

<b>Document Title</b>	Specification of TCP/IP Stack
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	617

<b>Document Status</b>	published
<b>Part of AUTOSAR Standard</b>	Classic Platform
<b>Part of Standard Release</b>	R24-11

Document Change History			
Date	Release	Changed by	Description
2024-11-27	R24-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>Minor corrections and clarifications</li> <li>Error reporting scenarios update in TcpIp</li> <li>DoIP - Trace links from AP to CP</li> <li>Extension of Security Events for TcpIp</li> <li>Description update of TCP retransmissions handling</li> <li>Interpretation update of <a href="#">[SWS_TcpIp_00077]</a></li> <li>Adaptations for LSduR as lower layer</li> </ul>
2023-11-23	R23-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>Minor corrections and clarifications</li> <li>Editorial changes</li> <li>Wildcards descriptions improvement</li> <li>Introduction of IND, TCP window scale option, TCP SACK and TLS updates for Charging Interface</li> </ul>
2022-11-24	R22-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>ARP defensive processing</li> <li>Transition OFFLINE to ONLINE</li> </ul>
2021-11-25	R21-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>Add <a href="#">TcpIp_IsConnectionReady()</a></li> <li>Minor corrections and clarifications</li> <li>Editorial changes</li> </ul>





2020-11-30	R20-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Introduction of IdsM (DRAFT)</li> <li>• Minor corrections and clarifications</li> <li>• Editorial changes</li> </ul>
2019-11-28	R19-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Introduction of IPsec</li> <li>• Minor corrections and clarifications</li> <li>• Changed Document Status from Final to published</li> </ul>
2018-10-31	4.4.0	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Introduction of Transport Layer Security - TLS (DRAFT)</li> <li>• ARP timing improvements</li> <li>• minor corrections / clarifications / editorial changes</li> </ul>
2017-12-08	4.3.1	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Clarifications and corrections of requirements</li> <li>• Editorial changes</li> </ul>
2016-11-30	4.3.0	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Improvements for robustness</li> <li>• Introduction of diagnostic features</li> <li>• Clarifications and corrections of requirements</li> <li>• Editorial changes</li> </ul>
2015-07-31	4.2.2	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Support for transmission of fragmented IPv4/IPv6 frames</li> <li>• Clarifications and corrections of requirements</li> <li>• Editorial changes</li> </ul>
2014-10-31	4.2.1	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Introduction of IPv6 for in-vehicle communication</li> <li>• Support for Switch Control/Configuration, Semi-Static Auto-Configuration</li> <li>• TcpIp generic upper layer support (CDD)</li> <li>• Clarifications and corrections of requirements and sequence charts</li> </ul>





2014-03-31	4.1.3	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Clarifications and corrections of requirements</li> <li>• Editorial changes</li> </ul>
2013-10-31	4.1.2	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Added control functions for ARP</li> <li>• Clarifications and corrections of requirements</li> <li>• Editorial changes</li> <li>• Removed chapter(s) on change documentation</li> </ul>
2013-03-15	4.1.1	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Initial Release</li> </ul>

## Disclaimer

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

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

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

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

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

## Contents

1	Introduction and functional overview	10
2	Acronyms and Abbreviations	11
3	Related documentation	12
3.1	Input documents & related standards and norms	12
3.2	Related specification	16
4	Constraints and assumptions	19
4.1	Limitations	19
4.2	Applicability to car domains	19
5	Dependencies to other modules	20
5.1	LSduR	20
5.2	EthIf	20
5.3	EthSM	20
5.4	Socket Adaptor	20
5.5	KeyM	20
5.6	CSM	20
5.7	File structure	21
5.7.1	Code file structure	21
5.8	Version check	21
6	Requirements Tracing	22
7	Functional specification	25
7.1	System Scalability	25
7.1.1	Background & Rationale	25
7.1.2	Requirements	27
7.2	Internet Protocol Version 4	27
7.2.1	Internet Protocol (IPv4)	27
7.2.2	Address Resolution Protocol (ARP)	28
7.2.3	Dynamic Configuration of IPv4 Link-Local Addresses (Auto-IP)	30
7.2.4	Internet Control Message Protocol (ICMPv4)	30
7.3	Internet Protocol Version 6	30
7.3.1	Internet Protocol (IPv6)	31
7.3.2	Internet Control Message Protocol (ICMPv6)	32
7.3.3	Neighbor Discovery Protocol (NDP)	33
7.4	Internet Protocol Security (IPsec)	34
7.5	IP Based Protocols	35
7.5.1	Local Address Table	35
7.5.2	User Datagram Protocol (UDP)	36
7.5.3	Transmission Control Protocol (TCP)	36
7.5.4	Transport Layer Security (TLS)	37
7.5.5	Dynamic Host Configuration Protocol	46

7.5.5.1	Dynamic Host Configuration Protocol (DHCPv4)	47
7.5.5.2	Dynamic Host Configuration Protocol (DHCPv6)	47
7.6	Message Reception	48
7.7	Message Transmission	51
7.8	State handling	59
7.8.1	State handling of PDUs	59
7.8.2	TCP/IP Stack state handling	60
7.9	Error Classification	64
7.9.1	Development Errors	64
7.9.2	Runtime Errors	65
7.9.3	Production Errors	67
7.9.4	Extended Production Errors	68
7.10	Version checking	68
7.11	Security Events	68
8	API specification	72
8.1	Imported types	72
8.2	Type definitions	72
8.3	Symbol definitions	80
8.4	Function definitions	81
8.4.1	General	81
8.4.1.1	Tcplp_Init	81
8.4.1.2	Tcplp_GetVersionInfo	82
8.4.2	Core Communication Control	83
8.4.2.1	Tcplp_Close	83
8.4.2.2	Tcplp_Bind	84
8.4.2.3	Tcplp_TcpConnect	85
8.4.2.4	Tcplp_TcpListen	86
8.4.2.5	Tcplp_TcpReceived	87
8.4.2.6	Tcplp_RequestComMode	87
8.4.3	Extended Communication Control and Information	89
8.4.3.1	Tcplp_RequestIpAddrAssignment	89
8.4.3.2	Tcplp_ReleaseIpAddrAssignment	91
8.4.3.3	Tcplp_ResetIpAssignment	91
8.4.3.4	Tcplp_IcmpTransmit	92
8.4.3.5	Tcplp_IcmpV6Transmit	93
8.4.3.6	Tcplp_DhcpReadOption	94
8.4.3.7	Tcplp_DhcpV6ReadOption	96
8.4.3.8	Tcplp_DhcpWriteOption	97
8.4.3.9	Tcplp_DhcpV6WriteOption	99
8.4.3.10	Tcplp_ChangeParameter	100
8.4.3.11	Tcplp_GetIpAddr	101
8.4.3.12	Tcplp_GetPhysAddr	102
8.4.3.13	Tcplp_GetRemotePhysAddr	102
8.4.3.14	Tcplp_GetCtrlIdx	104
8.4.3.15	Tcplp_GetArpCacheEntries	104

8.4.3.16	Tcplp_GetNdpCacheEntries . . . . .	105
8.4.3.17	Tcplp_GetAndResetMeasurementData . . . . .	106
8.4.3.18	Tcplp_IsConnectionReady . . . . .	109
8.4.4	Transmission . . . . .	109
8.4.4.1	Tcplp_UdpTransmit . . . . .	109
8.4.4.2	Tcplp_TcpTransmit . . . . .	111
8.5	Callback notifications . . . . .	112
8.5.1	Tcplp_RxIndication . . . . .	112
8.5.2	Tcplp_TriggerTransmit . . . . .	113
8.5.3	Tcplp_TxConfirmation . . . . .	114
8.6	Scheduled functions . . . . .	115
8.6.1	Terms and definitions . . . . .	115
8.6.2	Tcplp_MainFunction . . . . .	116
8.7	Expected interfaces . . . . .	116
8.7.1	Mandatory interfaces . . . . .	116
8.7.2	Optional interfaces . . . . .	117
8.7.3	Configurable interfaces . . . . .	118
8.7.3.1	Tcplp_<Up>GetSocket . . . . .	119
8.7.3.2	<Up_PhysAddrTableChg> . . . . .	120
8.7.3.3	SocketOwner functions . . . . .	120
8.7.3.3.1	<Up_RxIndication> . . . . .	121
8.7.3.3.2	<Up_TcplpEvent> . . . . .	121
8.7.3.3.3	<Up_TxConfirmation> . . . . .	122
8.7.3.3.4	<Up_TcpAccepted> . . . . .	123
8.7.3.3.5	<Up_TcpConnected> . . . . .	123
8.7.3.3.6	<Up_CopyTxData> . . . . .	124
8.7.3.3.7	<Up_LocallpAddrAssignmentChg> . . . . .	125
8.7.3.4	<Up_IcmpMsgHandler> . . . . .	125
8.7.3.5	<Up_DADAddressConflict> . . . . .	126
8.7.3.6	<Up_TlsGetCurrentTime> . . . . .	127
8.7.3.7	<Up_TlsServerGetPskIdentityHint> . . . . .	128
8.7.3.8	<Up_TlsClientGetPskIdentity > . . . . .	129
8.7.3.9	<Up_TlsServerGetPskIdentity> . . . . .	130
8.8	Service Interfaces . . . . .	130
9	Sequence diagrams . . . . .	131
9.1	TCP Connection Setup - Client . . . . .	131
9.2	TCP Connection Setup - Server . . . . .	132
9.3	Reception . . . . .	133
9.4	Transmission TCP . . . . .	134
9.5	Transmission UDP . . . . .	135
9.6	Connection setup for a TLS server . . . . .	136
9.7	TLS connection assignment to socket . . . . .	137
10	Configuration specification . . . . .	138
10.1	How to read this chapter . . . . .	138
10.2	Containers and configuration parameters . . . . .	138

10.2.1	Tcplp	138
10.2.2	TcplpGeneral	140
10.2.3	TcplpV4General	146
10.2.4	TcplpV6General	149
10.2.5	TcplpSecurityEventRefs	154
10.2.6	TcplpConfig	160
10.2.7	TcplpCtrl	161
10.2.8	TcplpCtrlRxPdu	164
10.2.9	TcplpCtrlTxPdu	165
10.2.10	TcplpVXCtrl	167
10.2.11	TcplpV4Ctrl	168
10.2.12	TcplpV4MtuConfig	170
10.2.13	TcplpV6Ctrl	172
10.2.14	TcplpV6MtuConfig	174
10.2.15	TcplpDhcpServerConfig	176
10.2.16	TcplpDhcpAddressAssignment	178
10.2.17	TcplpDuplicateAddressDetectionConfig	180
10.2.18	TcplpConfig	181
10.2.19	TcplpV4Config	182
10.2.20	TcplpArpConfig	184
10.2.21	TcplpAutoIpConfig	187
10.2.22	TcplpDhcpConfig	188
10.2.23	TcplpIcmpConfig	189
10.2.24	TcplpIcmpMsgHandler	190
10.2.25	TcplpFragmentationConfig	191
10.2.26	TcplpV6Config	194
10.2.27	TcplpDhcpV6Config	196
10.2.28	TcplpIcmpV6Config	199
10.2.29	TcplpIcmpV6MsgHandler	202
10.2.30	TcplpV6ConfigExtHeaderFilter	202
10.2.31	TcplpV6FragmentationConfig	204
10.2.32	TcplpNdpConfig	208
10.2.33	TcplpNdpArNudConfig	211
10.2.34	TcplpNdpPrefixRouterDiscoveryConfig	219
10.2.35	TcplpNdpPrefixList	224
10.2.36	TcplpNdpPrefixListEntry	225
10.2.37	TcplpNdpSlaacConfig	227
10.2.38	TcplpLocalAddr	230
10.2.39	TcplpAddrAssignment	234
10.2.40	TcplpStaticIpAddressConfig	237
10.2.41	TcplpNvmBlock	239
10.2.42	TcplpPhysAddrConfig	240
10.2.43	TcplpPhysAddrChgHandler	240
10.2.44	TcplpSocketOwnerConfig	242
10.2.45	TcplpSocketOwner	243
10.2.46	TcplpTcpConfig	250

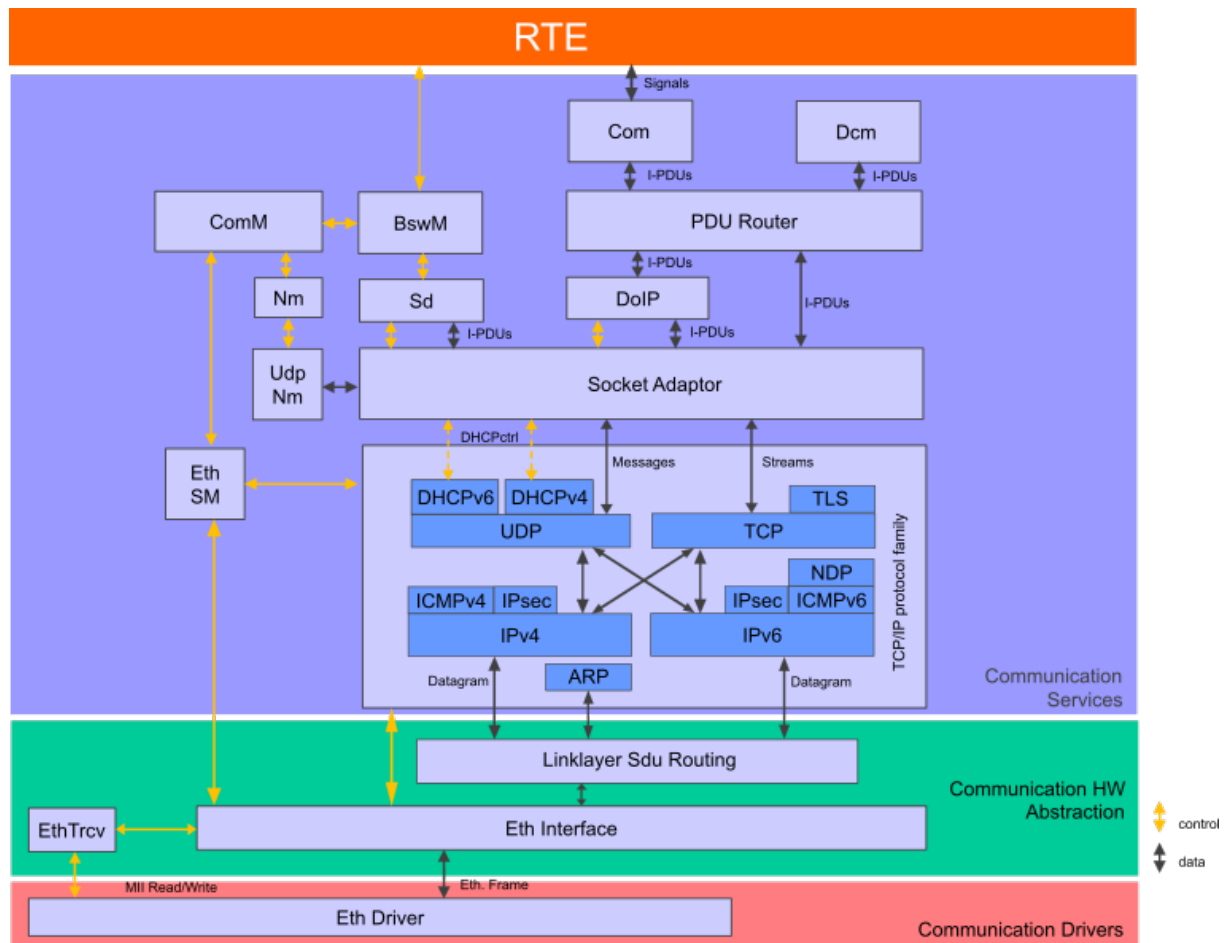


10.2.47	TcplpTcpConfigOptionFilter . . . . .	260
10.2.48	TcplpUdpConfig . . . . .	262
10.2.49	TcplpTlsConfig . . . . .	263
10.2.50	TcplpTlsConnectionGroup . . . . .	265
10.2.51	TcplpTlsConnection . . . . .	267
10.2.52	TcplpTlsCiphersuites . . . . .	275
10.2.53	TcplpTlsCiphersuiteDefinition . . . . .	278
10.2.54	TcplpTlsCiphersuiteWorker . . . . .	283
10.2.55	TcplpTlsHandshake . . . . .	290
10.2.56	TcplpTlsCertificateIdentity . . . . .	298
10.2.57	TcplpTlsPskIdentity . . . . .	302
10.2.58	TcplpSecConfigSet . . . . .	306
10.2.59	TcplpEncryptionAlgorithm . . . . .	309
10.2.60	TcplpEncryptionTransformJobPair . . . . .	311
10.2.61	TcplpIntegrityAlgorithm . . . . .	313
10.2.62	TcplpIntegrityTransformJobPair . . . . .	315
10.2.63	TcplpSpdEntry . . . . .	318
10.2.64	IKE . . . . .	324
10.2.65	IKEGeneral . . . . .	325
10.2.66	IKEMessageFormat . . . . .	327
10.2.67	IKEConnections . . . . .	335
10.2.68	IKEConnection . . . . .	341
10.2.69	IKESession . . . . .	347
10.2.70	IKEChildSaProposal . . . . .	348
10.2.71	IKEIkeSaProposal . . . . .	351
10.2.72	IKESignatureAuthenticationVariant . . . . .	354
10.2.73	IKETransforms . . . . .	358
10.2.74	IKEDhTransform . . . . .	361
10.2.75	IKEEncrTransform . . . . .	364
10.2.76	KEIntegTransform . . . . .	368
10.2.77	IKEPrfTransform . . . . .	371
10.2.78	IKECertificates . . . . .	374
10.2.79	IKECertificate . . . . .	376
10.3	Published Information . . . . .	378
A	Change history of AUTOSAR traceable items . . . . .	379
A.1	Traceable item history of this document according to AUTOSAR Release R24-11 . . . . .	379
A.1.1	Added Specification Items in R24-11 . . . . .	379
A.1.2	Changed Specification Items in R24-11 . . . . .	379
A.1.3	Deleted Specification Items in R24-11 . . . . .	379
A.1.4	Added Constraints in R24-11 . . . . .	380
A.1.5	Changed Constraints in R24-11 . . . . .	380
A.1.6	Deleted Constraints in R24-11 . . . . .	380

# 1 Introduction and functional overview

This specification describes the functionality, API and the configuration for the AUTOSAR Basic Software module TCP/IP.

The AUTOSAR TCP/IP module offers functionality to send and receive Internet Protocol data. The TCP/IP Stack (Tcplp) is located between the Socket Adaptor (SoAd) and Linklayer SDU Routing Module (LSduR). The Tcplp module exchange L-SDUs via the LSduR with the Ethernet Interface (EthIf) module.



**Figure 1.1: Extended AUTOSAR Communication Stack**

## 2 Acronyms and Abbreviations

The glossary below includes acronyms and abbreviations relevant to the Tcp/Ip module that are not included in the [1, AUTOSAR glossary].

