---------------------|
| SWS Item | ECUC_Tcplp_00070 : | | |
| Name | TcplpTcpKeepAliveInterval | | |
| Description | Specifies the interval in [s] between subsequent keepalive probes. | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0 .. INF | | |
| Default value | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local dependency: TcplpTcpKeepAliveEnabled | | |

| | | | |
|----------------------------------|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00071 : | | |
| Name | TcplpTcpKeepAliveProbesMax | | |
| Description | Maximum number of times that a TCP Keep Alive is retransmitted before the connection is closed. | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 65535 | | |
| Default value | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local dependency: TcplpTcpKeepAliveEnabled | | |

| | | | |
|----------------------------------|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00087 : | | |
| Name | TcplpTcpKeepAliveTime | | |
| Description | Specifies the time in [s] between the last data packet sent (simple ACKs are not considered data) and the first keepalive probe. Note: Setting this configuration parameter to a value smaller or equal to the value of TcplpMainFunctionPeriod results in the transmission of keep alive probes within every MainFunction cycle. | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0 .. INF | | |
| Default value | 7200 | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | |
|--|--------------------------------------|
| | dependency: TcplpTcpKeepAliveEnabled |
|--|--------------------------------------|

| | | | |
|----------------------------------|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00069 : | | |
| Name | TcplpTcpMaxRtx | | |
| Description | Maximum number of times that a TCP segment is retransmitted before the TCP connection is closed. This parameter is only valid if TcplpTcpRetransmissionTimeout is configured. Note: This parameter also applies for FIN retransmissions. | | |
| Multiplicity | 0..1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 255 | | |
| Default value | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | dependency: TcplpTcpRetransmissionTimeout | | |

| | | | |
|----------------------------------|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00067 : | | |
| Name | TcplpTcpMsl | | |
| Description | Maximum segment lifetime in [s]. (Note: TIME-WAIT = 2 x TcplpTcpMsl " to ensure that the remote node received the acknowledgment to its connection termination request.) | | |
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0 .. INF | | |
| Default value | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|----|--------------|
| SWS Item | ECUC_Tcplp_00059 : | | |
| Name | TcplpTcpNagleEnabled | | |
| Description | Enables (TRUE) or disables (FALSE) support of Nagle's algorithm according to IETF RFC 896. If enabled the Nagle's algorithm is activated per default for all TCP sockets, but can be deactivated via Tcplp_ChangeParameter() API. | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

| | | | |
|---------------------------------|---|--|--|
| SWS Item | ECUC_Tcplp_00073 : | | |
| Name | TcplpTcpReceiveWindowMax | | |
| Description | Default value of maximum receive window in bytes. | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 65535 | | |
| Default value | -- | | |
| Post-Build Variant Value | true | | |

| | | | |
|----------------------------------|-------------------------|---|---------------------|
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|---|---------------------|
| SWS Item | ECUC_Tcplp_00068 : | | |
| Name | TcplpTcpRetransmissionTimeout | | |
| Description | Timeout in [s] before an unacknowledged TCP segment is sent again. If the timeout is disabled or set to INF, no TCP segments shall be retransmitted. | | |
| Multiplicity | 0..1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0.001 .. INF | | |
| Default value | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|--|----|--------------|
| SWS Item | ECUC_Tcplp_00060 : | | |
| Name | TcplpTcpSlowStartEnabled | | |
| Description | Enables (TRUE) or disables (FALSE) support of TCP slow start algorithm according to IETF RFC 5681. | | |
| Multiplicity | 1 | | |
| Type | EcucBooleanParamDef | | |
| Default value | -- | | |
| Post-Build Variant Value | false | | |
| Value Configuration Class | Pre-compile time | X | All Variants |
| | Link time | -- | |
| | Post-build time | -- | |
| Scope / Dependency | scope: local | | |

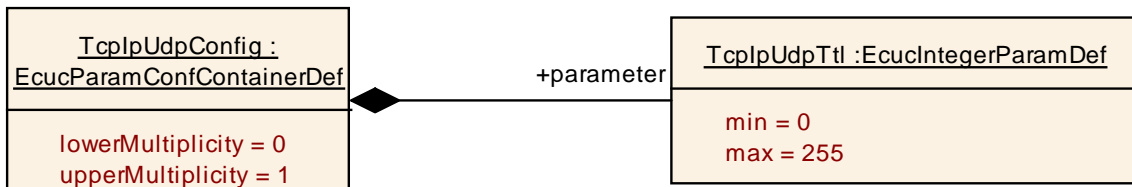
| | | | |
|----------------------------------|--|---|---------------------|
| SWS Item | ECUC_Tcplp_00064 : | | |
| Name | TcplpTcpSynMaxRtx | | |
| Description | Maximum number of times that a TCP SYN is retransmitted. Note: SYN will be retried after TcplpTcpRetransmissionTimeout. The connection will be dropped if no matching connection request has been received after the last TCP SYN has been sent and TcplpTcpRetransmissionTimeout has been expired. | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 255 | | |
| Default value | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|--------------------|--|--|--|
| SWS Item | ECUC_Tcplp_00065 : | | |
| Name | TcplpTcpSynReceivedTimeout | | |
| Description | Timeout in [s] to complete a remotely initiated TCP connection establishment, i.e. maximum time waiting in SYN-RECEIVED for a confirming connection request acknowledgment after having both received and sent a connection request. | | |

| | | | |
|----------------------------------|-------------------------|---|---------------------|
| Multiplicity | 1 | | |
| Type | EcucFloatParamDef | | |
| Range | 0 .. INF | | |
| Default value | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

| | | | |
|----------------------------------|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00072 : | | |
| Name | TcplpTcpttl | | |
| Description | Default Time-to-live value of outgoing TCP packets. | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 255 | | |
| Default value | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

No Included Containers



10.2.40 TcplpUdpConfig

| | | | |
|---------------------------------|---|--|--|
| SWS Item | ECUC_Tcplp_00026 : | | |
| Container Name | TcplpUdpConfig | | |
| Description | Specifies the configuration parameters of the UDP (User Datagram Protocol) sub-module | | |
| Configuration Parameters | | | |

| | | | |
|----------------------------------|---|---|---------------------|
| SWS Item | ECUC_Tcplp_00075 : | | |
| Name | TcplpUdpTtl | | |
| Description | Default Time-to-live value of outgoing UDP packets. | | |
| Multiplicity | 1 | | |
| Type | EcucIntegerParamDef | | |
| Range | 0 .. 255 | | |
| Default value | -- | | |
| Post-Build Variant Value | true | | |
| Value Configuration Class | Pre-compile time | X | VARIANT-PRE-COMPILE |
| | Link time | X | VARIANT-LINK-TIME |
| | Post-build time | X | VARIANT-POST-BUILD |
| Scope / Dependency | scope: local | | |

No Included Containers

10.3 Published Information

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