Abbreviation / Acronym:	Description:
ARP	Address Resolution Protocol
DAD	Duplicate Address Detection
DEM	Diagnostic Event Manager
DET	Default 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
ECC	Elliptic Curve Cryptography
ECU	Electronic Control Unit
EthIf	Ethernet Interface
EthSM	Ethernet State Manager
HSM	Hardware Security Module
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
IND	Inverse Neighbor Discovery
IPsec	Internet Protocol Security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
MTU	Maximum Transmission Unit
NDP	Neighbor Discovery Protocol
PKI	Public Key Infrastructure
PRF	Pseudo Random Function
RNG	Random Number Generator
RSA	Rivest-Shamir-Adleman. A method using public and private key for data encryption and decryption.
SACK	Selective Acknowledgment
SNI	Server Name Identification
SoAd	Socket Adaptor
TCP	Transmission Control Protocol
TCP/IP	A family of communication protocols used in computer networks
TLS	Transport Layer Security
TP	Transport Protocol
UDP	User Datagram Protocol

**Table 2.1: Acronyms and abbreviations used in the scope of this Document**

## 3 Related documentation

### 3.1 Input documents & related standards and norms

- [1] Glossary  
AUTOSAR\_FO\_TR\_Glossary
- [2] Layered Software Architecture  
AUTOSAR\_CP\_EXP\_LayeredSoftwareArchitecture
- [3] Specification of Basic Software Mode Manager  
AUTOSAR\_CP\_SWS\_BSWModeManager
- [4] Specification of Linklayer Sdu Routing Module  
AUTOSAR\_CP\_SWS\_LSduRouter
- [5] Specification of Socket Adaptor  
AUTOSAR\_CP\_SWS\_SocketAdaptor
- [6] General Specification of Basic Software Modules  
AUTOSAR\_CP\_SWS\_BSWGeneral
- [7] Requirements on Ethernet Support in AUTOSAR  
AUTOSAR\_CP\_RS\_Ethernet
- [8] Specification of ECU Configuration  
AUTOSAR\_CP\_TPS\_ECUConfiguration
- [9] Specification of Crypto Service Manager  
AUTOSAR\_CP\_SWS\_CryptoServiceManager
- [10] Specification of Key Manager  
AUTOSAR\_CP\_SWS\_KeyManager
- [11] Requirements on IPsec Protocol  
AUTOSAR\_FO\_RS\_IPsecProtocol
- [12] Dynamic Configuration of IPv4 Link-Local Addresses  
<https://rfc-editor.org/rfc/rfc3927.txt>
- [13] Requirements for Internet Hosts - Communication Layers  
<https://rfc-editor.org/rfc/rfc1122.txt>
- [14] An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware  
<https://rfc-editor.org/rfc/rfc826.txt>
- [15] A Standard for the Transmission of IP Datagrams over Ethernet Networks  
<https://www.rfc-editor.org/info/rfc894>
- [16] Internet Protocol  
<https://rfc-editor.org/rfc/rfc791.txt>

- [17] IP DATAGRAM REASSEMBLY ALGORITHMS  
<https://www.rfc-editor.org/info/rfc815>
- [18] Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan  
<https://www.rfc-editor.org/info/rfc4632>
- [19] Host Extensions for IP Multicasting  
<https://www.rfc-editor.org/info/rfc1112>
- [20] Internet Control Message Protocol  
<https://rfc-editor.org/rfc/rfc792.txt>
- [21] Path MTU Discovery  
<https://www.rfc-editor.org/info/rfc1191>
- [22] Dynamic Host Configuration Protocol  
<https://rfc-editor.org/rfc/rfc2131.txt>
- [23] User Datagram Protocol  
<https://rfc-editor.org/rfc/rfc768.txt>
- [24] Transmission Control Protocol  
<https://rfc-editor.org/rfc/rfc793.txt>
- [25] TCP Congestion Control  
<https://rfc-editor.org/rfc/rfc5681.txt>
- [26] Internet Protocol, Version 6 (IPv6) Specification  
<https://www.rfc-editor.org/info/rfc8200>
- [27] IP Version 6 Addressing Architecture  
<https://rfc-editor.org/rfc/rfc4291.txt>
- [28] Transmission of IPv6 Packets over Ethernet Networks  
<https://rfc-editor.org/rfc/rfc2464.txt>
- [29] Default Address Selection for Internet Protocol Version 6 (IPv6)  
<https://rfc-editor.org/rfc/rfc6724.txt>
- [30] Handling of Overlapping IPv6 Fragments  
<https://rfc-editor.org/rfc/rfc5722.txt>
- [31] Deprecation of Type 0 Routing Headers in IPv6  
<https://rfc-editor.org/rfc/rfc5095.txt>
- [32] IPv6 Stateless Address Autoconfiguration  
<https://rfc-editor.org/rfc/rfc4862.txt>
- [33] Path MTU Discovery for IP version 6  
<https://rfc-editor.org/rfc/rfc1981.txt>
- [34] Optimistic Duplicate Address Detection (DAD) for IPv6  
<https://rfc-editor.org/rfc/rfc4429.txt>

- [35] Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification  
<https://rfc-editor.org/rfc/rfc4443.txt>
- [36] Neighbor Discovery for IP version 6 (IPv6)  
<https://rfc-editor.org/rfc/rfc4861.txt>
- [37] Dynamic Host Configuration Protocol for IPv6 (DHCPv6)  
<https://rfc-editor.org/rfc/rfc3315.txt>
- [38] The Dynamic Host Configuration Protocol (DHCP) Client Fully Qualified Domain Name (FQDN) Option  
<https://rfc-editor.org/rfc/rfc4702.txt>
- [39] The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option  
<https://rfc-editor.org/rfc/rfc4704.txt>
- [40] The NewReno Modification to TCP's Fast Recovery Algorithm  
<https://rfc-editor.org/rfc/rfc6582.txt>
- [41] DHCP Options and BOOTP Vendor Extensions  
<https://rfc-editor.org/rfc/rfc2132.txt>
- [42] IPv6 Subnet Model: The Relationship between Links and Subnet Prefixes  
<https://rfc-editor.org/rfc/rfc5942.txt>
- [43] IPv6 Flow Label Specification  
<https://rfc-editor.org/rfc/rfc6437.txt>
- [44] Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers  
<https://rfc-editor.org/rfc/rfc2474.txt>
- [45] The Transport Layer Security (TLS) Protocol Version 1.2  
<https://rfc-editor.org/rfc/rfc5246.txt>
- [46] Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)  
<https://rfc-editor.org/rfc/rfc4492.txt>
- [47] Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)  
<https://www.rfc-editor.org/info/rfc7525>
- [48] Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)  
<https://rfc-editor.org/rfc/rfc4279.txt>
- [49] Encrypt-then-MAC for Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)  
<https://www.rfc-editor.org/info/rfc7366>
- [50] The Transport Layer Security (TLS) Protocol Version 1.3

<https://tools.ietf.org/html/rfc8446>

- [51] Record Size Limit Extension for TLS  
<https://tools.ietf.org/html/rfc8449>
- [52] Dynamic Host Configuration Protocol (DHCP) and Bootstrap Protocol (BOOTP) Parameters  
<https://www.iana.org/assignments/bootp-dhcp-parameters/bootp-dhcp-parameters.xhtml>
- [53] Dynamic Host Configuration Protocol for IPv6 (DHCPv6)  
<https://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xhtml>
- [54] RFC 4301, Security Architecture for the Internet Protocol
- [55] RFC 4302, IP Authentication Header
- [56] RFC 4303, IP Encapsulating Security Payload (ESP)
- [57] RFC 7296, Internet Key Exchange Protocol Version 2 (IKEv2)
- [58] RFC 4304, Extended Sequence Number (ESN) Addendum to IPsec Domain of Interpretation (DOI) for Internet Security Association
- [59] RFC 8221, Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)
- [60] RFC 4478, Repeated Authentication in Internet Key Exchange (IKEv2) Protocol
- [61] RFC 3706, A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers
- [62] RFC 7427, Signature Authentication in the Internet Key Exchange Version 2 (IKEv2)
- [63] RFC 4543, The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH
- [64] RFC 4494, The AES-CMAC-96 Algorithm and Its Use with IPsec
- [65] RFC 4106, The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)
- [66] RFC 4309, Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP)
- [67] RFC 6379, Suite B Cryptographic Suites for IPsec
- [68] RFC 8247, Algorithm Implementation Requirements and Usage Guidance for the Internet Key Exchange Protocol Version 2 (IKEv2)
- [69] Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation  
<https://www.rfc-editor.org/info/rfc7383>

- [70] ISO 13400-2:2019 – Road vehicles – Diagnostic communication over Internet Protocol (DoIP) – Part 2: Network and transport layer requirements and services (Release 2019-12)  
<https://www.iso.org/standard/74785.html>

## 3.2 Related specification

AUTOSAR provides a Specification of Layered Software Architecture [2], which is also valid for TcpIp.

AUTOSAR provides a Specification of Basis Software Mode Manager [3].

AUTOSAR provides a Specification of Linklayer SDU Routing Module (LSduR) [4].

AUTOSAR provides a Specification of Socket Adaptor [5].

AUTOSAR provides a General Specification for Basic Software modules [6].

AUTOSAR provides a Specification of Ethernet [7].

AUTOSAR provides a Specification of ECU Configuration [8], which is also valid for TcpIp.

AUTOSAR provides a Specification of Crypto Service Manager [9].

AUTOSAR provides a Specification of Key Manager [10].

AUTOSAR provides a Specification of IPsecProtocol [11].

IETF RFC 3927 [12].

IETF RFC 1122 [13].

IETF RFC 826 [14].

IETF RFC 894 [15].

IETF RFC 791 [16].

IETF RFC 815 [17].

IETF RFC 4632 [18].

IETF RFC 1112 [19].

IETF RFC 792 [20].

IETF RFC 1191 [21].

IETF RFC 2131 [22].

IETF RFC 768 [23].

IETF RFC 793 [24].



IETF RFC 5681 [25].  
IETF RFC 8200 [26].  
IETF RFC 4291 [27].  
IETF RFC 2464 [28].  
IETF RFC 6724 [29].  
IETF RFC 5722 [30].  
IETF RFC 5095 [31].  
IETF RFC 4862 [32].  
IETF RFC 1981 [33].  
IETF RFC 4429 [34].  
IETF RFC 4443 [35].  
IETF RFC 4861 [36].  
IETF RFC 3315 [37].  
IETF RFC 4702 [38].  
IETF RFC 4704 [39].  
IETF RFC 6582 [40].  
IETF RFC 2132 [41].  
IETF RFC 5942 [42].  
IETF RFC 6437 [43].  
IETF RFC 2474 [44].  
IETF RFC 5246 [45].  
IETF RFC 4492 [46].  
IETF RFC 7525 [47].  
IETF RFC 4279 [48].  
IETF RFC 7366 [49].  
IETF RFC 8446 [50].  
IETF RFC 8449 [51].  
IANA DHCP Options [52].  
IANA DHCPv6 Options [53].  
IETF RFC 4301 [54].

IETF RFC 4302 [55].  
IETF RFC 4303 [56].  
IETF RFC 7296 [57].  
IETF RFC 4304 [58].  
IETF RFC 8221 [59].  
IETF RFC 4478 [60].  
IETF RFC 3706 [61].  
IETF RFC 7427 [62].  
IETF RFC 4543 [63].  
IETF RFC 4494 [64].  
IETF RFC 4106 [65].  
IETF RFC 4309 [66].  
IETF RFC 6379 [67].  
IETF RFC 8247 [68].  
IETF RFC 7383 [69].  
IETF ISO13400-2 [70].

## 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.

The AUTOSAR TLS implementation has the following limitations:

- A TLS implementation shall not support data compression or decompression.
- Session renegotiation shall not be supported.
- No support for secure connection over UDP (e.g. for DTLS)
- No support of FQDN
- No client Hello padding extension IETF RFC7685
- No session hash and extended master secret IETF RFC 7627
- No support for TLS versions lower than 1.2.
- No support for dynamic "downgrading" of a TCP connection with an established TLS connection to a plain TCP connection (without TLS)
- Static TLS connection assignment is bound to the port configuration of the server. Thus, using different TLS settings for different connections (possibly originating from different clients) to the same server port is not possible.

The AUTOSAR IPsec implementation has the following limitations:

- IPsec in "tunnel mode" is not supported right now. Transport mode only.
- IPv6 is not supported
- Multicast is not supported

### 4.2 Applicability to car domains

No restrictions.

## 5 Dependencies to other modules

### 5.1 LSduR

The Linklayer SDU Router is the lower layer module of the Tcp/Ip module regarding the data path (e.g. transmission/reception of PDUs). Supported MetaDataItemTypes are provided/consumed by EthIf.

### 5.2 EthIf

The Ethernet Interface is the lower layer module of the Tcp/Ip module regarding the control path (e.g. setting the local physical address of an Ethernet controller via the EthIf (EthIf\_SetPhysAddr)), and ultimately via LSduR also for the data path.

### 5.3 EthSM

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

### 5.4 Socket Adaptor

The Socket Adaptor is the upper layer module of the Tcp/Ip module.

### 5.5 KeyM

The Key Manager module provides operations for certificate handling for the TLS and IPsec sub module.

### 5.6 CSM

The crypto service manager allows to perform crypto job and key operations used by the TLS and IPsec sub module.

## **5.7 File structure**

### **5.7.1 Code file structure**

For details refer to the chapter 5.1.6 "Code file structure" in SWS\_BSWGeneral.

## **5.8 Version check**

For details refer to the chapter 5.1.8 "Version Check" in SWS\_BSWGeneral.

## 6 Requirements Tracing

The following tables reference the requirements specified in [Chapter 3](#) and links to the fulfillment of these. Please note that if column “Satisfied by” is empty for a specific requirement this means that this requirement is not fulfilled by this document.

Requirement	Description	Satisfied by
[RS_IPSEC_00004]	The Internet Key Exchange (IKEv2) Protocol shall be supported according to IETF RFC 7296	<a href="#">[SWS_Tcplp_00353]</a>
[RS_IPSEC_00010]	IKEv2 shall support periodic reauthentication and rekeying	<a href="#">[SWS_Tcplp_00355]</a>
[RS_IPSEC_00011]	IKEv2 shall support a seamless handover of exchanged keys	<a href="#">[SWS_Tcplp_00355]</a>
[RS_IPSEC_00013]	IKEv2 shall support dead peer detection	<a href="#">[SWS_Tcplp_00355]</a>
[RS_IPSEC_00014]	IKEv2 shall support authentication based on X.509v3 certificates with digital signatures	<a href="#">[SWS_Tcplp_00356]</a>
[RS_IPSEC_00021]	All algorithms which are classified as "MUST" in IETF RFC 8247 shall be supported by IKEv2	<a href="#">[SWS_Tcplp_00353]</a>
[RS_IPSEC_00022]	IPsec's Security Policy Database (SPD) shall be configurable for IPs, IP ranges, protocols, ports and port ranges	<a href="#">[SWS_Tcplp_00357]</a>
[RS_IPSEC_00023]	IPsec's Security Policy Database (SPD) default behavior shall be BYPASS	<a href="#">[SWS_Tcplp_00357]</a>
[RS_IPSEC_00025]	IPsec's Peer Authorization Database (PAD) shall be configurable for use with X.509v3	<a href="#">[SWS_Tcplp_00356]</a>
[RS_IPSEC_00027]	It shall be possible to define the priority order of the algorithms used by IKEv2 during the IKE_INIT negotiations	<a href="#">[SWS_Tcplp_00358]</a>
[RS_Ids_00810]	Basic SW security events	<a href="#">[SWS_Tcplp_00361]</a> <a href="#">[SWS_Tcplp_00362]</a> <a href="#">[SWS_Tcplp_00382]</a> <a href="#">[SWS_Tcplp_00383]</a> <a href="#">[SWS_Tcplp_00384]</a> <a href="#">[SWS_Tcplp_00385]</a> <a href="#">[SWS_Tcplp_00386]</a> <a href="#">[SWS_Tcplp_00387]</a> <a href="#">[SWS_Tcplp_00388]</a> <a href="#">[SWS_Tcplp_00389]</a> <a href="#">[SWS_Tcplp_00394]</a> <a href="#">[SWS_Tcplp_00395]</a> <a href="#">[SWS_Tcplp_00396]</a>
[SRS_BSW_00323]	All AUTOSAR Basic Software Modules shall check passed API parameters for validity	<a href="#">[SWS_Tcplp_00147]</a>
[SRS_BSW_00452]	Classification of runtime errors	<a href="#">[SWS_Tcplp_00282]</a> <a href="#">[SWS_Tcplp_00283]</a>
[SRS_Eth_00016]	ICMPv4 shall be implemented according to IETF RFC 792	<a href="#">[SWS_Tcplp_00277]</a> <a href="#">[SWS_Tcplp_00297]</a>
[SRS_Eth_00019]	TCP and UDP related requirement specified in IETF RFC 1122 shall be implemented	<a href="#">[SWS_Tcplp_00279]</a> <a href="#">[SWS_Tcplp_00280]</a>
[SRS_Eth_00045]	TCPIP automatic IP address assignment	<a href="#">[SWS_Tcplp_00254]</a>





Requirement	Description	Satisfied by
[SRS_Eth_00065]	An API shall be available to fill DHCP options field	[SWS_TCPIP_00020] [SWS_TCPIP_00190] [SWS_Tcplp_00243] [SWS_Tcplp_00244] [SWS_Tcplp_00245] [SWS_Tcplp_00246] [SWS_Tcplp_00247] [SWS_Tcplp_00248] [SWS_Tcplp_00249] [SWS_Tcplp_00250] [SWS_Tcplp_00251] [SWS_Tcplp_00252]
[SRS_Eth_00066]	An API shall be available to read any received DHCP options field	[SWS_TCPIP_00040] [SWS_TCPIP_00189] [SWS_Tcplp_00233] [SWS_Tcplp_00234] [SWS_Tcplp_00235] [SWS_Tcplp_00236] [SWS_Tcplp_00237] [SWS_Tcplp_00238] [SWS_Tcplp_00239] [SWS_Tcplp_00240] [SWS_Tcplp_00241] [SWS_Tcplp_00242]
[SRS_Eth_00087]	Semi-Static Auto-Configuration	[SWS_Tcplp_00058] [SWS_Tcplp_00201] [SWS_Tcplp_00216] [SWS_Tcplp_00217] [SWS_Tcplp_00218] [SWS_Tcplp_00219]
[SRS_Eth_00088]	DHCP Server	[SWS_Tcplp_00058] [SWS_Tcplp_00200]
[SRS_Eth_00090]	The Neighbor Discovery Protocol shall be implemented according to IETF RFC 4861	[SWS_Tcplp_00164] [SWS_Tcplp_00263] [SWS_Tcplp_00264] [SWS_Tcplp_00281]
[SRS_Eth_00091]	The Optimistic Duplicate Address Detection (DAD) for IPv6 shall be implemented according to IETF RFC 4429	[SWS_Tcplp_00282] [SWS_Tcplp_00283]
[SRS_Eth_00092]	The IPv6 Addressing Architecture shall be implemented according to IETF RFC 4291	[SWS_Tcplp_00162] [SWS_Tcplp_00269]
[SRS_Eth_00097]	The Path MTU Discovery for IPv6 shall be implemented according to IETF RFC 1981	[SWS_Tcplp_00267] [SWS_Tcplp_00268]
[SRS_Eth_00098]	ICMPv6 shall be implemented according to IETF RFC 4443	[SWS_Tcplp_00278] [SWS_Tcplp_00298]
[SRS_Eth_00103]	Tcplp shall support generic upper layers	[SWS_TCPIP_00018] [SWS_TCPIP_00223] [SWS_TCPIP_00224] [SWS_TCPIP_00225] [SWS_TCPIP_00226] [SWS_TCPIP_00227] [SWS_TCPIP_00228] [SWS_TCPIP_00229] [SWS_Tcplp_00220] [SWS_Tcplp_00221] [SWS_Tcplp_00222]
[SRS_Eth_00109]	TCP shall support the Nagle algorithm according to IETF RFC 896	[SWS_Tcplp_00063]
[SRS_Eth_00110]	The Relationship between Links and Subnet Prefixes shall be considered according to IETF RFC 5942	[SWS_Tcplp_00265]
[SRS_Eth_00111]	Robustness against unexpected communication patterns	[SWS_Tcplp_00260] [SWS_Tcplp_00261] [SWS_Tcplp_00262] [SWS_Tcplp_00266] [SWS_Tcplp_00370] [SWS_Tcplp_00371]
[SRS_Eth_00112]	Ethernet-related BSW modules shall report relevant runtime errors from the used protocols	[SWS_TCPIP_00255] [SWS_Tcplp_00256] [SWS_Tcplp_00257] [SWS_Tcplp_00258] [SWS_Tcplp_00259]
[SRS_Eth_00129]	The TCPIP shall support access to measurement counter values	[SWS_Tcplp_00284] [SWS_Tcplp_00286] [SWS_Tcplp_00287] [SWS_Tcplp_00288] [SWS_Tcplp_00289] [SWS_Tcplp_00290] [SWS_Tcplp_00291] [SWS_Tcplp_00292] [SWS_Tcplp_00293] [SWS_Tcplp_00294] [SWS_Tcplp_00295] [SWS_Tcplp_00296]
[SRS_Eth_00134]	Configuration of ciphersuites for TLS connections	[SWS_Tcplp_00311]





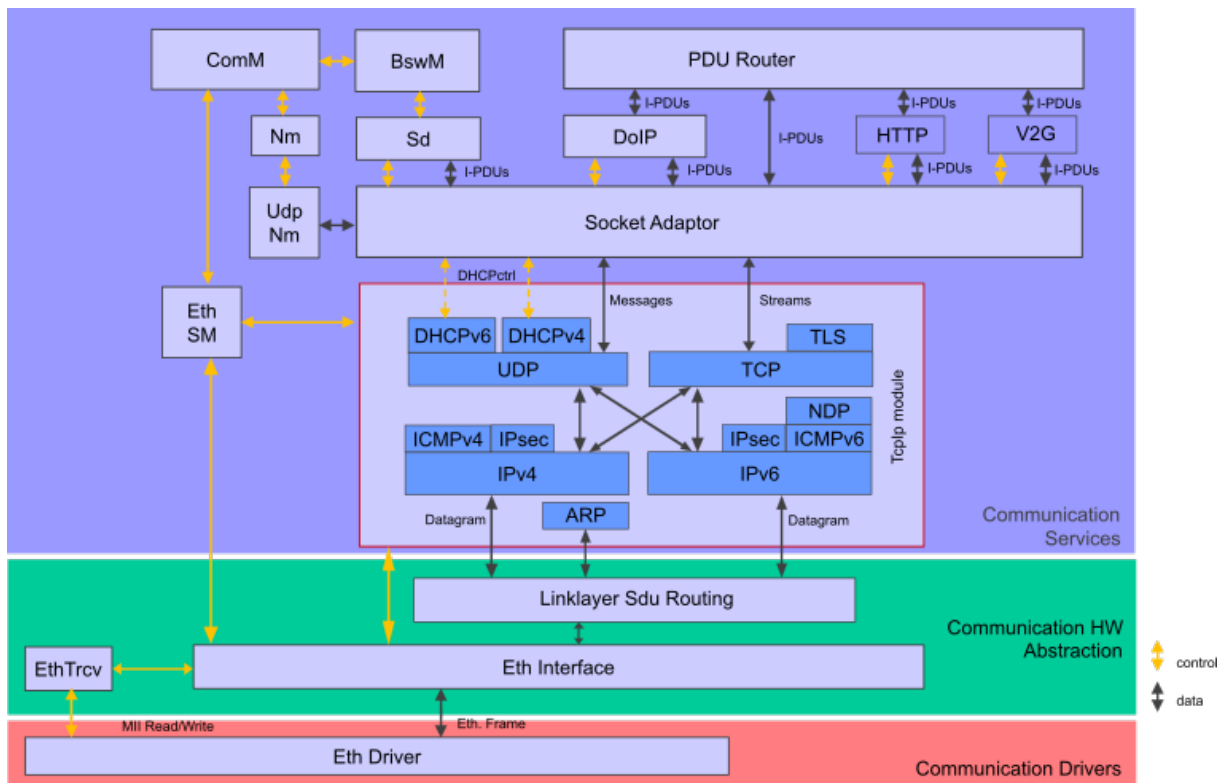
Requirement	Description	Satisfied by
[SRS_Eth_00135]	The number of TLS connections that can be opened in parallel shall be configurable	[SWS_Tcplp_00326]
[SRS_Eth_00136]	The size of a TLS fragment length shall be configurable	[SWS_Tcplp_00327]
[SRS_Eth_00137]	PSK Identity to PSK mapping shall be possible using custom software.	[SWS_TCPIP_91013] [SWS_TCPIP_91014] [SWS_TCPIP_91015] [SWS_Tcplp_00325]
[SRS_Eth_00138]	TLS shall support at least basic requirements as defined in IETF RFC 5246 for version 1.2 or higher	[SWS_Tcplp_00300] [SWS_Tcplp_00302]
[SRS_Eth_00139]	TLS shall support elliptic curve cryptography as defined in IETF RFC 4492	[SWS_Tcplp_00304]
[SRS_Eth_00141]	TLS shall support the use of pre-shared keys as defined in IETF RFC 4279	[SWS_Tcplp_00325]
[SRS_Eth_00142]	The Security Architecture for the Internet Protocol shall be implemented according to IETF RFC 4301	[SWS_Tcplp_00352]
[SRS_Eth_00143]	The IP Authentication Header (AH) shall be implemented according to IETF RFC 4302	[SWS_Tcplp_00352]
[SRS_Eth_00144]	IP Encapsulating Security Payload (ESP) shall be implemented according to IETF RFC 4303	[SWS_Tcplp_00352]
[SRS_Eth_00145]	The Internet Key Exchange (IKEv2) Protocol shall be implemented according to IETF RFC 7296	[SWS_Tcplp_00352]
[SRS_Eth_00187]	The Tcplp module shall support PDU based communication	[SWS_Tcplp_00401] [SWS_Tcplp_00402] [SWS_Tcplp_00403] [SWS_Tcplp_00404] [SWS_Tcplp_00405] [SWS_Tcplp_00406] [SWS_Tcplp_00407] [SWS_Tcplp_00408] [SWS_Tcplp_00409] [SWS_Tcplp_00410] [SWS_Tcplp_00411] [SWS_Tcplp_00412] [SWS_Tcplp_00413] [SWS_Tcplp_00414] [SWS_Tcplp_00415] [SWS_Tcplp_00416] [SWS_Tcplp_00417] [SWS_Tcplp_00418] [SWS_Tcplp_00419] [SWS_Tcplp_00420]

**Table 6.1: Requirements Tracing**



## 7 Functional specification

Figure 7.1 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 7.1: TCP/IP Architecture Overview**

[SWS\_Tcplp\_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 protocols 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

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

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

**Table 7.1: TcpIp Scalability Classes**

In addition to the scalability classes, the following Feature Groups allow a more fine-grained selection of optional features to address the specific needs of certain ECUs.

**IPv4-Global Communication Feature Group:**

The following features are available for Scalability Classes 1 and 3.

- Path MTU Discovery

**IPv6-Global Communication Feature Group:**

The following features are available for Scalability Classes 2 and 3.

- Path MTU Discovery
- IPv6 Anycasts Addresses
- NDP Redirect Messages

**Special Features Group:**

The following features are available for Scalability Classes 1, 2 and 3.

- DHCP Server

## Security Features Group:

The following features are available for Scalability Classes 1, 2 and 3.

- TLS
- IPsec

### 7.1.2 Requirements

**[SWS\_Tcplp\_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\_Tcplp\_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\_Tcplp\_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\_Tcplp\_00053]** [The Tcplp shall implement the Internet Protocol as defined in IETF RFC 791 (Internet Protocol of version 4).]

**[SWS\_Tcplp\_00095]** [The Tcplp shall encapsulate IP packets in Ethernet frames according to IETF RFC 894.]

**[SWS\_Tcplp\_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\_Tcplp\_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\_Tcplp\_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\_Tcplp\_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\_Tcplp\_00054]** [The Tcplp shall be able to reassemble incoming datagrams that are fragmented according to IETF RFC 815 (IP Datagram Reassembly Algorithms).]

**[SWS\_Tcplp\_00231]** [The Tcplp shall fragment oversized IPv4 frames before transmission according to the description in IETF 791 Section Fragmentation and Reassembly.]

**[SWS\_Tcplp\_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\_Tcplp\_00056]** [The Tcplp shall implement the Address Resolution Protocol (ARP) as defined in IETF RFC 826.]

**[SWS\_Tcplp\_00090]** [The Tcplp shall limit the number of ARP table (address resolution cache) entries to the number specified by the configuration parameter `TcpIpArpTableSizeMax`.]

**[SWS\_Tcplp\_00091]** [The Tcplp shall remove entries of the ARP table if they are not used for the timeout specified by the configuration parameter `TcpIpArpTableEntryTimeout`. If `TcpIpArpTableEntryTimeout` is set to INF, the Tcplp module shall never remove entries from the ARP table.]

**[SWS\_Tcplp\_00092]** [If `TcpIpArpDefensiveProcessing` is set to FALSE, the Tcplp shall use the information from each received IP packet to update the ARP table in addition to received ARP packets.]

**[SWS\_Tcplp\_00142]** [The Tcplp shall call `Up_PhysAddrTableChg()` directly after each ARP table change:

- 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.
- 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\_Tcplp\_00350]** [After the transmission of an ARP request the Tcplp shall skip the transmission of any further ARP requests to the same destination within a duration of `TcpIpArpRequestTimeout` seconds, according to the mechanism to prevent ARP flooding described in IETF RFC 1122, section 2.3.2.1 ARP Cache Validation.]

**[SWS\_Tcplp\_00351]** [The Tcplp shall process received ARP packets either directly within the context of the `TcpIp_RxIndication()` or the first subsequent `TcpIp_MainFunction()`.]

**[SWS\_Tcplp\_00093]** [On assignment of a new IP address the Tcplp shall send a configurable number (`TcpIpArpNumGratuitousARPOnStartup`) of gratuitous ARP replies according to IETF RFC 2002, section 4.6, second indent. These announcements shall be timed according to IETF RFC 5227 section 2.3. Announcing an Address.]

#### **[SWS\_Tcplp\_00370]**

*Upstream requirements:* [SRS\\_Eth\\_00111](#)

[If `TcpIpArpDefensiveProcessing` is set to TRUE, the ARP shall silently discard all received ARP packets that have not been requested by a previously transmitted ARP request.]

#### **[SWS\_Tcplp\_00371]**

*Upstream requirements:* [SRS\\_Eth\\_00111](#)

[If `TcpIpArpDefensiveProcessing` is set to TRUE, the ARP shall skip the update of the ARP table upon processing received Gratuitous ARP packets.]

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

**[SWS\_Tcplp\_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\_Tcplp\_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).]

**[SWS\_Tcplp\_00277]**

*Upstream requirements:* [SRS\\_Eth\\_00016](#)

[The Tcplp shall only reply to ICMPv4 Echo Request Messages if they are valid and [TcpIpIcmpEchoReplyEnabled](#) is set to TRUE.]

**[SWS\_Tcplp\_00297]**

*Upstream requirements:* [SRS\\_Eth\\_00016](#)

[If a [TcpIpIcmpMsgHandler](#) is configured, the Tcplp shall call the respective [Up\\_IcmpMsgHandler\(\)](#) if an ICMPv4 message is received and not handled by the Tcplp directly.]

Note: For example, if the Tcplp replies to an ICMP echo request [Up\\_IcmpMsgHandler\(\)](#) is not called for this message.

## 7.3 Internet Protocol Version 6

**[SWS\_Tcplp\_00376]** [Tcplp shall process a received Inverse Neighbor Discovery (IND) Solicitation message and respond with an Advertisement message, as described in IETF RFC 3122. The neighbor cache shall be updated with the values provided in the Solicitation message.]

**[SWS\_Tcplp\_00377]** [TLS shall be able to process X.509 v3 certificates and X.509 v2 certificate revocation list (CRL) profiles, as described in IETF RFC 5280.]

**[SWS\_Tcplp\_00378]** [If [TcpIpTcpWindowScaleOptionEnabled](#) is set to TRUE, the TCP window scale option (WSopt) and mechanism shall be supported according to IETF RFC 7323, chapter 2. The value given by [TcpIpTcpWindowScale](#) shall be the value transmitted in SYN message and the limiting factor when replying with SYN-ACK message.]

**[SWS\_Tcplp\_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\_Tcplp\_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\_Tcplp\_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\_Tcplp\_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\_Tcplp\_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\_Tcplp\_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\_Tcplp\_00160]** [The Tcplp shall support the basic IPv6 header and the initially defined IPv6 extension headers and options as defined in IETF RFC 8200 (Internet Protocol, Version 6 (IPv6) Specification).]

**[SWS\_Tcplp\_00161]** [The Tcplp shall support the reception and reassembly of fragmented IPv6 frames according to IETF RFC 8200 Section 4.5 Fragment Header.]

**[SWS\_Tcplp\_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\_Tcplp\_00232]** [The Tcplp shall fragment oversized IPv6 frames before transmission according to IETF RFC 8200 Section 4.5 Fragment Header.]

**[SWS\_Tcplp\_00162]**

*Upstream requirements:* [SRS\\_Eth\\_00092](#)

[The Tcplp shall support the section 2, IPv6 Addressing of IETF RFC 4291 (IP Version 6 Addressing Architecture) excluding Section 2.6. Anycast Addresses. Section 2.8 A Node's Required Addresses shall be limited to the node requirements for host only.]

**[SWS\_Tcplp\_00269]**

*Upstream requirements:* [SRS\\_Eth\\_00092](#)

[The Tcplp shall support the Section 2.6. Anycast Addresses of IETF RFC 4291 (IP Version 6 Addressing Architecture).]

### 7.3.2 Internet Control Message Protocol (ICMPv6)

**[SWS\_Tcplp\_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).]

**[SWS\_Tcplp\_00278]**

*Upstream requirements:* [SRS\\_Eth\\_00098](#)

[The Tcplp shall only reply to ICMPv6 Echo Request Messages if they are valid and [TcpIpIcmpV6EchoReplyEnabled](#) is set to TRUE.]

**[SWS\_Tcplp\_00298]**

*Upstream requirements:* [SRS\\_Eth\\_00098](#)

[If a [TcpIpIcmpV6MsgHandler](#) is configured, the Tcplp shall call the respective [Up\\_IcmpMsgHandler\(\)](#) if an ICMPv6 message is received and not handled by the Tcplp directly.]



Note: For example, if the Tcplp replies to an ICMPv6 echo request `Up_IcmpMsgHandler()` is not called for this message.

### 7.3.3 Neighbor Discovery Protocol (NDP)

#### [SWS\_Tcplp\_00164]

*Upstream requirements:* [SRS\\_Eth\\_00090](#)

[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)) except the sections 4.5 Redirect Message Format, 6.2. Router Specification, 7.2.8. Proxy Neighbor Advertisements and 8. Redirect Function.]

#### [SWS\_Tcplp\_00281]

*Upstream requirements:* [SRS\\_Eth\\_00090](#)

[The Tcplp shall support the handling of redirect messages as defined in IETF RFC 4861 (Neighbor Discovery for IP version 6 (IPv6)) Section 8.3. Host Specification.]

#### [SWS\_Tcplp\_00261]

*Upstream requirements:* [SRS\\_Eth\\_00111](#)

[If `TcpIpNdpDefensiveProcessing` is set to TRUE, the NDP shall silently discard all received Neighbor Advertisements that have not been requested by a previously transmitted Neighbor Solicitation.]

#### [SWS\_Tcplp\_00262]

*Upstream requirements:* [SRS\\_Eth\\_00111](#)

[If `TcpIpNdpDefensiveProcessing` is set to TRUE, the NDP shall skip the update of the Neighbor Cache upon processing received Neighbor Solicitations.]

#### [SWS\_Tcplp\_00263]

*Upstream requirements:* [SRS\\_Eth\\_00090](#)

[The Tcplp shall limit the number of neighbor cache entries to the number specified by the configuration parameter `TcpIpNdpMaxNeighborCacheSize` ([\[ECUC-Tcplp\\_00129\]](#))]

#### [SWS\_Tcplp\_00264]

*Upstream requirements:* [SRS\\_Eth\\_00090](#)

[In case the neighbor cache is full and a new entry shall be added, the Tcplp shall drop the oldest entry to be able to add the new entry]

**[SWS\_Tcplp\_00265]**

*Upstream requirements:* [SRS\\_Eth\\_00110](#)

[The Tcplp shall adhere to the rules defined in IETF RFC 5942 - Section 4 "Host Rules" and shall use the updated definition of "on-link" according to IETF RFC 5942 - Section 6 "Updates to RFC 4861".]

**[SWS\_Tcplp\_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 Internet Protocol Security (IPsec)

**[SWS\_Tcplp\_00352]**

*Upstream requirements:* [SRS\\_Eth\\_00142](#), [SRS\\_Eth\\_00143](#), [SRS\\_Eth\\_00144](#), [SRS\\_Eth\\_00145](#)

[Tcplp shall support IPsec according to AUTOSAR foundation RS\_IPsecProtocol [\[11\]](#).]

**[SWS\_Tcplp\_00353]**

*Upstream requirements:* [RS\\_IPSEC\\_00004](#), [RS\\_IPSEC\\_00021](#)

[IKEv2 shall be implemented according to IETF RFC 7296 and [\[RS\\_IPSEC\\_00021\]](#) with the limitations defined in [\[RS\\_IPSEC\\_00004\]](#). IKEv1 shall not be supported.]

Note: To ensure that IKEv2 is interoperable with the IETF IPsec standards in general and resolve any ambiguities, the open source IPsec implementation strongSwan ([strongswan.org](http://strongswan.org)) is used as reference.

**[SWS\_Tcplp\_00355]**

*Upstream requirements:* [RS\\_IPSEC\\_00010](#), [RS\\_IPSEC\\_00011](#), [RS\\_IPSEC\\_00013](#)

[The general IKEv2 connection configuration, e.g. connection lifetime and re-keying / re-authentication timeouts, dead peer detection, may be configured via the settings in the container "[IKEConnections](#)".]

**[SWS\_Tcplp\_00356]**

*Upstream requirements:* [RS\\_IPSEC\\_00014](#), [RS\\_IPSEC\\_00025](#)

[The IKEv2 certificates used for authentication with other IKEv2 nodes may be configured via the settings in the container "[IKECertificates](#)" and "[IKECertificate](#)".]

#### [SWS\_Tcplp\_00357]

*Upstream requirements:* [RS\\_IPSEC\\_00022](#), [RS\\_IPSEC\\_00023](#)

[The security policy database, which defines which connections shall be protected by IPsec and by which protections, may be configured via the settings in the container "[TcpIpSpdEntry](#)" and "[TcpIpIpSecPriority](#)". The [IpSecPriority](#) is used to establish the order in which the [SpdEntries](#) are checked. The first successful rule match will be executed, disregarding all lower priority rules.]

#### [SWS\_Tcplp\_00358]

*Upstream requirements:* [RS\\_IPSEC\\_00027](#)

[The priority of proposed algorithms for IKEv2 handshakes may be configured in the container "[IKEIkeSaProposal](#)".]

## 7.5 IP Based Protocols

### 7.5.1 Local Address Table

[SWS\_Tcplp\_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 [TcpIpLocalAddr](#) (including its subcontainers).]

Note: Each entry of the local IP address table is uniquely identified by the configuration parameter [TcpIpAddrId](#).

[SWS\_Tcplp\_00100] [In case no [TcpIpStaticIpAddressConfig](#) is provided, the Tcplp shall enable to specify a multicast IP address during runtime via [TcpIp\\_RequestIpAddrAssignment\(\)](#).]

[SWS\_Tcplp\_00130] [The Local IP address used for a socket is specified via [TcpIp\\_Bind\(\)](#).]

#### [SWS\_Tcplp\_00219]

*Upstream requirements:* [SRS\\_Eth\\_00087](#)

[If a [TcpIpAddrAssignment](#) 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 [TcpIpAddrAssignment](#). Once the procedure is complete, Tcplp shall store the new address in the [NvMBlock](#).]

## 7.5.2 User Datagram Protocol (UDP)

**[SWS\_Tcplp\_00060]** [The Tcplp shall implement the User Datagram Protocol (UDP) as defined in IETF RFC 768 (User Datagram Protocol).]

**[SWS\_Tcplp\_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).]

## 7.5.3 Transmission Control Protocol (TCP)

**[SWS\_Tcplp\_00373]** [If [TcpIpTcpSackEnabled](#) is set to TRUE, the Selective Acknowledgement (SACK) mechanism shall be supported according to IETF RFC 2018. If enabled, the SACK option shall be sent in the TCP handshake]

**[SWS\_Tcplp\_00061]** [The Tcplp shall implement the Transmission Control Protocol (TCP) as defined in IETF RFC 793 (Transmission Control Protocol)]

**[SWS\_Tcplp\_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"), 4.2.2.16 (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\_Tcplp\_00062]** [The Tcplp shall support the Window and Acknowledgment Strategies in TCP as defined in IETF RFC 1122:

- 4.2.3.2 When to Send an ACK Segment
- 4.2.3.3 When to Send a Window Update
- 4.2.3.4 When to Send Data.

]

## **[SWS\_Tcplp\_00390] TCP retransmission handling**

*Status:* DRAFT

[The Tcplp shall support the Retransmission Strategy with clamped exponential back-off with base 2 in TCP by using the configuration parameters [TcpIpTcpRetransmis-](#)

`sionTimeout`, `TcpIpTcpMaxRetransmissionTimeout` and `TcpIpTcpMaxRtx` as defined in ETF RFC 1122:

- 4.2.2.15 Retransmission Timeout
- 4.2.3.1 Retransmission Timeout Calculation

while excluding Jacobson's and Karn's algorithm for measuring round time trip (RTT) and calculation of the retransmission timeout (RTO) from it.]

#### [SWS\_Tcplp\_00063]

*Upstream requirements:* [SRS\\_Eth\\_00109](#)

[The Tcplp shall implement the Nagle Algorithm as defined in IETF RFC 1122: 4.2.3.4 When to Send Data.]

[SWS\_Tcplp\_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\_Tcplp\_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.]

## 7.5.4 Transport Layer Security (TLS)

[SWS\_Tcplp\_00374] [Tcplp shall be able to process the `BasicOCSPResponse` according to IETF RFC 6960, section 4.2.1, sent by a TLS server in the `CertificateStatus` handshake message.]

[SWS\_Tcplp\_00375] [If the parameter `TcpIpTlsRootCertUpdateCalloutFunction` is enabled, the callout function, provided in it's value and defined in the header file given by `TcpIpTlsRootCertUpdateCalloutHeaderFile`, shall be called once a new, valid, root certificate is received during TLS handshake.]

[SWS\_Tcplp\_00379] [If `TcpIpTlsUseExtensionMaxFragmentLength` is set to TRUE then the `max_fragment_length` extension shall be used to negotiate the max. fragment length between TLS server and client according to IETF RFC 6066, chapter 4. If `TcpIpTlsUseExtensionRecordSizeLimit` is set to TRUE, this extension must not be sent and must be ignored on reception.]

**[SWS\_Tcplp\_00380]** [If [TcpIpTlsUseExtensionTrustedCAKeys](#) is set to TRUE then the TLS client shall transmit the elements of [TcpIpTlsTrustedCAList](#) in its ClientHello message using the trusted\_ca\_keys extension according to IETF RFC 6066, chapter 6. Each element of [TcpIpTlsTrustedCAList](#) represents a KeyM root certificate and is referenced by [TcpIpTlsTrustedCAListEntry](#).]

**[SWS\_Tcplp\_00381]** [If [TcpIpTlsUseExtensionCertificateStatusRequest](#) is set to TRUE the status\_request\_v2 extension shall be supported according to IETF RFC 6961. The TLS Client shall transmit a CertificateStatusRequest containing the elements of [TcpIpTlsUseExtensionTrustedCAKeys](#) in its ClientHello message. If [TcpIpTlsUseExtensionTrustedCAKeys](#) contains more than one element, the CertificateStatusRequest shall be of type ocspl\_multi.]

### **[SWS\_Tcplp\_00300]**

*Upstream requirements:* [SRS\\_Eth\\_00138](#)

[Tcplp shall support the Transport Layer Security for TCP communication according to IETF RFC5246, at least chapters 7 and 8.]

At least those parts from IETF RFC5246 need to be implemented that are required for a basic and compatible interoperability with other nodes without any optional extensions.

**[SWS\_Tcplp\_00301]** [Further recommendation according to IETF RFC 7525 for a secure TLS implementation shall be considered.]

### **[SWS\_Tcplp\_00302]**

*Upstream requirements:* [SRS\\_Eth\\_00138](#)

[TLS connection requests with TLS version lower than 1.2 (IETF RFC5246) shall be disregarded respectively rejected with an alert. Thus, no backward compatibility handling to TLS versions lower than TLS 1.2 as described in IETF RFC5246, App. E shall be implemented or supported.]

**[SWS\_Tcplp\_00346]** [If the TLS connection references TlsCiphersuiteDefinition of type [TLS\\_VERSION\\_V13](#), then TLS V1.3 shall be the preferred protocol version. Only if this fails and ciphersuites for TLS V1.2 are also assigned to the TLS connection, then a downgrade operation to TLS V1.2 shall be allowed.]

Info: If the TLS connection does not contain ciphersuites for TLS V1.3, then the handshake shall be initiated indicating TLS V1.2 protocol.

**[SWS\_Tcplp\_00303]** [Session renegotiation shall be discarded by AUTOSAR TLS implementation.]

The KeyExchange algorithms as described in section 7.4.7 and section 8 of IETF RFC5246 depend on the ciphersuites. The necessary CSM jobs for key exchange are therefore referenced in the ciphersuite configuration.

**[SWS\_Tcplp\_00304]**

*Upstream requirements:* [SRS\\_Eth\\_00139](#)

[If ciphersuites for TLS include support for elliptic curves then mandatory parts of IETF RFC 4492 shall be supported accordingly.]

At least, the corresponding Key Exchange algorithms according to section 2 of IETF RFC 4492 have to be implemented such as ECDHE. Extensions according to section 5 only have to be supported if certificates with respective elliptic curve parameters are expected to be used.

**[SWS\_Tcplp\_00329]** [The TLS implementation must support at least one ciphersuite that corresponds to the DoIP specification ISO13400-2 so that an upper layer is able to connect such a socket to a diagnostic communication.]

**[SWS\_Tcplp\_00305]** [The TLS connection shall have a configuration parameter that defines if the socket is used for TLS client or TLS server communication from the node's perspective.]

**[SWS\_Tcplp\_00306]** [A TLS connection that is used for TLS server requires a reference to a local certificate with its private key.]

In the configuration, TLS connections can be collected in [TcpIpTlsConnection-Group](#). If one TLS connection in a group is already active, another TLS connection of the same group shall not be activated. In other words, only one TLS connection of a group shall be active at the same time. This allows to define exclusive resources for a TLS connection group and resources for TLS connections in the same group can be shared.

**[SWS\_Tcplp\_00315]** [A TLS Server shall request client authentication if the selected TLS connection is configured accordingly (i.e. the config parameter [TcpIpTlsUse-ClientAuthenticationRequest](#) is set to TRUE). In this case, a local certificate with its private key is also required for a TLS client and shall be provided to the server on demand during the TLS handshake.]

**[SWS\_Tcplp\_00349]** [If [TcpIpTlsUseExtensionRecordSizeLimit](#) is set to TRUE then the record\_size\_limit extension shall be used to negotiate the max. fragment length between TLS server and client according to IETF RFC 8449, chapter 4.1.]



The assignment of TLS connections to TCP sockets is either based on static configuration (static TLS connection assignment) or done dynamically by means of an API call (dynamic TLS connection assignment).

**[SWS\_Tcplp\_00307]** [In dynamic TLS connection assignment a TLS connection shall be assigned to a TCP socket through a function call to `TcpIp_ChangeParameter()` with the ParameterId `TCPIP_PARAMID_TLS_CONNECTION_ASSIGNMENT`. The ParameterValue of the function provides a reference to a TLS connection for this socket.]

Note: A typical approach to dynamically assign a TLS connection to a socket is during the channel set-up before a socket connection has been established. However, it shall also be possible to perform this operation after the socket connection has been established. This might be useful starting with plain text communication and later on switching to TLS encrypted communication to accomplish for e.g. a STARTTLS operation.

**[SWS\_Tcplp\_00337]** [For dynamic TLS connection assignment via `TcpIp_ChangeParameter()`, the call to `TcpIp_ChangeParameter()` shall initiate the TLS handshake as follows:

- a TLS Server shall wait for a ClientHello as the next message on this socket.
- a TLS Client shall start sending a ClientHello message.
- after that Tcplp shall no longer pass on plain messages to upper or lower layer but pass it on to TLS.

]

The successful completion of the TLS handshake is signaled according to [\[SWS\\_Tcplp\\_00345\]](#).

**[SWS\_Tcplp\_00308]** [For static TLS connection assignment a port and optionally an address is defined for at least one TLS connection, TCP shall check during TCP SYN (either reception or transmission of SYN) if a port assignment is available for any TLS connection and if this TLS connection is not in use. If so, the TCP shall check the ports and automatically assign this TLS connection to the socket if a port matches.]

**[SWS\_Tcplp\_00343]** [For static TLS connection assignment the TCP client shall check its remote port configuration when the SYN frame will be transmitted. If the TLS port configuration matches it shall assign the corresponding TLS connection to the socket.]

Note: This approach rules out use cases where one client uses different TLS settings (including not using TLS at all) for different local sockets when connecting to the same



remote listening socket. However, having one client connecting to the same remote listening socket via different local sockets using different TLS settings is deemed an exotic use case and is thus deliberately not supported.

**[SWS\_Tcplp\_00344]** [For static TLS connection assignment the TCP server shall check its local port configuration when the SYN frame is received. If the TLS port configuration matches it shall assign the corresponding TLS connection to the socket.]

Note: This approach rules out use cases where one server uses different TLS settings (including not using TLS at all) for different remote sockets but the same local listening socket. However, having one server using different TLS settings for different clients with the same listening socket is deemed an exotic use case and is thus deliberately not supported.

**[SWS\_Tcplp\_00336]** [For static TLS connection assignment the TCP client shall initiate the TLS handshake if a TLS connection is assigned to the socket after the SYN ACK has been transmitted successfully.]

**[SWS\_Tcplp\_00309]** [For static TLS connection assignment at the TCP client the interface `<Up_TcpConnected>` shall not be called after sending the ACK of the SYN to the server. Instead, this function shall be called after the TLS handshake has been finished successfully.]

**[SWS\_Tcplp\_00328]** [For static TLS connection the TCP server shall expect a TLS handshake after the ACK for the SYN has been received. All incoming messages for this socket shall further be passed on to TLS.]

**[SWS\_Tcplp\_00310]** [For static TLS connection assignment at the TCP server side the interface `<Up_TcpAccepted>` shall not be called after the ACK has been received. Instead, this function shall be called after the TLS handshake has been finished successfully.]

**[SWS\_Tcplp\_00345]** [For both dynamic and static TLS connection assignment, the socket owner shall be informed with `<Up_TcpIpEvent>()` and the event type `TCPIP_TLS_HANDSHAKE_SUCCEEDED` if an event callback is defined for a socket owner and the TLS handshake has been finished successfully. For static TLS connection assignment the call to `<Up_TcpIpEvent>` and the event type `TCPIP_TLS_HANDSHAKE_SUCCEEDED` shall take place after the call to `<Up_TcpAccepted>/<Up_TcpConnected>`.]

**[SWS\_Tcplp\_00311]**

*Upstream requirements:* [SRS\\_Eth\\_00134](#)

[A TLS server shall select the locally assigned ciphersuite with the highest priority that matches with one of the received ciphersuites. The local certificate that was assigned to this combination of TLS connection and TLS ciphersuite shall be provided during the handshake.]

**[SWS\_Tcplp\_00316]** [The TLS SERVER shall provide the certificate referenced by [TcpIpTlsConnection](#)/ [TcpIpTlsCipherKeyMLocalCertificate](#) through the `server_certificate` message. The certificate shall be requested from the Key Manager with the function `KeyM_GetCertificate()`.]

**[SWS\_Tcplp\_00338]** [If a certificate is received with the certificate or `certificateVerify` handshake message of TLS it shall be provided to the Key Manager using the function `KeyM_SetCertificate()` with the reference [TcpIpTlsCipherKeyMRemoteCertificate](#) of [TcpIpTlsConnection](#). Afterwards, the certificate is verified using the function `KeyM_VerifyCertificate()` or, if more than one certificate has been received with the handshake message, with the function `KeyM_VerifyCertificateChain()`. This function also uses the [TcpIpTlsCipherKeyMRemoteCertificate](#) reference.]

The TLS module uses CSM jobs that are assigned to the ciphersuite to perform the cryptographic operations. The key material will be negotiated and loaded during the handshake.

Note: CSM jobs can run synchronously or asynchronously. If a job shall run in asynchronous or synchronous mode depends on its configuration. For asynchronous jobs a callback is needed which are not defined in this document. They are vendor specific and shall be configured accordingly in the CSM as documented.

**[SWS\_Tcplp\_00339]** [TLS shall use the CSM job referenced by [TcpIpTlsCsmRandomGenerateJobRef](#) referenced by [TcpIpTlsHandshake](#) and referenced in the [TcpIpTlsConnection](#) to generate random values. The system outside the TLS is responsible to collect entropy to seed the RNG if needed.]

**[SWS\_Tcplp\_00340]** [After selection of the ciphersuite the assigned [TcpIpTlsHandshake](#) of the TLS connection will provide all necessary references to CSM jobs and keys necessary to accomplish the key exchange algorithms.]

Info: Not all CSM jobs referenced in the [TcpIpTlsHandshake](#) container are required. Which of the jobs and keys configured for a TLS handshake are needed for operation mainly depends on the ciphersuite and its associated certificate. They must be pre-configured and assigned accordingly. It also depends on the TLS type if it is a TLS Server or a TLS Client, which ciphersuites are assigned to the TLS connections and

which public key type is contained in the certificate, i.e. if it is an ECC or RSA public key.

The following table provides an overview of jobs and keys for CSM that needs to be configured for the handshake operation:

Job type	RSA	ECC
TcpIpTlsCsmPrfMac[Job Key]Ref	C/S	C/S
TcpIpTlsCsmHashVerifyJobRef	C/S	C/S
TcpIpTlsCsmMasterSecretKeyRef	C/S	C/S
TcpIpTlsCsmKeyExchangeCalcPubValJobRef	-	C/S <sup>1</sup>
TcpIpTlsCsmKeyExchangeKeyRef	-	C/S <sup>2</sup>
TcpIpTlsCsmKeyExchangeCalcSecretJobRef	-	C/S <sup>1</sup>
TcpIpTlsCsmKeyExchangeSignatureGenerate[Job Key]Ref	-	S/B
TcpIpTlsCsmKeyExchangeSignatureVerify[Job Key]Ref	-	C/B
TcpIpTlsCsmKeyExchangeEncrypt[Job Key]Ref	C/B	-
TcpIpTlsCsmKeyExchangeDecrypt[Job Key]Ref	S/B	-

C: TLS Client implementation

S: TLS Server implementation

B: Additionally required if client authentication is activated.

<sup>1</sup> Reference is used for asynchronous DH(E) operation.

<sup>2</sup> Reference is used for synchronous DH(E) operation.

The following examples can be used as a guideline.

### Example #1:

A ciphersuite that references RSA provides [TcpIpTlsCsmKeyExchangeEncryptJobRef](#) for the TLS client to encrypt the pre-master secret. First, the TLS client verifies the received certificate, will take the public key and copy it into the CSM key location referenced by [TcpIpTlsCsmKeyExchangeEncryptKeyRef](#). Then encrypts the pre-master secret and send it to the TLS server. The Server uses [TcpIpTlsCsmKeyExchangeDecryptJobRef](#) to decrypt the pre-master secret. The job either references statically the private key or, if [TcpIpTlsConnection/ TcpIpTlsCipherKeyMLocalCertificate/ KeyM CertPrivateKeyStorageCryptoKeyRef/ KeyMCryptoKeyCsmKeyTargetRef](#) is available, copy this key into [TcpIpTlsCsmKeyExchangeDecryptKeyRef](#).

### Example #2:

A ciphersuite references `ECDHE_ECDSA` and the used certificate contains appropriate ECC keys, ECDSA capable in this case. The server generates DH-parameter

using the crypto job `Csm_KeyExchangeCalcPubVal()` using the reference to `TcpIpTlsCsmKeyExchangeKeyRef` and signs the result using `TcpIpTlsHandshake/ TcpIpTlsCsmKeyExchangeSignatureGenerateKeyRef` holding a reference to the certificate private key. If the key is not statically assigned to the job it must be copied accordingly (see Example #1). The resulting data is sent to the TLS client, who verifies the certificate and uses the key of the certificate to verify the provided ECDSA signature from the server using `TcpIpTlsHandshake/ TcpIpTlsCsmKeyExchangeSignatureVerifyKeyRef`. Afterwards, if successful, calculates its own DH parameter and provides this to the server. Both, TLS client and server will then calculate the pre-master secret using `Csm_KeyExchangeCalcSecret()`.

### Example #3:

The selected ciphersuite defines a pre-shared key according to IETF RFC 4279. The server provides the `psk_identity_hint` in the `ServerKeyExchange` message. This can either be derived from the `TcpIpTlsPskIdentity/ TcpIpTlsPresharedKeyIdentityHint` or, if not specified, it can be queried from the user callback `TcpIpTlsPskGetKeyIdentityHintFunc`. The TLS client uses the hint to select a pre-shared key that is known by both the TLS Client and this TLS Server. If one key can uniquely be identified with the identity hint, then the `TcpIpTlsPskIdentity` configuration can be used as an alternative to the callback functions. In this case, the selected key can be determined by `TcpIpTlsPresharedKeyIdentityHint` and the `TcpIpTlsPresharedKeyIdentity` with `TcpIpTlsPresharedKeyCsmKeyRef` can be used further. A more flexible solution provides the usage of the callback `TcpIpTlsPskGetClientKeyIdentityFunc` that allows the selection of a key with its identity at runtime. After the key and its identity has been selected on the client side, the `psk_identity` will be provided back to the TLS server through the `ClientKeyExchange` message. On the TLS server side, the corresponding key can be identified in the same way, either through the static configuration of `TcpIpTlsPskIdentity/ TcpIpTlsPresharedKeyIdentity` or can be queried through a callback function determined by `TcpIpTlsPskGetServerKeyIdentityFunc` on server side. After the key has been selected, the master secret can be determined with the corresponding CSM jobs that are allocated in the `TcpIpTlsHandshake` container.

**[SWS\_Tcplp\_00341]** [TLS shall use `TcpIpTlsHandshake /TcpIpTlsCsmHashVerifyJobRef` to calculate the hash over the handshake messages which is provided with the finish handshake message.]

**[SWS\_Tcplp\_00347]** [TLS shall use `TcpIpTlsCsmPrfMacJobRef` to calculate the master secret. The configuration item `TcpIpTlsCsmPRFSupportType` shall specify how the CSM job supports the generation of the master secret.]

If `TcpIpTlsCsmPRFSupportType` is set to `TLS_PRF_CSM_NO_SUPPORT` then `TcpIpTlsCsmPrfMacJobRef` references a job for MAC generation. If it

is set to `TLS_PRF_CSM_INOUT_REDIRECT_SUPPORT`, then the re-direction support mentioned below shall be used. If the configuration is set to `TLS_PRF_CSM_FULL_SUPPORT` then the CSM job will generate the master secret completely on its own. The TLS just need to call the job and the master secret will be available in the element ID #1 of `TcpIpTlsCsmMasterSecretKeyRef`. A key distribution to the worker jobs must be done in any case.

It is recommended to use input and output re-direction for the `TcpIpTlsCsmPrfMacJobRef`, that was introduced in CSM with AUTOSAR V4.4. This allows to leave the master secret and intermediate results of the calculation within the crypto driver (e.g. in HSM). The key elements of `TcpIpTlsCsmPrfMacKeyRef` is used for input and `TcpIpTlsCsmMasterSecretKeyRef` as output reference for this job. `Csm_KeyElementSet()` is used for initial value settings, `Csm_KeyCopy()` and `Csm_KeyElementCopyPartial()` are used to set-up the input values for the job operation. `Csm_KeyElementCopyPartial()` is finally used to distribute the master secret results to the `TcpIpTlsCiphersuiteWorker` key references that are used by the worker jobs during application data transmission.

**[SWS\_Tcplp\_00312]** [If `TcpIpTlsServerNameIdentification` is configured for a TLS connection the configured name shall be added to the Client Hello message as the server name identification (SNI).]

**[SWS\_Tcplp\_00313]** [If a TLS server receives a ClientHello message that contains a server name identification with length greater than 0 the server shall search in `TcpIpTlsCertificateIdentity` for a matching identity reference and shall provide the certificate that is located in this container during the handshake.]

**[SWS\_Tcplp\_00314]** [The time stamp information that is contained in the ClientHello message shall be provided through the configured `TcpIpTlsConnectionGetTimeFunc` callout function.]

**[SWS\_Tcplp\_00325]**

*Upstream requirements:* `SRS_Eth_00141`, `SRS_Eth_00137`

[If a ciphersuite is used for pre-shared keys and `TcpIpTlsUsePresharedKeys` is set to TRUE, callback functions shall provide the necessary information on the TLS client and the TLS server side to select the pre-shared keys according to IETF RFC 4279. The callbacks are used to provide the identity hint and eventually the key identification during the handshake. The callback functions are used to select the CSM key that is used for further processing. Alternatively, if callback functions are not configured, the static parameter configuration from `TcpIpTlsPskIdentity` can be used.]

**[SWS\_Tcplp\_00326]**

*Upstream requirements:* [SRS\\_Eth\\_00135](#)

[TLS shall be able to open and maintain a maximum number of connections as defined in [TcpIpTlsMaxConnections](#).]

**[SWS\_Tcplp\_00327]**

*Upstream requirements:* [SRS\\_Eth\\_00136](#)

[TCP data streams shall be segmented by TLS into fragments. The maximum size of a fragment shall be used as configured in [TcpIpTlsMaxFragmentLength](#). A TCP socket must be able to transmit at least such a fragment within one segment.]

**[SWS\_Tcplp\_00348]** [On reception of a TLS "close\_notify" message the TLS connection shall be closed and all security related resources shall be destroyed. It shall not be possible to perform further plain text communication through TCP on this socket after the TLS connection was closed. Thus, it is recommended to close the TCP socket, too.]

## 7.5.5 Dynamic Host Configuration Protocol

**[SWS\_Tcplp\_00200]**

*Upstream requirements:* [SRS\\_Eth\\_00088](#)

[The server part of the Dynamic Host Configuration Protocol shall be pre compile time configurable ON/OFF by the configuration parameter [TcpIpDhcpServerEnabled](#) (see [[ECUC\\_Tcplp\\_00183](#)])]

**[SWS\_Tcplp\_00201]**

*Upstream requirements:* [SRS\\_Eth\\_00087](#)

[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 [TcpIpCtrl](#).]

**[SWS\_Tcplp\_00218]**

*Upstream requirements:* [SRS\\_Eth\\_00087](#)

[If the configuration contains [TcpIpDhcpAddressAssignment](#) 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 [TcpIpDhcpAddressAssignment](#) related to the same port.]

#### 7.5.5.1 Dynamic Host Configuration Protocol (DHCPv4)

##### [SWS\_Tcplp\_00058]

*Upstream requirements:* [SRS\\_Eth\\_00087](#), [SRS\\_Eth\\_00088](#)

[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).]

**[SWS\_Tcplp\_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.]

#### 7.5.5.2 Dynamic Host Configuration Protocol (DHCPv6)

**[SWS\_Tcplp\_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\_Tcplp\_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 sec-



tion 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.]

## 7.6 Message Reception

### [SWS\_Tcplp\_00411] Reception parameters derived from PDU

*Status:* DRAFT

*Upstream requirements:* [SRS\\_Eth\\_00187](#)

[If the packet is received, the Tcplp module shall derive the frame type (`EthIfFrameType`) and the `EthIfCtrl` (`EthIfController`) configured in `EthIf` via the PDU that is referenced by the `TcpIpCtrlRxPdu` which is identified by the given `RxPduId`.]

### [SWS\_Tcplp\_00412] Reception parameters derived from meta data items

*Status:* DRAFT

*Upstream requirements:* [SRS\\_Eth\\_00187](#)

[If the packet is received, the Tcplp module shall consume meta data items `PduInfoPtr.MetaDataPtr` configured at the `TcpIpCtrlRxPdu` that corresponds to the given `RxPduId` in the following order:

- `ETHERNET_MAC_64` indicating the Physical source address (MAC address in network byte order)
- `BROADCAST_8` indicating a broadcast frame

]

### [SWS\_Tcplp\_CONSTR\_00001] Reception PDU constraint for keeping the local buffer

*Status:* DRAFT

[Each `TcpIpCtrlRxPdu` shall refer to global PDU that has `KeepLocalPduBuffer` set to `FALSE`.]

**[SWS\_Tcplp\_00169]** [The Tcplp IP-layer shall map received IP datagrams to an entry in the local address table (`TcpIpAddrId`).

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

1. The receiving interface matches the interface assigned to the local address table entry (`EthIfCtrl`).
2. 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\_Tcplp\_00359]** [If IPsec is has been configured, all received IP datagrams shall be mapped to a Security Policy entry and processed as below:

1. **TCPIP\_IPSEC\_POLICY\_PROTECT:**

The IP datagram is only forwarded to the upper layer if it contains a valid Authentication header as per IETF RFC 4302. Otherwise the IP Datagram shall be dropped and optional callback invoked.

2. **TCPIP\_IPSEC\_POLICY\_BYPASS:**

The IP datagram is forwarded to the upper layer without any IPsec processing.

3. **TCPIP\_IPSEC\_POLICY\_DISCARD:**

The IP datagram shall be dropped without any IPsec processing.

]

**[SWS\_Tcplp\_00260]**

*Upstream requirements:* [SRS\\_Eth\\_00111](#)

[All IP datagrams mapped to an IPv6 entry in the local address table, configured with the optional [TcpIpLocalAddrIPv6ExtHeaderFilterRef](#) ([\[ECUC\\_Tcplp\\_00200\]](#)), that contains at least one IPv6 extension header not listed in the referenced [TcpIpIPv6ConfigExtHeaderFilter](#) ([\[ECUC\\_Tcplp\\_00198\]](#)) shall be silently discarded. If the Ipv6 entry in the local address table is not configured with the optional [TcpIpLocalAddrIPv6ExtHeaderFilterRef](#), then this frame shall be processed.]

**[SWS\_Tcplp\_00170]** [The Tcplp 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:

- The socket is bound to the local address ([TcpIpAddrId](#))
- The socket local address uses the wildcard "ANY" AND the socket [EthIfCtrl](#) is identical to the [EthIfCtrl](#) used in the local address ([TcpIpAddrId](#))
- 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\_Tcplp\_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\_Tcplp\_00172]** [The Tcplp 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:

- The socket is bound to a unicast local address ([TcpIpAddrId](#))
- The socket local address uses the wildcard "ANY" AND the socket `EthIfCtrl` is identical to the used in the local address ([TcpIpAddrId](#))
- The socket is bound to `TCPIP_LOCALADDRID_ANY`

]

**[SWS\_Tcplp\_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 [\[SWS\\_Tcplp\\_00172\]](#).]

**[SWS\_Tcplp\_00174]** [Received TCP datagrams where the local address ([TcpIpAddrId](#)) is a broadcast or multicast address, shall be silently discarded.]

### **[SWS\_Tcplp\_00266]**

*Upstream requirements:* [SRS\\_Eth\\_00111](#)

[If the filtering of TCP options has been enabled on a socket via [TcpIp\\_ChangeParameter\(\)](#), the Tcplp shall check received segments against the allowed list of options ([\[ECUC\\_Tcplp\\_00202\]](#) [TcpIpTcpConfigOptionFilter](#)) and if it contains at least one TCP option not listed the segment shall be silently discarded.]

**[SWS\_Tcplp\_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:

- for IPv4 frames if IPv4 checksum verification in hardware is enabled, i.e. `EthCtrlEnableOffloadChecksumIPv4` is set to TRUE
- for ICMP frames if ICMP checksum verification in hardware is enabled, i.e. `EthCtrlEnableOffloadChecksumICMP` is set to TRUE

- for TCP frames if TCP checksum verification in hardware is enabled, i.e. `EthCtrlEnableOffloadChecksumTCP` is set to TRUE
- 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.]

#### [SWS\_Tcplp\_00279]

*Upstream requirements:* [SRS\\_Eth\\_00019](#)

[For receptions the `Tcplp` Module shall accept UDP datagrams containing a zero checksum only on sockets that have been configured accordingly (i.e. `TcpIp_ChangeParameter()` has been called with `TCPIP_PARAMID_UDP_CHECKSUM` set to FALSE).]

#### [SWS\_Tcplp\_00296]

*Upstream requirements:* [SRS\\_Eth\\_00129](#)

[If the measurement data is enabled (see [TcpIpGetAndResetMeasurementDataApi](#)), `Tcplp` shall increment the corresponding measurement data whenever a received datagram is discarded.]

The following guidelines are recommended for TLS data handling:

- If a TCP datagram is accepted and the socket is assigned to a TLS connection, TCP should pass the data to TLS for further processing.
- If a received TLS application message was successfully processed and verified, the data contents should be passed back to TCP to further provide it to the configured upper layer. This provides full transparency of data reception to the upper layer.
- If message reception is passed on to TLS but cannot be processed, because a TLS connection has not yet been established or the message cannot be authenticated and/or decrypted correctly, the message should be dropped.
- After TLS has processed a message and all data has been consumed completely, TCP should be notified to release all related resources for this message, regardless if the message was processed successfully or not.

## 7.7 Message Transmission

[SWS\_Tcplp\_00175] [If data is transmitted using a socket which is bound to an IPv4 Unicast local address ([TcpIpAddrId](#)) the `Tcplp` shall use the IP address assigned to

the local address ([TcpIpAddrId](#)) as source IP address in the IP datagram header. The IP datagram shall be transmitted using the `EthIfCtrl` the local address ([TcpIpAddrId](#)) is mapped to.]

**[SWS\_Tcplp\_00176]** [If data is transmitted using an IPv4 socket which is bound to a local address ([TcpIpAddrId](#)) using the wildcard "ANY", then the Tcplp shall use the IP address of the configured local address ([TcpIpAddrId](#)), which is of type IPv4 Unicast and assigned to the same `EthIfCtrl`, as the bound local address ([TcpIpAddrId](#)) as source IP address in the IP datagram header.]

**[SWS\_Tcplp\_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 ([TcpIpAddrId](#)), 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 ([TcpIpAddrId](#)) of `EthIfCtrl = 0` is selected.]

**[SWS\_Tcplp\_00178]** [If data is transmitted using an IPv4 UDP socket which is bound to a local address ([TcpIpAddrId](#)) of type Multicast, then the Tcplp shall use the IP address of the configured local address ([TcpIpAddrId](#)), which is of type IPv4 Unicast and assigned to the same `EthIfCtrl`, as the bound local address ([TcpIpAddrId](#)) as source IP address in the IP datagram header.]

**[SWS\_Tcplp\_00179]** [If data is transmitted using an IPv4 UDP socket which is bound to a local address ([TcpIpAddrId](#)) of type Broadcast, then the Tcplp shall use the IP address of the configured local address ([TcpIpAddrId](#)), which is of type IPv4 Unicast and assigned to the same `EthIfCtrl`, as the bound local address ([TcpIpAddrId](#)) as source IP address in the IP datagram header.]

**[SWS\_Tcplp\_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 ([TcpIpAddrId](#)), 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 ([TcpIpAddrId](#)) of `EthIfCtrl = 0` is selected.]

**[SWS\_Tcplp\_00181]** [If data is transmitted using a socket which is bound to an IPv6 Unicast local address ([TcpIpAddrId](#)) the Tcplp shall use the IP address assigned to local address ([TcpIpAddrId](#)) as source IP address in the IP datagram header. The IP datagram shall be transmitted using the `EthIfCtrl` the local address ([TcpIpAddrId](#)) is mapped to.]

**[SWS\_Tcplp\_00182]** [If data is transmitted using an IPv6 socket which is bound to a local address ([TcpIpAddrId](#)) 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 ([TcpIpAddrId](#)) on the same `EthIfCtrl` as the bound local address ([TcpIpAddrId](#)) only.]

**[SWS\_Tcplp\_00183]** [If data is transmitted using an IPv6 socket which is bound to `TCP_IP_LOCALADDRID_ANY`, the `Tcplp` shall select the interface that has a local address ([TcpIpAddrId](#)) 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\_Tcplp\_00184]** [If data is transmitted using an IPv6 UDP socket which is bound to a local address ([TcpIpAddrId](#)) 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 ([TcpIpAddrId](#)) on the same `EthIfCtrl` as the bound local address ([TcpIpAddrId](#)) only.]

**[SWS\_Tcplp\_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 ([TcpIpAddrId](#)) 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\_Tcplp\_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\_Tcplp\_00191]** [If the parameter [TcpIpArpPacketQueueEnabled](#) 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 and accept the transmission request with `E_OK`.]

**[SWS\_Tcplp\_00192]** [If the parameter [TcpIpArpPacketQueueEnabled](#) 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\_Tcplp\_00193]** [If the parameter [TcpIpNdpPacketQueueEnabled](#) 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 and accept the transmission request with `E_OK`.]

**[SWS\_Tcplp\_00194]** [If the parameter `TcpIpNdpPacketQueueEnabled` 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\_Tcplp\_00391] TCP retransmission time change**

*Status:* DRAFT

[If `TcpIpChangeParameter()` is called with the parameter `TCPIP_PARAMID_TCP_RETRANSMIT_TIMEOUT`, it shall override the `TcpIpTcpRetransmissionTimeout` value for a given socket.]

### **[SWS\_Tcplp\_00392] TCP maximum retransmission time change**

*Status:* DRAFT

[If `TcpIpChangeParameter()` is called with the parameter `TCPIP_PARAMID_TCP_MAX_RETRANSMIT_TIMEOUT`, it shall override the `TcpIpTcpMaxRetransmissionTimeout` value for a given socket.]

### **[SWS\_Tcplp\_00393] TCP maximum number of retransmissions change**

*Status:* DRAFT

[If `TcpIpChangeParameter()` is called with the parameter `TCPIP_PARAMID_TCP_MAXRTX`, it shall override the `TcpIpTcpMaxRtx` value for a given socket.]

**[SWS\_Tcplp\_00202]** [After the maximum retries configured via [\[ECUC\\_Tcplp\\_00069\]](#) and [\[SWS\\_Tcplp\\_00393\]](#) are transmitted, the timer according to

- either the clamped exponential backoff if enabled via [\[ECUC\\_Tcplp\\_00068\]](#), [\[SWS\\_Tcplp\\_00391\]](#), [\[ECUC\\_Tcplp\\_00340\]](#) and [\[SWS\\_Tcplp\\_00392\]](#) or
- static via [\[ECUC\\_Tcplp\\_00068\]](#) and [\[SWS\\_Tcplp\\_00391\]](#)

shall be restarted the last time before the TCP connection is closed.]

**[SWS\_Tcplp\_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:

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



- for not fragmented ICMP frames if ICMP checksum calculation in hardware is enabled, `EthCtrlEnableOffloadChecksumICMP` is set to TRUE
- for TCP frames if TCP checksum calculation in hardware is enabled, `EthCtrlEnableOffloadChecksumTCP` is set to TRUE
- 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.]

#### [SWS\_Tcplp\_00280]

*Upstream requirements:* [SRS\\_Eth\\_00019](#)

[For transmissions the `Tcplp` Module shall skip the calculation of the UDP protocol checksum and use the value zero instead, on sockets that have been configured accordingly (i.e. `TcpIp_ChangeParameter()` has been called with `TCPIP_PARAMID_UDP_CHECKSUM` set to FALSE).]

#### [SWS\_Tcplp\_00267]

*Upstream requirements:* [SRS\\_Eth\\_00097](#)

[Per default or if `TcpIp_ChangeParameter()` with `ParameterId` set to `TCPIP_PARAMID_PATHMTU_ENABLE` and the value set to TRUE has been called for a socket, the maximum size for outbound datagrams from this socket shall be determined by the Path MTU discovery.]

#### [SWS\_Tcplp\_00268]

*Upstream requirements:* [SRS\\_Eth\\_00097](#)

[If `TcpIp_ChangeParameter()` with `ParameterId` set to `TCPIP_PARAMID_PATHMTU_ENABLE` and the value set to FALSE has been called for a socket, the maximum size for outbound datagrams from this socket is determined by the static configuration.]

**[SWS\_Tcplp\_00320]** [If transmission is requested from upper layer to TCP and the connection is configured for TLS but the handshake has not yet been started or completed, the message transmission request shall return `E_NOT_OK`.]

**[SWS\_Tcplp\_00360]** [If IPsec has been configured, each IP datagram to be sent by `Tcplp` shall be mapped to a Security Policy entry and processed as following:

- **TCPIP\_IPSEC\_POLICY\_PROTECT:**

Authentication header as per IETF RFC 4302 shall be inserted after the IP header.

- **TCPIP\_IPSEC\_POLICY\_BYPASS:**

The IP datagram is transmitted without any IPsec processing.

- **TCPIP\_IPSEC\_POLICY\_DISCARD:**

The IP datagram shall be dropped.

]

**[SWS\_Tcplp\_00363]** [If [TcpIp\\_IsConnectionReady\(\)](#) is called and a security association is configured, the module shall:

- check if socket exists and is bound to an assigned local address.
- check if the provided remote address has a corresponding physical address.
- check if a security association is established for this socket.

If all checks are successful, the function shall return `TCPIP_E_OK`.]

**[SWS\_Tcplp\_00365]** [If [TcpIp\\_IsConnectionReady\(\)](#) is called and a security association is not configured, the module shall:

- check if socket exists and is bound to an assigned local address.
- check if the provided remote address has a corresponding physical address.

If all checks are successful, the function shall return `TCPIP_E_OK`.]

**[SWS\_Tcplp\_00366]** [If [TcpIp\\_IsConnectionReady\(\)](#) is called and the socket is not bound to an assigned local address, the function shall return `TCPIP_E_NOT_OK`.]

**[SWS\_Tcplp\_00367]** [If [TcpIp\\_IsConnectionReady\(\)](#) is called and the provided remote address has no corresponding physical address, Tcplp shall start the address resolution (if not already started) and return `TCPIP_E_PENDING`.]

**[SWS\_Tcplp\_00368]** [If [TcpIp\\_IsConnectionReady\(\)](#) is called and for the socket a security association is configured but not established:

- If the security association establishment is in progress, Tcplp shall return `TCPIP_E_PENDING`.
- If the security association establishment is not started and the security association allows to initiate the secure connection, Tcplp shall start establishment and return `TCPIP_E_PENDING`.
- If the security association establishment is not started and the security association does not allow to initiate the secure connection, Tcplp shall return `TCPIP_E_NOT_OK`.



]

The transmission request towards the LSduR could be performed either with direct data provision or with indirect data provision. With direct data provision, the data for transmission is forwarded in one single call via the LSduR to the lower layer. The lower layer is responsible to transfer the data to a transmit buffer. With indirect data provision, a transmission request is forwarded to the lower layer with data pointer set to `NULL_PTR` and data length set to length of data to be transmitted. The data pointer set to `NULL_PTR` indicated the lower layer to initiate the transmission by calling the `TriggerTransmit` function via the LSduR. In context of the `TcpIp_TriggerTransmit`, data is copied from the upper layer to the given data pointer to the lower layer. Both approaches perform a transmission. The usage of the approaches is implementation specific. The following recommendations are guidelines, but do not force an implementation to follow this:

- If `TcpIp_UdpTransmit()` with `DataPtr` set to `NULL_PTR`, then use `LSduR_TcpIpTransmit()` with data pointer set to `NULL_PTR`
- If `TcpIp_UdpTransmit()` with `DataPtr` set to data pointer, then use `LSduR_TcpIpTransmit()` with data pointer set
- If TcpIp segment which resides in `TcpIpBufferMemory` where the buffer is split, then use `LSduR_TcpIpTransmit()` with data pointer set to `NULL_PTR`
- If TcpIp segment which resides in `TcpIpBufferMemory` where the buffer is a linear buffer, then use `LSduR_TcpIpTransmit()` with data pointer set

#### [SWS\_Tcplp\_00401] Transmission request with direct data provision

*Status:* DRAFT

*Upstream requirements:* [SRS\\_Eth\\_00187](#)

[If there is a transmission request and a direct data provision is used to forward the given data to LSduR with the given data length, then the Tcplp module shall call `LSduR_TcpIpTransmit()` with the following arguments:

- `TxPduId` equal to the PDU id that is referenced by the used `TcpIpCtrlTxPdu`
- `PduInfoPtr.SduDataPtr` equal to the data pointer
- `PduInfoPtr.SduLength` equal to the data length
- `PduInfoPtr.MetaDataPtr` equal to the pointer of the created `MetaDataItem` configured at the `TcpIpCtrlTxPdu` that corresponds to the given `TxPduId`.

]

**[SWS\_Tcplp\_00402] Transmission request with indirect data provision***Status:* DRAFT*Upstream requirements:* [SRS\\_Eth\\_00187](#)

[If there is a transmission request and an indirect data provision is used to forward the given data to LSduR with the given data length, then the Tcplp module shall call LSduR\_TcpIpTransmit() with the following arguments:

- TxPduId equal to the PDU id that is referenced by the used [TcpIpCtrlTxPdu](#)
- PduInfoPtr.SduDataPtr equal to NULL\_PTR
- PduInfoPtr.SduLength equal to the data length
- PduInfoPtr.MetaDataPtr equal to the pointer of the created MetaDataItem configured at the [TcpIpCtrlTxPdu](#) that corresponds to the given TxPduId.

]

**[SWS\_Tcplp\_00403] Meta data handling while containing headers***Status:* DRAFT*Upstream requirements:* [SRS\\_Eth\\_00187](#)

[If the Tcplp module provides headers necessary for the transmission through PduInfoPtr.MetaDataPtr then MetaDataItem shall be set in the following order:

- ETHERNET\_MAC\_64 equal to the destination MAC address
- LISTELEM\_PTR equal to the pointer of the created instance of type ListElemStructType in the following order:
  - create an instance of type ListElemStructType and set NextListElemPtr to NULL\_PTR
  - set DataPtr to address of the created header and DataLength to the length of the created header
- PRIORITY\_8 equal to the value of [TcpIpIpFramePrioDefault](#) or to the value if changed by [TcpIp\\_ChangeParameter\(\)](#)

]

**[SWS\_Tcplp\_00404] Meta data handling while not containing headers***Status:* DRAFT*Upstream requirements:* [SRS\\_Eth\\_00187](#)

[If the Tcplp module provides headers necessary for the transmission through PduInfoPtr.SduDataPtr then MetaDataItem shall be set in the following order:

- ETHERNET\_MAC\_64 equal to the Physical destination address (MAC address in network byte order)
- LISTELEM\_PTR equal to NULL\_PTR

- `PRIORITY_8` equal to the value of `TcpIpIpFramePrioDefault` or to the value if changed by `TcpIp_ChangeParameter()`

]

Note: The `TcpIp_TriggerTransmit()` should be called for each `TcpIpCtrlTxPdu` where the Tcplp module successfully requested a transmission with indirect data provision.

#### **[SWS\_Tcplp\_00405] Data provision for transmission request with indirect data provision**

*Status:* DRAFT

*Upstream requirements:* [SRS\\_Eth\\_00187](#)

[If the `TcpIp_TriggerTransmit()` is called and the Tcplp module requested a transmission with indirect data provision for a specific `TcpIpCtrlTxPdu`, and given `TxPduId` is in state `PDU_IN_USE` then the Tcplp module shall provide the data (copy the payload and add the headers) to the given `PduInfoPtr.SduDataPtr` and update `PduInfoPtr.SduLength`.]

Note: The `TcpIp_TxConfirmation()` should be called for each `TcpIpCtrlTxPdu` where the Tcplp module successfully requested a transmission with either indirect or direct data provision.

## **7.8 State handling**

### **7.8.1 State handling of PDUs**

PDUs are used to transfer data across the layers in the AUTOSAR communication stack. Each `TcpIpCtrl` references a list of `TcpIpCtrlRxPdu` and `TcpIpCtrlTxPdu` to interchange data with the lower layer. The Tcplp module requests data transmission via PDUs and is indicated for data reception via PDUs.

#### **[SWS\_Tcplp\_00406] Transmission PDU states usage**

*Status:* DRAFT

*Upstream requirements:* [SRS\\_Eth\\_00187](#)

[The Tcplp module shall maintain a separate state for each transmission PDU used by the Tcplp module (`TcpIpCtrlTxPdu`) and distinguish at least the states in [\[SWS\\_Tcplp\\_00407\]](#).]

### [SWS\_Tcplp\_00407] Transmission PDU states description

*Status:* DRAFT

*Upstream requirements:* [SRS\\_Eth\\_00187](#)

[

PDU state	PDU state behavior
PDU_AVAILABLE	PDU for a specific transmission is available and ready to be used (PDU resources are released)
PDU_IN_USE	PDU for a specific transmission is not available and is already used (PDU resources are valid)

]

### [SWS\_Tcplp\_00408] Starting transmission request

*Status:* DRAFT

*Upstream requirements:* [SRS\\_Eth\\_00187](#)

[The Tcplp module shall request transmission only on PDU in state PDU\_AVAILABLE, allocate PDU resources and necessary buffer depending on the transmission type (see [\[SWS\\_Tcplp\\_00401\]](#), [\[SWS\\_Tcplp\\_00402\]](#), [\[SWS\\_Tcplp\\_00403\]](#), [\[SWS\\_Tcplp\\_00404\]](#)), enter the state PDU\_IN\_USE and call LSduR\_TcpIpTransmit().]

### [SWS\_Tcplp\_00409] Finishing transmission request

*Status:* DRAFT

*Upstream requirements:* [SRS\\_Eth\\_00187](#)

[If the transmission confirmation [TcpIp\\_TxConfirmation\(\)](#) is called on PDU in state PDU\_IN\_USE, the Tcplp module shall release all PDU resources and enter the state PDU\_AVAILABLE.]

### [SWS\_Tcplp\_00410] Aborting transmission request

*Status:* DRAFT

*Upstream requirements:* [SRS\\_Eth\\_00187](#)

[If the Tcplp module requested to transmit data and the LSduR\_TcpIpTransmit() returned E\_NOT\_OK, then the Tcplp module shall release all PDU resources and set the state of the affected PDU back to PDU\_AVAILABLE.]

## 7.8.2 TCP/IP Stack state handling

**[SWS\_Tcplp\_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\_Tcplp\_00136]** [The Tcplp module shall initiate according actions to achieve the requested state if the stored state request is not the active state.]

**[SWS\_Tcplp\_00084]** [After each transition the Tcplp module shall report the new state to EthSM via `EthSM_TcpIpModeIndication()`.]

**[SWS\_Tcplp\_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

- enable all IP address assignments according to the configured assignment methods (`TcpIpAssignmentMethod`) and triggers (`TcpIpAssignmentTrigger`) 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
- enter the state `TCPIP_STATE_STARTUP` for the EthIf controller.

]

**[SWS\_Tcplp\_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\_Tcplp\_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

- abort all ongoing IP address assignment actions appropriate and
- enter the state `TCPIP_STATE_OFFLINE` for the EthIf controller.

]

**[SWS\_Tcplp\_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_LocalIpAddrAssignmentChg>()` with State `TCPIP_IPADDR_STATE_ASSIGNED`.

**[SWS\_Tcplp\_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

- notify the upper layer via `<Up_LocalIpAddressAssignmentChg>()` with State TCPIP\_IPADDR\_STATE\_ONHOLD for all assigned IP addresses of the related EthIf controller, and
- deactivate the communication within the Tcplp module for the related EthIf controller, and
- enter the state TCPIP\_STATE\_ONHOLD for the EthIf controller.

]

**[SWS\_Tcplp\_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

- reactivate the communication within the Tcplp module for the related EthIf controller,
- call `<Up_LocalIpAddressAssignmentChg>()` with State TCPIP\_IPADDR\_STATE\_ASSIGNED for all assigned IP addresses of the related EthIf controller, and
- enter the state TCPIP\_STATE\_ONLINE for the EthIf controller.

]

**[SWS\_Tcplp\_00077]** **An offline request for an online state** [If TCPIP\_STATE\_OFFLINE is requested or all assigned IP addresses have been released for an EthIf controller and the current state is TCPIP\_STATE\_ONLINE for that EthIf controller, the Tcplp module shall

- call `<Up_LocalIpAddressAssignmentChg>()` with state TCPIP\_IPADDR\_STATE\_UNASSIGNED for all assigned IP addresses of the related EthIf controller,
- deactivate the communication (requests from and to the upper layer) within the Tcplp module for the related EthIf controller,
- in regards to [SWS\_Tcplp\_00072] request releasing related resources, i.e. any socket using the EthIf controller shall start closing procedure (TCP) or be closed (UDP) and after sockets are closed, any IP address assigned to the EthIf controller shall be unassigned,
- in case no EthIf controller is assigned any more, all unbound sockets shall be released as well, and
- enter the state TCPIP\_STATE\_SHUTDOWN for the EthIf controller.

]

**[SWS\_Tcplp\_00397] An offline request for an onhold state** [If TCPIP\_STATE\_OFFLINE is requested or all assigned IP addresses have been released for an Ethlf controller and the current state is TCPIP\_STATE\_ONHOLD for that Ethlf controller, the Tcplp module shall

- call `<Up_LocalIpAddrAssignmentChg>()` with state TCPIP\_IPADDR\_STATE\_UNASSIGNED for all assigned IP addresses of the related Ethlf controller,
- in regards to [SWS\_Tcplp\_00072] and [SWS\_Tcplp\_00074] request releasing related resources, i.e. any socket using the Ethlf controller shall be closed and thereafter any IP address assigned to the Ethlf controller shall be unassigned,
- in case no Ethlf controller is assigned any more, all unbound sockets shall be released as well, and
- enter the state TCPIP\_STATE\_SHUTDOWN for the Ethlf controller.

]

**[SWS\_Tcplp\_00398] TCP sockets immediate release in an onhold state** [If TCPIP\_STATE\_OFFLINE is requested and the current state of that Ethlf controller is TCPIP\_STATE\_ONHOLD, the Tcplp module shall in regards to [SWS\_Tcplp\_00074] immediately release TCP sockets without any transmission.]

**[SWS\_Tcplp\_00399] TCP sockets release with handshake in an online state** [If TCPIP\_STATE\_OFFLINE is requested, the current state of that Ethlf controller is TCPIP\_STATE\_ONLINE and TCP sockets are in one of the states SYN-RECEIVED, FIN-WAIT-1, FIN-WAIT-2, LAST-ACK, TIME-WAIT, CLOSING, CLOSE-WAIT or ESTABLISHED, the Tcplp module shall release TCP sockets with handshake as defined in IETF RFC 793.]

**[SWS\_Tcplp\_00400] TCP sockets immediate release in an online state** [If TCPIP\_STATE\_OFFLINE is requested, the current state of that Ethlf controller is TCPIP\_STATE\_ONLINE and TCP sockets are in the state SYN-SENT or LISTEN, the Tcplp module shall immediately release TCP sockets without handshake as defined in IETF RFC 793.]

**[SWS\_Tcplp\_00372]** [If TCPIP\_STATE\_ONLINE is requested and the current state of an Ethlf controller is TCPIP\_STATE\_SHUTDOWN, then Tcplp module shall

- immediately finish releasing all related resources stated in [SWS\_Tcplp\_00077],
- TCP connections shall be aborted and the ones that are still in one of the states (SYN-RECEIVED, CLOSE-WAIT, FIN-WAIT-1, FIN-WAIT-2) shall transmit a

RST-segment to inform a remote host as soon as possible that the connection was closed,

- enter the state `TCPIP_STATE_OFFLINE` for the EthIf controller without indication this state to the EthSM,
- after all resources have been released and state `TCPIP_STATE_OFFLINE` was entered, start assigning the resources according to the requirement [SWS\_Tcplp\_00075].

]

**[SWS\_Tcplp\_00087]** [If the current state of an EthIf controller is `TCPIP_STATE_SHUTDOWN` and all related resources have been released (i.e. socket closed which leads to unassignment of IP address assigned to that EthIf controller), the Tcplp module shall enter the state `TCPIP_STATE_OFFLINE` for the EthIf controller.]

**[SWS\_Tcplp\_00094]** [The Tcplp module shall only accept new TCP connections if the related EthIf controller is in state `TCPIP_STATE_ONLINE`.]

**[SWS\_Tcplp\_00144]** [The Tcplp module shall indicate events related to sockets to the upper layer module by using the `<Up_TcpIpEvent>` API and the following events: `TCPIP_TCP_RESET`, `TCPIP_TCP_CLOSED`, `TCPIP_TCP_FIN_RECEIVED` and `TCPIP_UDP_CLOSED`.]

## 7.9 Error Classification

Section "Error Handling" of the document [6] "General Specification of Basic Software Modules" describes the error handling of the Basic Software in detail. Above all, it constitutes a classification scheme consisting of five error types which may occur in BSW modules.

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

### 7.9.1 Development Errors

The following table lists development error IDs the Tcp/Ip shall use for reporting of development errors to the Default Error Tracer:



## [SWS\_TCPIP\_00042] Definition of development errors in module Tcplp [

Type of error	Related error code	Error value
Error code as specified by SWS_BSW_00243, if any API service (except Tcplp_GetVersionInfo and Tcplp_MainFunction) is called before the AUTOSAR Tcplp module was initialized with TcpIp_Init.	TCPIP_E_UNINIT	0x01
API service called with NULL pointer	TCPIP_E_PARAM_POINTER	0x02
Invalid argument	TCPIP_E_INV_ARG	0x03
No buffer space available	TCPIP_E_NOBUFS	0x04
Message too long	TCPIP_E_MSGSIZE	0x07
Protocol wrong type for socket	TCPIP_E_PROTOTYPE	0x08
Address already in use	TCPIP_E_ADDRINUSE	0x09
Can't assign requested address	TCPIP_E_ADDRNOTAVAIL	0x0A
Socket is already connected	TCPIP_E_ISCONN	0x0B
Socket is not connected	TCPIP_E_NOTCONN	0x0C
Protocol not available	TCPIP_E_NOPROTOPT	0x0D
Address family not supported by protocol family	TCPIP_E_AFNOSUPPORT	0x0E
Invalid configuration set selection	TCPIP_E_INIT_FAILED	0x0F
TriggerTransmit was called for an unexpected PDU	TCPIP_E_PDU_DATA_FAILED	0x10
An API was called with an unknown PDU ID	TCPIP_E_INVALID_PDU_SDU_ID	0x11

]

## 7.9.2 Runtime Errors

The following table lists runtime error IDs the Tcp/Ip shall use for reporting of runtime errors to the Default Error Tracer:

## [SWS\_TCPIP\_00255] Definition of runtime errors in module Tcplp

Upstream requirements: [SRS\\_Eth\\_00112](#)

[

Type of error	Related error code	Error value
Operation timed out	TCPIP_E_TIMEDOUT	0x01
Connection refused	TCPIP_E_CONNREFUSED	0x02
No route to host	TCPIP_E_HOSTUNREACH	0x03
Path does not support frame size	TCPIP_E_PACKETTOBIG	0x04
Duplicate IP Address detected	TCPIP_E_DADCONFLICT	0x05
A message could not be stored in the transmission request queue	TCPIP_E_TX_QUEUE_OVERRUN	0x06
Internal transmission processing aborted	TCPIP_E_TX_INTERNAL_PROCESSING_FAILED	0x07

]

**[SWS\_Tcplp\_00256]**

*Upstream requirements:* [SRS\\_Eth\\_00112](#)

[The Tcplp shall report the runtime error by calling `Det_ReportRuntimeError(TCPIP_E_TIMEDOUT)` if one of the following conditions applies:

- Tcplp module has sent a SYN to establish a connection but did not receive any response.
- 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.
- 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 [\[SWS\\_Tcplp\\_00202\]](#).

]

**[SWS\_Tcplp\_00257]**

*Upstream requirements:* [SRS\\_Eth\\_00112](#)

[The Tcplp shall report the runtime error by calling `Det_ReportRuntimeError(TCPIP_E_CONNREFUSED)` if one of the following conditions applies:

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

]

**[SWS\_Tcplp\_00258]**

*Upstream requirements:* [SRS\\_Eth\\_00112](#)

[The Tcplp shall report the runtime error by calling `Det_ReportRuntimeError(TCPIP_E_HOSTUNREACH)` if one of the following conditions applies:

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

]

**[SWS\_Tcplp\_00259]**

*Upstream requirements:* [SRS\\_Eth\\_00112](#)

[The Tcplp shall report the runtime error by calling `Det_ReportRuntimeError(TCPIP_E_PACKETTOBIG)` if one of the following conditions applies:

- 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.

]

#### [SWS\_Tcplp\_00282]

*Upstream requirements:* [SRS\\_Eth\\_00091](#), [SRS\\_BSW\\_00452](#)

[The Tcplp shall report the runtime error by calling `Det_ReportRuntimeError(TCPIP_E_DADCONFLICT)` if one of the following conditions applies:

- A duplicate IP address was found by the Duplicate Address Detection (DAD) algorithm.

]

#### [SWS\_Tcplp\_00413] Error report for transmission request queue

*Status:* DRAFT

*Upstream requirements:* [SRS\\_Eth\\_00187](#)

[The Tcplp shall report the runtime error by calling `Det_ReportRuntimeError(TCPIP_E_TX_QUEUE_OVERRUN)` if one of the following conditions applies:

- A transmission processing is aborted because the message could not be stored in the transmission request queue.

]

#### [SWS\_Tcplp\_00414] Error report for aborting the transmission request

*Status:* DRAFT

*Upstream requirements:* [SRS\\_Eth\\_00187](#)

[The Tcplp shall report the runtime error by calling `Det_ReportRuntimeError(TCPIP_E_TX_INTERNAL_PROCESSING_FAILED)` if one of the following conditions applies:

- A transmission processing is requested from `LSduR_TcpIpTransmit()` and then aborted.

]

### 7.9.3 Production Errors

There are no production errors.

## 7.9.4 Extended Production Errors

There are no extended production errors.

## 7.10 Version checking

For details refer to the chapter 5.1.8 "Version Check" in SWS\_BSWGeneral.

## 7.11 Security Events

### [SWS\_Tcplp\_00361]

*Upstream requirements:* [RS\\_Ids\\_00810](#)

[If security event reporting has been enabled for the Tcplp module ([TcpIpEnableSecurityEventReporting](#) = true) the respective security events shall be reported to the IdsM via the interfaces defined in AUTOSAR\_SWS\_BSWGeneral.]

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

### [SWS\_Tcplp\_00362] Security events for Tcplp

*Status:* DRAFT

*Upstream requirements:* [RS\\_Ids\\_00810](#)

[

Name	Description	ID
SEV_ARP_IP_ADDR_CONFLICT	Received local IP address in ARP reply for different MAC.	10
SEV_TCP_DROP_INV_PORT	Dropped TCP packet because of invalid destination TCP-Port.	11
SEV_UDP_DROP_INV_PORT	Dropped UDP packet because of invalid destination UDP-Port.	12
SEV_IPV4_DROP_INV_ADDR	Dropped datagram because of invalid IPV4 address.	13
SEV_IPV6_DROP_INV_ADDR	Dropped datagram because of invalid IPV6 address.	14
SEV_TLS_ERROR	An alert message (warning or fatal) was detected (either received or generated) by TLS.	90
SEV_TLS_CONNECTION_ESTABLISHED	A TLS connection was successfully established.	91
SEV_TLS_CONNECTION_CLOSED	A TLS connection was closed normally.	92

]

**[SWS\_Tcplp\_00382] ARP conflict IdsM reporting***Status:* DRAFT*Upstream requirements:* [RS\\_Ids\\_00810](#)

[Upon reception of a local IP address in ARP reply for a different MAC, Tcplp shall raise SEV\_ARP\_IP\_ADDR\_CONFLICT to the IdsM.]

**[SWS\_Tcplp\_00383] TCP invalid port IdsM reporting***Status:* DRAFT*Upstream requirements:* [RS\\_Ids\\_00810](#)

[Upon dropping of a TCP packet due to invalid destination TCP-Port, Tcplp shall raise SEV\_TCP\_DROP\_INV\_PORT to the IdsM.]

**[SWS\_Tcplp\_00384] UDP invalid port IdsM reporting***Status:* DRAFT*Upstream requirements:* [RS\\_Ids\\_00810](#)

[Upon dropping of a UDP packet due to invalid destination UDP-Port, Tcplp shall raise SEV\_UDP\_DROP\_INV\_PORT to the IdsM.]

**[SWS\_Tcplp\_00385] IPv4 invalid address IdsM reporting***Status:* DRAFT*Upstream requirements:* [RS\\_Ids\\_00810](#)

[Upon dropping a datagram due to invalid IPv4 address, Tcplp shall raise SEV\_IPV4\_DROP\_INV\_ADDR to the IdsM.]

**[SWS\_Tcplp\_00386] IPv6 invalid address IdsM reporting***Status:* DRAFT*Upstream requirements:* [RS\\_Ids\\_00810](#)

[Upon dropping a datagram due to invalid IPv6 address, Tcplp shall raise SEV\_IPV6\_DROP\_INV\_ADDR to the IdsM.]

**[SWS\_Tcplp\_00387] TLS error IdsM reporting***Status:* DRAFT*Upstream requirements:* [RS\\_Ids\\_00810](#)

[Upon an alert message being detected (either received or generated) by TLS, Tcplp shall raise SEV\_TLS\_ERROR to the IdsM.]

### [SWS\_Tcplp\_00394] Security event context data definition: SEV\_TLS\_ERROR

Status: DRAFT

Upstream requirements: [RS\\_Ids\\_00810](#)

[

SEV Name	SEV_TLS_ERROR	
ID	90	
Description	An alert message (warning or fatal) was detected (either received or generated) by TLS.	
Context Data Version	1	
Context Data	Data Type	Allowed Values
ReasonForFailure	uint8	Alert message as described in the the Alert Protocol in - RFC5246 for TLS Version 1.2 - RFC8446 for TLS Version 1.3
TLSVersion	uint16	Version as defined in RFC5246, RFC8446 - 0x0303 for TLS Version 1.2 - 0x0304 for TLS Version 1.3
SourceIpAddress	uint8 [16]	All IPv6 addresses and IPv4 addresses shall be encoded as specified in RFC 4291 Section 2.5.5.2
SourcePort	uint16	
DestinationIpAddress	uint8 [16]	All IPv6 addresses and IPv4 addresses shall be encoded as specified in RFC 4291 Section 2.5.5.2
DestinationPort	uint16	

]

### [SWS\_Tcplp\_00388] TLS established connection IdsM reporting

Status: DRAFT

Upstream requirements: [RS\\_Ids\\_00810](#)

[Upon a TLS connection being successfully established, Tcplp shall raise SEV\_TLS\_CONNECTION\_ESTABLISHED to the IdsM.]

### [SWS\_Tcplp\_00395] Security event context data definition: SEV\_TLS\_CONNECTION\_ESTABLISHED

Status: DRAFT

Upstream requirements: [RS\\_Ids\\_00810](#)

[

SEV Name	SEV_TLS_CONNECTION_ESTABLISHED	
ID	91	
Description	A TLS connection was successfully established.	
Context Data Version	1	
Context Data	Data Type	Allowed Values
TLSVersion	uint16	Version as defined in RFC5246, RFC8446 - 0x0303 for TLS Version 1.2 - 0x0304 for TLS Version 1.3
SourceIpAddress	uint8 [16]	All IPv6 addresses and IPv4 addresses shall be encoded as specified in RFC 4291 Section 2.5.5.2
SourcePort	uint16	

▽



SEV Name	SEV_TLS_CONNECTION_ESTABLISHED	
DestinationIpAddress	uint8 [16]	All IPv6 addresses and IPv4 addresses shall be encoded as specified in RFC 4291 Section 2.5.5.2
DestinationPort	uint16	

]

### [SWS\_Tcplp\_00389] TLS closed connection IdsM reporting

Status: DRAFT

Upstream requirements: [RS\\_Ids\\_00810](#)

[Upon a TLS connection being closed normally (via `close_notify(0)` or `user_canceled(90)`), Tcplp shall raise `SEV_TLS_CONNECTION_CLOSED` to the IdsM.]

### [SWS\_Tcplp\_00396] Security event context data definition: SEV\_TLS\_CONNECTION\_CLOSED

Status: DRAFT

Upstream requirements: [RS\\_Ids\\_00810](#)

[

SEV Name	SEV_TLS_CONNECTION_CLOSED	
ID	92	
Description	A TLS connection was closed normally.	
Context Data Version	1	
Context Data	Data Type	Allowed Values
ReasonForClosure	uint8	<code>close_notify(0)</code> <code>user_canceled(90)</code>
TLSVersion	uint16	Version as defined in RFC5246, RFC8446 - 0x0303 for TLS Version 1.2 - 0x0304 for TLS Version 1.3
SourceIpAddress	uint8 [16]	All IPv6 addresses and IPv4 addresses shall be encoded as specified in RFC 4291 Section 2.5.5.2
SourcePort	uint16	
DestinationIpAddress	uint8 [16]	All IPv6 addresses and IPv4 addresses shall be encoded as specified in RFC 4291 Section 2.5.5.2
DestinationPort	uint16	

]

## 8 API specification

### 8.1 Imported types

In this chapter all types included from the following files are listed.

#### [SWS\_TCPIP\_00008] Definition of imported datatypes of module Tcplp [

Module	Header File	Imported Type
Comtype	ComStack_Types.h	BufReq_ReturnType
	ComStack_Types.h	PduIdType
	ComStack_Types.h	PduInfoType
	ComStack_Types.h	PduLengthType
Csm	Rte_Csm_Type.h	Crypto_OperationModeType
	Rte_Csm_Type.h	Crypto_VerifyResultType
Dem	Rte_Dem_Type.h	Dem_EventIdType
	Rte_Dem_Type.h	Dem_EventStatusType
Eth	Eth_GeneralTypes.h	Eth_FilterActionType
IdsM	IdsM_Types.h	IdsM_SecurityEventIdType
KeyM	KeyM.h	KeyM_CertDataType
	Rte_KeyM_Type.h	KeyM_CertificateIdType
Std	Std_Types.h	Std_ReturnType
	Std_Types.h	Std_VersionInfoType

]

### 8.2 Type definitions

#### [SWS\_TCPIP\_00067] Definition of datatype Tcplp\_ConfigType [

<b>Name</b>	Tcplp_ConfigType	
<b>Kind</b>	Structure	
<b>Elements</b>	implementation specific	
	<b>Type</b>	–
	<b>Comment</b>	The content of the configuration data structure is implementation specific.
<b>Description</b>	Configuration data structure of the Tcplp module.	
<b>Available via</b>	Tcplp.h	

]



**[SWS\_TCPIP\_00009] Definition of datatype Tcplp\_DomainType** [

<b>Name</b>	Tcplp_DomainType		
<b>Kind</b>	Type		
<b>Derived from</b>	uint16		
<b>Range</b>	TCPIP_AF_INET	0x02	Use IPv4
	TCPIP_AF_INET6	0x1c	Use IPv6
<b>Description</b>	Tcplp address families.		
<b>Available via</b>	Tcplp.h		

]

**[SWS\_TCPIP\_00010] Definition of datatype Tcplp\_ProtocolType** [

<b>Name</b>	Tcplp_ProtocolType		
<b>Kind</b>	Enumeration		
<b>Range</b>	TCPIP_IPPROTO_TCP	0x06	Use TCP
	TCPIP_IPPROTO_UDP	0x11	Use UDP
<b>Description</b>	Protocol type used by a socket.		
<b>Available via</b>	Tcplp.h		

]

**[SWS\_TCPIP\_00012] Definition of datatype Tcplp\_SockAddrType** [

<b>Name</b>	Tcplp_SockAddrType	
<b>Kind</b>	Structure	
<b>Elements</b>	domain	
	<b>Type</b>	<a href="#">Tcplp_DomainType</a>
	<b>Comment</b>	This is the code for the address format of this address
<b>Description</b>	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.)	
<b>Available via</b>	Tcplp.h	

]

**[SWS\_TCPIP\_00013] Definition of datatype Tcplp\_SockAddrInetType** [

<b>Name</b>	Tcplp_SockAddrInetType	
<b>Kind</b>	Structure	
<b>Elements</b>	domain	
	<b>Type</b>	Tcplp_DomainType
	<b>Comment</b>	This is the code for the address format of this address
	port	
	<b>Type</b>	uint16
	<b>Comment</b>	port number
	addr	





	<b>Type</b>	Array of uint32
	<b>Size</b>	1
	<b>Comment</b>	IPv4 address in network byte order
<b>Description</b>	This structure defines an IPv4 address type which can be derived from the generic address structure via cast.	
<b>Available via</b>	Tcplp.h	

]

#### [SWS\_TCPIP\_00014] Definition of datatype Tcplp\_SockAddrInet6Type [

<b>Name</b>	Tcplp_SockAddrInet6Type	
<b>Kind</b>	Structure	
<b>Elements</b>	domain	
	<b>Type</b>	<a href="#">Tcplp_DomainType</a>
	<b>Comment</b>	This is the code for the address format of this address
	port	
	<b>Type</b>	uint16
	<b>Comment</b>	port number
	addr	
	<b>Type</b>	Array of uint32
	<b>Size</b>	4
	<b>Comment</b>	IPv6 address in network byte order
<b>Description</b>	This structure defines a IPv6 address type which can be derived from the generic address structure via cast.	
<b>Available via</b>	Tcplp.h	

]

#### [SWS\_TCPIP\_00030] Definition of datatype Tcplp\_LocalAddrIdType [

<b>Name</b>	Tcplp_LocalAddrIdType	
<b>Kind</b>	Type	
<b>Derived from</b>	uint8	
<b>Description</b>	Address identification type for unique identification of a local IP address and EthIf Controller configured in the Tcplp module.	
<b>Available via</b>	Tcplp.h	

]

#### [SWS\_TCPIP\_00038] Definition of datatype Tcplp\_SocketIdType [

<b>Name</b>	Tcplp_SocketIdType	
<b>Kind</b>	Type	
<b>Derived from</b>	<b>Basetype</b>	<b>Variation</b>
	uint16	–





	uint8	–
<b>Description</b>	Socket identifier type for unique identification of a Tcplp stack socket. TCPIP_SOCKETID_INVALID shall specify an invalid socket handle.	
<b>Available via</b>	Tcplp.h	

]

### [SWS\_TCPIP\_00073] Definition of datatype Tcplp\_StateType [

<b>Name</b>	Tcplp_StateType		
<b>Kind</b>	Enumeration		
<b>Range</b>	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 EthIf controller is ONHOLD, i.e. no communication is currently possible (e.g. link down).
	TCPIP_STATE_OFFLINE	–	TCP/IP stack state for a specific EthIf controller is OFFLINE, i.e. no communication is possible.
	TCPIP_STATE_STARTUP	–	TCP/IP stack state for a specific EthIf 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 EthIf controller is SHUTDOWN, i.e. release of resources using the EthIf controller, release of IP address assignment.
<b>Description</b>	Specifies the Tcplp state for a specific EthIf controller.		
<b>Available via</b>	Tcplp.h		

]

### [SWS\_TCPIP\_00082] Definition of datatype Tcplp\_IpAddrStateType [

<b>Name</b>	Tcplp_IpAddrStateType		
<b>Kind</b>	Enumeration		
<b>Range</b>	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
<b>Description</b>	Specifies the state of local IP address assignment		
<b>Available via</b>	Tcplp.h		

]

### [SWS\_TCPIP\_00031] Definition of datatype Tcplp\_EventType [

<b>Name</b>	Tcplp_EventType		
<b>Kind</b>	Enumeration		
<b>Range</b>	TCPIP_TCP_RESET	0x01	TCP connection was reset, TCP socket and all related resources have been released.
	TCPIP_TCP_CLOSED	0x02	TCP connection was closed successfully, TCP socket and all related resources have been released.
	TCPIP_TCP_FIN_RECEIVED	0x03	A FIN signal was received on the TCP connection, TCP socket is still valid.
	TCPIP_UDP_CLOSED	0x04	UDP socket and all related resources have been released.
	TCPIP_TLS_HANDSHAKE_SUCCEEDED	0x05	TLS handshake successfully established, TLS connection available.
<b>Description</b>	Events reported by Tcplp.		
<b>Available via</b>	Tcplp.h		

]

### [SWS\_TCPIP\_00065] Definition of datatype Tcplp\_IpAddrAssignmentType [

<b>Name</b>	Tcplp_IpAddrAssignmentType		
<b>Kind</b>	Enumeration		
<b>Range</b>	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.
	TCPIP_IPADDR_ASSIGNMENT_ALL	–	All configured TcplpAssignmentMethods with TcplpAssignmentTrigger set to TCPIP_MANUAL
<b>Description</b>	Specification of IPv4/IPv6 address assignment policy.		
<b>Available via</b>	Tcplp.h		

]

### [SWS\_TCPIP\_00066] Definition of datatype Tcplp\_ReturnType [

<b>Name</b>	Tcplp_ReturnType		
<b>Kind</b>	Enumeration		
<b>Range</b>	TCPIP_E_OK	–	operation completed successfully.
	TCPIP_E_NOT_OK	–	operation failed.





	TCPIP_E_PHYS_ADDR_MISS	–	operation failed because of an ARP/NDP cache miss.
	TCPIP_E_PENDING	–	operation in progress
<b>Description</b>	Tcplp specific return type.		
<b>Available via</b>	Tcplp.h		

## [SWS\_TCPIP\_00126] Definition of datatype Tcplp\_ParamIdType [

<b>Name</b>	Tcplp_ParamIdType		
<b>Kind</b>	Type		
<b>Derived from</b>	uint8		
<b>Range</b>	TCPIP_PARAMID_TCP_RXWND_MAX	0x00	Specifies the maximum TCP receive window for the socket. [uint16]
	TCPIP_PARAMID_FRAMEPRIO	0x01	Specifies the frame priority for outgoing frames on the socket. [uint8]
	TCPIP_PARAMID_TCP_NAGLE	0x02	Specifies if the Nagle Algorithm according to IETF RFC 1122 (chapter 4.2.3.4 When to Send Data) is enabled or not. [boolean]
	TCPIP_PARAMID_TCP_KEEPALIVE	0x03	Specifies if TCP Keep Alive Probes are sent on the socket connection. [boolean]
	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. [uint8]
	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. [uint32]
	TCPIP_PARAMID_TCP_KEEPALIVE_PROBES_MAX	0x06	Specifies the maximum number of times that a keepalive probe is retransmitted. [uint16]
	TCPIP_PARAMID_TCP_KEEPALIVE_INTERVAL	0x07	Specifies the interval in [s] between subsequent keepalive probes. [uint32]
	TCPIP_PARAMID_TCP_OPTIONFILTER	0x08	Specifies which TCP option filter shall be applied on the related socket. [uint8]
	TCPIP_PARAMID_PATHMTU_ENABLE	0x09	Specifies if the Path MTU Discovery shall be performed on the related socket. [boolean]
	TCPIP_PARAMID_FLOWLABEL	0x0a	The 20-bit Flow Label according to IETF RFC 6437. [uint32]
	TCPIP_PARAMID_DSCP	0x0b	The 6-bit Differentiated Service Code Point according to IETF RFC 2474. [uint8]





	TCPIP_PARAMID_UDP_CHECKSUM	0x0c	0x0c Specifies if UDP checksum handling shall be enabled (TRUE) or skipped (FALSE) on the related socket. [boolean]
	TCPIP_PARAMID_TLS_CONNECTION_ASSIGNMENT	0x0d	0x0d is used to assign a TLS connection reference to a TCP socket.
	TCPIP_PARAMID_TCP_RETRANSMIT_TIMEOUT	0x0e	Initial TCP Retransmission timeout before an unacknowledged segment is retransmitted (overrides TcplpTcpRetransmissionTimeout (ECUC_Tcplp_00068))
	TCPIP_PARAMID_TCP_MAXRTX	0x0f	Specifies the maximum number of TCP retransmissions
	TCPIP_PARAMID_TCP_MAX_RETRANSMIT_TIMEOUT	0x10	Maximal TCP Retransmission timeout before an unacknowledged segment is retransmitted (overrides TcplpTcpMaxRetransmissionTimeout (ECUC_Tcplp_00340))
	TCPIP_PARAMID_VENDOR_SPECIFIC	0x80	Start of vendor specific range of parameter IDs. [vendor specific]
<b>Description</b>		Type for the specification of all supported Parameter IDs and their data types.	
<b>Available via</b>		Tcplp.h	

### [SWS\_TCPIP\_91004] Definition of datatype Tcplp\_ArpCacheEntryType [

<b>Name</b>	Tcplp_ArpCacheEntryType		
<b>Kind</b>	Structure		
<b>Elements</b>	InetAddr		
	<b>Type</b>	Array of uint32	
	<b>Size</b>	1	
	<b>Comment</b>	IPv4 address in network byte order	
	PhysAddr		
	<b>Type</b>	Array of uint8	
	<b>Size</b>	6	
	<b>Comment</b>	physical address in network byte order	
	State		
	<b>Type</b>	uint8	
	<b>Comment</b>	state of the address entry (TCPIP_ARP_ENTRY_STATIC, TCPIP_ARP_ENTRY_VALID, TCPIP_ARP_ENTRY_STALE)	
<b>Description</b>	Tcplp_ArpCacheEntries elements type		
<b>Available via</b>	Tcplp.h		

### [SWS\_TCPIP\_91003] Definition of datatype Tcplp\_NdpCacheEntryType [

<b>Name</b>	Tcplp_NdpCacheEntryType		
<b>Kind</b>	Structure		
<b>Elements</b>	Inet6Addr		
	<b>Type</b>	Array of uint32	
	<b>Size</b>	4	
	<b>Comment</b>	IPv6 address in network byte order	
	PhysAddr		
	<b>Type</b>	Array of uint8	
	<b>Size</b>	6	
	<b>Comment</b>	physical address in network byte order	
	State		
	<b>Type</b>	uint8	
	<b>Comment</b>	state of the address entry (TCPIP_NDP_ENTRY_STATIC, TCPIP_NDP_ENTRY_VALID, TCPIP_NDP_ENTRY_STALE)	
<b>Description</b>	Tcplp_NdpCacheEntries elements type		
<b>Available via</b>	Tcplp.h		

### [SWS\_TCPIP\_91010] Definition of datatype Tcplp\_MeasurementIdxType [

<b>Name</b>	Tcplp_MeasurementIdxType		
<b>Kind</b>	Type		
<b>Derived from</b>	uint8		
<b>Range</b>	TCPIP_MEAS_DROP_TCP	0x01	Measurement index of dropped PDUs caused by invalid destination TCP-Port
	TCPIP_MEAS_DROP_UDP	0x02	Measurement index of dropped PDUs caused by invalid destination UDP-Port
	TCPIP_MEAS_DROP_IPV4	0x03	Measurement index of dropped datagrams caused by invalid IPv4 address
	TCPIP_MEAS_DROP_IPV6	0x04	Measurement index of dropped datagrams caused by invalid IPv6 address
	TCPIP_MEAS_RESERVED_1	0x05-0x7F	reserved by AUTOSAR
	TCPIP_MEAS_RESERVED_2	0x80-0xEF	Vendor specific range
	TCPIP_MEAS_RESERVED_3	0xF0-0xFE	reserved by AUTOSAR (future use)
	TCPIP_MEAS_ALL	0xFF	represents all measurement indexes
<b>Description</b>	Index to select specific measurement data		
<b>Available via</b>	Tcplp.h		

### [SWS\_TCPIP\_91011] Definition of datatype Tcplp\_TlsConnectionIdType [

<b>Name</b>	Tcplp_TlsConnectionIdType	
<b>Kind</b>	Type	
<b>Derived from</b>	<b>Basetype</b>	<b>Variation</b>
	uint16	–
	uint8	–
<b>Description</b>	TLS connection identifier type for unique identification of a TLS connection. TCPIP_TLSCONNECTIONID_INVALID shall specify an invalid TLS connection handle.	
<b>Available via</b>	Tcplp.h	

]

## 8.3 Symbol definitions

### [SWS\_TCPIP\_00133] Definition of symbol TCPIP\_IPADDR\_ANY [

<b>Name</b>	TCPIP_IPADDR_ANY
<b>Kind</b>	Symbol
<b>Base Type</b>	uint32
<b>Value</b>	implementation specific, defines the value used as wildcard
<b>Description</b>	IP address wildcard.
<b>Available via</b>	Tcplp.

]

### [SWS\_TCPIP\_00132] Definition of symbol TCPIP\_IP6ADDR\_ANY [

<b>Name</b>	TCPIP_IP6ADDR_ANY
<b>Kind</b>	Symbol
<b>Base Type</b>	uint32
<b>Value</b>	implementation specific, defines the value used as wildcard for all IP6 address parts
<b>Description</b>	IP6 address wildcard.
<b>Available via</b>	Tcplp.h

]

### [SWS\_TCPIP\_00134] Definition of symbol TCPIP\_PORT\_ANY [

<b>Name</b>	TCPIP_PORT_ANY
<b>Kind</b>	Symbol
<b>Base Type</b>	uint16
<b>Value</b>	Zero (0) is used as wildcard







<b>Description</b>	Port wildcard.
<b>Available via</b>	Tcplp.h

]

**[SWS\_TCPIP\_00135] Definition of symbol TCPIP\_LOCALADDRID\_ANY [**

<b>Name</b>	TCPIP_LOCALADDRID_ANY
<b>Kind</b>	Symbol
<b>Base Type</b>	<a href="#">Tcplp_LocalAddrIdType</a>
<b>Value</b>	implementation specific, defines the value used as wildcard
<b>Description</b>	LocalAddrId wildcard.
<b>Available via</b>	Tcplp.h

]

## 8.4 Function definitions

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

### 8.4.1 General

#### 8.4.1.1 Tcplp\_Init

**[SWS\_TCPIP\_00002] Definition of API function Tcplp\_Init [**

<b>Service Name</b>	Tcplp_Init	
<b>Syntax</b>	<pre>void TcpIp_Init (     const TcpIp_ConfigType* ConfigPtr )</pre>	
<b>Service ID [hex]</b>	0x01	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	ConfigPtr	Pointer to the configuration data of the Tcplp module
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	void	None
<b>Description</b>	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.	
<b>Available via</b>	Tcplp.h	

]

### 8.4.1.2 Tcplp\_GetVersionInfo

#### [SWS\_TCPIP\_00004] Definition of API function Tcplp\_GetVersionInfo [

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

]

[SWS\_Tcplp\_00005] [The function [TcpIp\\_GetVersionInfo\(\)](#) shall return the version information of this module. The version information includes:

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

]

[SWS\_Tcplp\_00006] [The function [TcpIp\\_GetVersionInfo\(\)](#) shall be pre compile time configurable On/Off by the configuration parameter: TCPIP\_VERSION\_INFO\_API]

## 8.4.2 Core Communication Control

### 8.4.2.1 Tcplp\_Close

#### [SWS\_TCPIP\_00017] Definition of API function Tcplp\_Close [

<b>Service Name</b>	Tcplp_Close	
<b>Syntax</b>	<pre>Std_ReturnType Tcplp_Close (     TcpIp_SocketIdType SocketId,     boolean Abort )</pre>	
<b>Service ID [hex]</b>	0x04	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant for different SocketIds. Non reentrant for the same SocketId.	
<b>Parameters (in)</b>	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.
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	Std_ReturnType	E_OK: The request has been accepted E_NOT_OK: The request has not been accepted.
<b>Description</b>	By this API service the TCP/IP stack is requested to close the socket and release all related resources.	
<b>Available via</b>	Tcplp.h	

[SWS\_Tcplp\_00109] [The service `TcpIp_Close()` shall perform the following actions for the socket specified by `SocketId` in case it is a TCP socket:

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

[SWS\_Tcplp\_00110] [The service `TcpIp_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_TcpIpEvent>`(TCPIP\_TCP\_CLOSED, TCPIP\_TCP\_RESET or TCPIP\_UDP\_CLOSED) after the socket and all related re-

sources have been released. After this call the `SocketId` is invalid until allocated again with `TcpIp_GetSocket()`.

#### 8.4.2.2 Tcplp\_Bind

##### [SWS\_TCPIP\_00015] Definition of API function Tcplp\_Bind [

<b>Service Name</b>	Tcplp_Bind	
<b>Syntax</b>	<pre>Std_ReturnType TcpIp_Bind (     TcpIp_SocketIdType SocketId,     TcpIp_LocalAddrIdType LocalAddrId,     uint16* PortPtr )</pre>	
<b>Service ID [hex]</b>	0x05	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different SocketIds. Non reentrant for the same SocketId.	
<b>Parameters (in)</b>	SocketId	Socket identifier of the related local socket resource.
	LocalAddrId	<p>IP address identifier representing the local IP address and EthIf controller to bind the socket to.</p> <p>Note: to listen to all EthIf controller, TCPIP_LOCALADDRID_ANY has to be specified as LocalAddrId.</p> <p>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.</p> <p>In case the socket shall be used as client socket, the IP address and EthIf controller represented by LocalAddrId is used for transmission.</p> <p>Note: for an automatic selection of the Local IP address and EthIf Controller, TCPIP_LOCALADDRID_ANY has to be specified as LocalAddrId.</p>
<b>Parameters (inout)</b>	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.
<b>Parameters (out)</b>	None	
<b>Return value</b>	Std_ReturnType	<p>Result of operation</p> <p>E_OK The request has been accepted</p> <p>E_NOT_OK The request has not been accepted (e.g. address in use)</p>
<b>Description</b>	By this API service the TCP/IP stack is requested to bind a UDP or TCP socket to a local resource.	
<b>Available via</b>	Tcplp.h	

]

[SWS\_Tcplp\_00111] [The service `TcpIp_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 [TcpIp\\_TcpConnect\(\)](#) or [TcpIp\\_UdpTransmit\(\)](#).

**[SWS\_Tcplp\_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\_Tcplp\_00147]**

*Upstream requirements:* [SRS\\_BSW\\_00323](#)

[If development error detection is enabled: [TcpIp\\_Bind\(\)](#) shall check if the parameter [LocalAddrId](#) is valid. If the check fails, [TcpIp\\_Bind\(\)](#) shall refuse the request and raise the development error code `TCPIP_E_ADDRNOTAVAIL` instead.]

#### **[SWS\_Tcplp\_00254]**

*Upstream requirements:* [SRS\\_Eth\\_00045](#)

[[TcpIp\\_Bind\(\)](#) shall check if the local address specified by [LocalAddrId](#) is assigned and if that is not the case refuse the request and return `E_NOT_OK`]

### **8.4.2.3 Tcplp\_TcpConnect**

#### **[SWS\_TCPIP\_00022] Definition of API function Tcplp\_TcpConnect [**

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

]

**[SWS\_Tcplp\_00112]** [The service `TcpIp_TcpConnect()` shall establish a TCP connection between the local socket specified by parameter `SocketId` and the remote socket specified with parameter `RemoteAddrPtr`.]

**[SWS\_Tcplp\_00129]** [If development error detection is enabled and the parameter `RemoteAddrPtr` equals `NULL_PTR`, the `TcpIp_TcpConnect()` function shall raise the development error code `TCPIP_E_PARAM_POINTER`.]

#### 8.4.2.4 Tcplp\_TcpListen

**[SWS\_TCPIP\_00023] Definition of API function Tcplp\_TcpListen** [

<b>Service Name</b>	Tcplp_TcpListen	
<b>Syntax</b>	<pre>Std_ReturnType TcpIp_TcpListen (     TcpIp_SocketIdType SocketId,     uint16 MaxChannels )</pre>	
<b>Service ID [hex]</b>	0x07	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Reentrant for different SocketIds. Non reentrant for the same SocketId.	
<b>Parameters (in)</b>	SocketId	Socket identifier of the related local socket resource.
	MaxChannels	Maximum number of new parallel connections established on this listen connection.
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	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.
<b>Description</b>	By this API service the TCP/IP stack is requested to listen on the TCP socket specified by the socket identifier.	
<b>Available via</b>	Tcplp.h	

]

**[SWS\_Tcplp\_00113]** [The service `TcpIp_TcpListen()` shall put the socket specified by `SocketId` to the listen state (i.e. local socket is listening for incoming connections).]

**[SWS\_Tcplp\_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`.]

## 8.4.2.5 Tcplp\_TcpReceived

## [SWS\_TCPIP\_00024] Definition of API function Tcplp\_TcpReceived [

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

]

[SWS\_Tcplp\_00115] [The service `TcpIp_TcpReceived()` shall increase the TCP receive window of the socket specified by `SocketId` considering the number of finally consumed bytes specified by `Length`.]

## 8.4.2.6 Tcplp\_RequestComMode

## [SWS\_TCPIP\_00070] Definition of API function Tcplp\_RequestComMode [

<b>Service Name</b>	Tcplp_RequestComMode	
<b>Syntax</b>	<pre>Std_ReturnType TcpIp_RequestComMode (     uint8 CtrlIdx,     TcpIp_StateType State )</pre>	
<b>Service ID [hex]</b>	0x09	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CtrlIdx	EthIf controller index to identify the communication network where the Tcplp state is requested.
	State	Requested Tcplp state.
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	





<b>Return value</b>	Std_ReturnType	E_OK: Service accepted E_NOT_OK: Service denied
<b>Description</b>	By this API service the TCP/IP stack is requested to change the Tcplp state of the communication network identified by EthIf controller index.	
<b>Available via</b>	Tcplp.h	

]

**[SWS\_Tcplp\_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\_Tcplp\_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\_Tcplp\_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\_Tcplp\_00089]** [If TCPIP\_STATE\_STARTUP or TCPIP\_STATE\_SHUTDOWN is requested as state the function [TcpIp\\_RequestComMode\(\)](#) shall abort with E\_NOT\_OK and report TCPIP\_E\_INV\_ARG if development error detection is enabled.]

Note: According to [\[SWS\\_Tcplp\\_00075\]](#) and [\[SWS\\_Tcplp\\_00077\]](#) 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.



### 8.4.3 Extended Communication Control and Information

#### 8.4.3.1 Tcplp\_RequestIpAddrAssignment

##### [SWS\_TCPIP\_00037] Definition of API function Tcplp\_RequestIpAddrAssignment

<b>Service Name</b>	Tcplp_RequestIpAddrAssignment	
<b>Syntax</b>	<pre>Std_ReturnType Tcplp_RequestIpAddrAssignment (     TcpIp_LocalAddrIdType LocalAddrId,     TcpIp_IpAddrAssignmentType Type,     const TcpIp_SockAddrType* LocalIpAddrPtr,     uint8 Netmask,     const TcpIp_SockAddrType* DefaultRouterPtr )</pre>	
<b>Service ID [hex]</b>	0x0A	
<b>Sync/Async</b>	Asynchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	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.
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	Std_ReturnType	E_OK: The request has been accepted E_NOT_OK: The request has not been accepted
<b>Description</b>	By this API service the local IP address assignment for the IP address specified by LocalAddrId shall be initiated.	
<b>Available via</b>	Tcplp.h	

[SWS\_Tcplp\_00116] [The service `Tcplp_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\_Tcplp\_00079] [In case `Tcplp_RequestIpAddrAssignment()` is called with parameter `Type` set to `TCPIP_IPADDR_ASSIGNMENT_STATIC` and no `TcpIp-StaticIpAddressConfig` container is configured for the `TcpIpLocalAddr` specified by parameter `LocalAddrId`, Tcplp shall assign the IP address, netmask and default router specified by parameter `LocalIpAddrPtr`, `Netmask` and `Default-`

`RouterPtr` as soon as `TCPIP_STATE_ONLINE` is requested or immediately if already requested.]

**[SWS\_Tcplp\_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\_Tcplp\_00299]** [In case `TcpIp_RequestIpAddrAssignment()` is called with parameter `Type` set to `TCPIP_IPADDR_ASSIGNMENT_ALL`, the IP address assignment for the IP address table entry specified by `LocalAddrId` shall be initiated for all configured `TcpIpAssignmentMethod` with `TcpIpAssignmentTrigger` set to `TCPIP_MANUAL`.]

**[SWS\_Tcplp\_00195]** [If `TcpIp_RequestIpAddrAssignment()` is called for a `LocalAddrId` configured with `TcpIpAssignmentTrigger` set to `TCPIP_MANUAL`, `Tcplp` shall consider the related assignment as available.]

**[SWS\_Tcplp\_00196]** [If `TcpIp_ReleaseIpAddrAssignment` is called for a `LocalAddrId` configured with `TcpIpAssignmentTrigger` set to `TCPIP_MANUAL`, `Tcplp` shall consider the related assignment as unavailable.]

**[SWS\_Tcplp\_00197]** [`TcpIpAddrAssignment` configured with `TcpIpAssignmentTrigger` set to `TCPIP_AUTOMATIC` shall always be available.]

**[SWS\_Tcplp\_00198]** [If `TcpIp_RequestIpAddrAssignment()` is called for a `LocalAddrId` configured with `TcpIpAssignmentTrigger` set to `TCPIP_AUTOMATIC`, `Tcplp` shall reject the request and return `E_NOT_OK`.]

**[SWS\_Tcplp\_00199]** [If `TcpIp_ReleaseIpAddrAssignment()` is called for a `LocalAddrId` configured with `TcpIpAssignmentTrigger` set to `TCPIP_AUTOMATIC`, `Tcplp` shall reject the request and return `E_NOT_OK`.]

## 8.4.3.2 Tcplp\_ReleaselpAddrAssignment

## [SWS\_TCPIP\_00078] Definition of API function Tcplp\_ReleaselpAddrAssignment

[

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

]

[SWS\_Tcplp\_00117] [The service `Tcplp_ReleaseIpAddrAssignment()` shall release the local IP address assignment related to the IP address table entry specified by `LocalAddrId`.]

## 8.4.3.3 Tcplp\_ResetIpAssignment

## [SWS\_TCPIP\_00215] Definition of API function Tcplp\_ResetIpAssignment [

<b>Service Name</b>	Tcplp_ResetIpAssignment	
<b>Syntax</b>	Std_ReturnType Tcplp_ResetIpAssignment ( void )	
<b>Service ID [hex]</b>	0x1b	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	None	
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	Std_ReturnType	E_OK: success E_NOT_OK: switch port could not be initialized
<b>Description</b>	Resets all learned IP-addresses to invalid values.	





Available via	Tcplp.h
---------------	---------

]

#### [SWS\_Tcplp\_00216]

Upstream requirements: [SRS\\_Eth\\_00087](#)

[The service [TcpIp\\_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).]

Note: The next time the [TcpIpAddrAssignment](#) configured with TCPIP\_STORE are started, the related address assignment method are started to obtain new IP addresses.

#### [SWS\_Tcplp\_00217]

Upstream requirements: [SRS\\_Eth\\_00087](#)

[The service [TcpIp\\_ResetIpAssignment\(\)](#) shall be pre compile time configurable On/Off by the configuration parameter: [TcpIpResetIpAssignmentApi](#) (see [\[ECUC\\_Tcplp\\_00182\]](#)).]

### 8.4.3.4 Tcplp\_IcmpTransmit

#### [SWS\_TCPIP\_00039] Definition of API function Tcplp\_IcmpTransmit [

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
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	Std_ReturnType	Result of operation E_OK The ICMP message has been sent successfully E_NOT_OK The ICMP message was not sent.
<b>Description</b>	By this API service the TCP/IP stack sends an ICMP message according to the specified parameters.	
<b>Available via</b>	Tcplp.h	

]

**[SWS\_Tcplp\_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`.]

#### 8.4.3.5 Tcplp\_IcmpV6Transmit

**[SWS\_TCPIP\_00187]** Definition of API function `Tcplp_IcmpV6Transmit` [

<b>Service Name</b>	Tcplp_IcmpV6Transmit	
<b>Syntax</b>	<pre>Std_ReturnType TcpIp_IcmpV6Transmit (     TcpIp_LocalAddrIdType LocalIpAddrId,     const TcpIp_SockAddrType* RemoteAddrPtr,     uint8 HopLimit,     uint8 Type,     uint8 Code,     uint16 DataLength,     const uint8* DataPtr )</pre>	
<b>Service ID [hex]</b>	0x18	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	LocalIpAddrId	IP address identifier representing the local IP address and EthIf controller which shall be used for transmission of the ICMPv6 message.
	RemoteAddrPtr	pointer to struct representing the remote address





	HopLimit	Hop Limit value to be used for the ICMPv6 message. If 0 is specified the default value shall be used.
	Type	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 field value to be used in the ICMPv6 message
	DataLength	length of ICMPv6 message
	DataPtr	Pointer to data which shall be sent as ICMPv6 message data
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	Std_ReturnType	Result of operation E_OK: The ICMPv6 message has been sent successfully E_NOT_OK: The ICMPv6 message was not sent.
<b>Description</b>	By this API service the TCP/IP stack sends an ICMPv6 message according to the specified parameters.	
<b>Available via</b>	Tcplp.h	

]

**[SWS\_Tcplp\_00230]** [The service `TcpIp_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`.]

#### 8.4.3.6 Tcplp\_DhcpReadOption

##### [SWS\_TCPIP\_00040] Definition of API function Tcplp\_DhcpReadOption

Upstream requirements: [SRS\\_Eth\\_00066](#)

[

<b>Service Name</b>	Tcplp_DhcpReadOption	
<b>Syntax</b>	<pre>Std_ReturnType TcpIp_DhcpReadOption (     TcpIp_LocalAddrIdType LocalIpAddrId,     uint8 Option,     uint8* DataLength,     uint8* DataPtr )</pre>	
<b>Service ID [hex]</b>	0x0D	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	LocalIpAddrId	IP address identifier representing the local IP address and EthIf controller for which the DHCP option shall be read.
	Option	DHCP option (note: according to IANA DHCP Options)





<b>Parameters (inout)</b>	DataLength	As input parameter, contains the length of the provided data buffer. Will be overwritten with the length of the actual data.
<b>Parameters (out)</b>	DataPtr	Pointer to memory containing DHCP option data
<b>Return value</b>	Std_ReturnType	Result of operation E_OK requested data retrieved successfully. E_NOT_OK requested data could not be retrieved.
<b>Description</b>	By this API service the TCP/IP stack retrieves DHCP option data identified by parameter option for already received DHCP options.	
<b>Available via</b>	Tcplp.h	

]

**[SWS\_Tcplp\_00233]***Upstream requirements:* [SRS\\_Eth\\_00066](#)

[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.]

**[SWS\_Tcplp\_00234]***Upstream requirements:* [SRS\\_Eth\\_00066](#)

[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.]

**[SWS\_Tcplp\_00235]***Upstream requirements:* [SRS\\_Eth\\_00066](#)

[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.]

**[SWS\_Tcplp\_00236]***Upstream requirements:* [SRS\\_Eth\\_00066](#)

[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.]

**[SWS\_Tcplp\_00237]***Upstream requirements:* [SRS\\_Eth\\_00066](#)

[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.]

## 8.4.3.7 Tcplp\_DhcpV6ReadOption

**[SWS\_TCPIP\_00189] Definition of API function Tcplp\_DhcpV6ReadOption**Upstream requirements: [SRS\\_Eth\\_00066](#)

[

<b>Service Name</b>	Tcplp_DhcpV6ReadOption	
<b>Syntax</b>	<pre>Std_ReturnType Tcplp_DhcpV6ReadOption (     Tcplp_LocalAddrIdType LocalIpAddrId,     uint16 Option,     uint16* DataLength,     uint8* DataPtr )</pre>	
<b>Service ID [hex]</b>	0x19	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	LocalIpAddrId	IP address identifier representing the local IP address and EthIf controller for which the DHCPv6 option shall be read.
	Option	DHCP option (note: according to IANA DHCP[v6] Options)
<b>Parameters (inout)</b>	DataLength	As input parameter, contains the length of the provided data buffer. Will be overwritten with the length of the actual data.
<b>Parameters (out)</b>	DataPtr	Pointer to memory containing DHCPv6 option data
<b>Return value</b>	Std_ReturnType	Result of operation E_OK: requested data retrieved successfully. E_NOT_OK: requested data could not be retrieved.
<b>Description</b>	By this API service the TCP/IP stack retrieves DHCPv6 option data identified by parameter option for already received DHCPv6 options.	
<b>Available via</b>	Tcplp.h	

]

**[SWS\_Tcplp\_00238]**Upstream requirements: [SRS\\_Eth\\_00066](#)

[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.]

**[SWS\_Tcplp\_00239]**Upstream requirements: [SRS\\_Eth\\_00066](#)

[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.]



**[SWS\_Tcplp\_00240]**

Upstream requirements: [SRS\\_Eth\\_00066](#)

[If development error detection is enabled: [TcpIp\\_DhcpV6ReadOption\(\)](#) shall check if the parameter [DataLength](#) is valid (i.e. the buffer is large enough for the requested option). If the check fails, [TcpIp\\_DhcpV6ReadOption\(\)](#) shall raise the development error `TCPIP_E_INV_ARG`.]

**[SWS\_Tcplp\_00241]**

Upstream requirements: [SRS\\_Eth\\_00066](#)

[If the requested option has been set for the address specified by [LocalIpAddrId](#), [TcpIp\\_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`.]

**[SWS\_Tcplp\_00242]**

Upstream requirements: [SRS\\_Eth\\_00066](#)

[If the requested option has not been set for the address specified by [LocalIpAddrId](#), [TcpIp\\_DhcpV6ReadOption\(\)](#) shall set the parameter [DataLength](#) to zero, leave the buffer provided by [DataPtr](#) unchanged and return `E_OK`.]

**8.4.3.8 Tcplp\_DhcpWriteOption****[SWS\_TCPIP\_00020] Definition of API function Tcplp\_DhcpWriteOption**

Upstream requirements: [SRS\\_Eth\\_00065](#)

[

Service Name	Tcplp_DhcpWriteOption	
Syntax	<pre>Std_ReturnType TcpIp_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 (note: according to IANA DHCP Options)
	DataLength	length of DHCP option data
	DataPtr	Pointer to memory containing DHCP option data
Parameters (inout)	None	





<b>Parameters (out)</b>	None	
<b>Return value</b>	Std_ReturnType	Result of operation E_OK no error occurred. E_NOT_OK DHCP option data could not be written.
<b>Description</b>	By this API service the TCP/IP stack writes the DHCP option data identified by parameter option.	
<b>Available via</b>	TcpIp.h	

]

### [SWS\_Tcplp\_00243]

*Upstream requirements:* [SRS\\_Eth\\_00065](#)

[If development error detection is enabled: [TcpIp\\_DhcpWriteOption\(\)](#) shall check if the parameter [LocalIpAddrId](#) is valid. If the check fails, [TcpIp\\_DhcpWriteOption\(\)](#) shall raise the development error `TCPIP_E_INV_ARG`.]

### [SWS\_Tcplp\_00244]

*Upstream requirements:* [SRS\\_Eth\\_00065](#)

[If development error detection is enabled: [TcpIp\\_DhcpWriteOption\(\)](#) shall check if the parameter [Option](#) is valid. If the check fails, [TcpIp\\_DhcpWriteOption\(\)](#) shall raise the development error `TCPIP_E_INV_ARG`.]

### [SWS\_Tcplp\_00245]

*Upstream requirements:* [SRS\\_Eth\\_00065](#)

[If development error detection is enabled: [TcpIp\\_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, [TcpIp\\_DhcpWriteOption\(\)](#) shall raise the development error `TCPIP_E_INV_ARG`.]

### [SWS\_Tcplp\_00246]

*Upstream requirements:* [SRS\\_Eth\\_00065](#)

[If the length indicated by [DataLength](#) is larger than zero [TcpIp\\_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`.]

### [SWS\_Tcplp\_00247]

*Upstream requirements:* [SRS\\_Eth\\_00065](#)

[If the length indicated by [DataLength](#) is equal to zero [TcpIp\\_DhcpWriteOption\(\)](#) shall unset the option identified by [Option](#) for the address specified by [LocalIpAddrId](#) and return `E_OK`.]

### 8.4.3.9 Tcplp\_DhcpV6WriteOption

#### [SWS\_TCPIP\_00190] Definition of API function Tcplp\_DhcpV6WriteOption

Upstream requirements: [SRS\\_Eth\\_00065](#)

<b>Service Name</b>	Tcplp_DhcpV6WriteOption	
<b>Syntax</b>	Std_ReturnType Tcplp_DhcpV6WriteOption ( Tcplp_LocalAddrIdType LocalIpAddrId, uint16 Option, uint16 DataLength, const uint8* DataPtr )	
<b>Service ID [hex]</b>	0x1a	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	LocalIpAddrId	IP address identifier representing the local IP address and EthIf controller for which the DHCPv6 option shall be written.
	Option	DHCP option (note: according to IANA DHCP[v6] Options
	DataLength	length of DHCPv6 option data
	DataPtr	Pointer to memory containing DHCPv6 option data
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	Std_ReturnType	Result of operation E_OK: no error occurred. E_NOT_OK: DHCPv6 option data could not be written.
<b>Description</b>	By this API service the TCP/IP stack writes the DHCPv6 option data identified by parameter option.	
<b>Available via</b>	Tcplp.h	

#### [SWS\_Tcplp\_00248]

Upstream requirements: [SRS\\_Eth\\_00065](#)

[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.]

#### [SWS\_Tcplp\_00249]

Upstream requirements: [SRS\\_Eth\\_00065](#)

[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.]

#### [SWS\_Tcplp\_00250]

Upstream requirements: [SRS\\_Eth\\_00065](#)

[If development error detection is enabled: [TcpIp\\_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, [TcpIp\\_DhcpV6WriteOption\(\)](#) shall raise the development error `TCPIP_E_INV_ARG`.]

#### [SWS\_Tcplp\_00251]

Upstream requirements: [SRS\\_Eth\\_00065](#)

[If the length indicated by [DataLength](#) is larger than zero [TcpIp\\_DhcpV6WriteOption\(\)](#) shall set the option identified by [Option](#) to the value provided by [DataPtr](#) internally for the address specified by [LocalIpAddrId](#) and return `E_OK`.]

#### [SWS\_Tcplp\_00252]

Upstream requirements: [SRS\\_Eth\\_00065](#)

[If the length indicated by [DataLength](#) is equal to zero [TcpIp\\_DhcpV6WriteOption\(\)](#) shall unset the option identified by [Option](#) for the address specified by [LocalIpAddrId](#) and return `E_OK`.]

### 8.4.3.10 Tcplp\_ChangeParameter

#### [SWS\_TCPIP\_00016] Definition of API function [Tcplp\\_ChangeParameter](#) [

<b>Service Name</b>	Tcplp_ChangeParameter	
<b>Syntax</b>	<pre>Std_ReturnType TcpIp_ChangeParameter (     TcpIp_SocketIdType SocketId,     TcpIp_ParamIdType ParameterId,     const uint8* ParameterValue )</pre>	
<b>Service ID [hex]</b>	0x0F	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different SocketIds. Non reentrant for the same SocketId.	
<b>Parameters (in)</b>	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
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	Std_ReturnType	<code>E_OK</code> : The parameter has been changed successfully. <code>E_NOT_OK</code> : The parameter could not be changed.
<b>Description</b>	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.	





Available via	Tcplp.h
---------------	---------

]

[SWS\_Tcplp\_00119] [The service `TcpIp_ChangeParameter()` shall change the parameter specified by `ParameterId` with the value (casted to the respective data type) specified by `ParameterValue` for the `SocketId`.]

#### 8.4.3.11 Tcplp\_GetIpAddr

[SWS\_TCPIP\_00032] Definition of API function `Tcplp_GetIpAddr` [

<b>Service Name</b>	Tcplp_GetIpAddr	
<b>Syntax</b>	<pre>Std_ReturnType TcpIp_GetIpAddr (     TcpIp_LocalAddrIdType LocalAddrId,     TcpIp_SockAddrType* IpAddrPtr,     uint8* NetmaskPtr,     TcpIp_SockAddrType* DefaultRouterPtr )</pre>	
<b>Service ID [hex]</b>	0x10	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	LocalAddrId	Local address identifier referring to the local IP address which shall be obtained.
<b>Parameters (inout)</b>	IpAddrPtr	Pointer to a struct where the IP address shall be stored. The struct member domain shall be set to the desired <code>Tcplp_DomainType</code> 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 <code>IpAddrPtr</code> .
<b>Parameters (out)</b>	NetmaskPtr	Pointer to memory where Network mask of IPv4 address or address prefix of IPv6 address in CIDR Notation is stored
<b>Return value</b>	Std_ReturnType	Result of operation E_OK: The request was successful E_NOT_OK: The request was not successful, e.g. domain in <code>IpAddrPtr</code> and the local domain type do not match
<b>Description</b>	Obtains the local IP address actually used by <code>LocalAddrId</code> , the netmask and default router	
<b>Available via</b>	Tcplp.h	

]

[SWS\_Tcplp\_00205] [`TcpIp_GetIpAddr()` shall refuse the request if the domain set in `IpAddrPtr` does not match the `TcpIp_DomainType` of the selected local address and return `E_NOT_OK`. If development error detection is enabled, the service `TcpIp_GetIpAddr()` shall also raise the development error `TCPIP_E_INV_ARG`.]

[SWS\_Tcplp\_00206] [`TcpIp_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 `TcpIp_GetIpAddr()` shall also raise the development error `TCPIP_E_INV_ARG`.]

#### 8.4.3.12 Tcplp\_GetPhysAddr

[SWS\_TCPIP\_00033] Definition of API function `Tcplp_GetPhysAddr` [

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

]

#### 8.4.3.13 Tcplp\_GetRemotePhysAddr

[SWS\_TCPIP\_00137] Definition of API function `Tcplp_GetRemotePhysAddr` [

<b>Service Name</b>	Tcplp_GetRemotePhysAddr	
<b>Syntax</b>	<pre>TcpIp_ReturnType TcpIp_GetRemotePhysAddr (     uint8 CtrlIdx,     const TcpIp_SockAddrType* IpAddrPtr,     uint8* PhysAddrPtr,     boolean initRes )</pre>	
<b>Service ID [hex]</b>	0x16	





<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	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.
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	PhysAddrPtr	Pointer to the memory where the physical address (MAC address) related to the specified IP address is stored in network byte order.
<b>Return value</b>	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)
<b>Description</b>	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.	
<b>Available via</b>	Tcplp.h	

]

**[SWS\_Tcplp\_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\_Tcplp\_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.]

#### 8.4.3.14 Tcplp\_GetCtrlIdx

##### [SWS\_TCPIP\_00140] Definition of API function Tcplp\_GetCtrlIdx [

<b>Service Name</b>	Tcplp_GetCtrlIdx	
<b>Syntax</b>	<pre>Std_ReturnType TcpIp_GetCtrlIdx (     TcpIp_LocalAddrIdType LocalAddrId,     uint8* CtrlIdxPtr )</pre>	
<b>Service ID [hex]</b>	0x17	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	LocalAddrId	Local address identifier implicitly specifying the EthIf controller that shall be returned.
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	CtrlIdxPtr	Pointer to the memory where the index of the controller related to LocalAddrId is stored
<b>Return value</b>	Std_ReturnType	Result of operation E_OK the request was successful E_NOT_OK the request was not successful.
<b>Description</b>	Tcplp_GetCtrlIdx returns the index of the controller related to LocalAddrId.	
<b>Available via</b>	Tcplp.h	

]

[SWS\_Tcplp\_00141] [TcpIp\_GetCtrlIdx()] shall return the index of the controller related to LocalAddrId.]

#### 8.4.3.15 Tcplp\_GetArpCacheEntries

##### [SWS\_TCPIP\_91002] Definition of API function Tcplp\_GetArpCacheEntries [

<b>Service Name</b>	Tcplp_GetArpCacheEntries	
<b>Syntax</b>	<pre>Std_ReturnType TcpIp_GetArpCacheEntries (     uint8 ctrlIdx,     uint32* numberOfElements,     TcpIp_ArpCacheEntryType* entryListPtr )</pre>	
<b>Service ID [hex]</b>	0x1d	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	ctrlIdx	EthIf controller index to identify the related ARP table.
<b>Parameters (inout)</b>	numberOfElements	In: Maximum number of entries that can be stored in output entry ListPtr. Out: Number of entries written to output entryListPtr (Number of all entries in the cache if input value is 0).
<b>Parameters (out)</b>	entryListPtr	Pointer to memory where the list of cache entries shall be stored.







<b>Return value</b>	Std_ReturnType	E_OK: physical address cache could be read. E_NOT_OK: physical address cache could not be read (i.e. no IPv4 instance active on this controller)
<b>Description</b>	Copies entries from the physical address cache of the IPv4 instance that is active on the EthIf controller specified by ctrlIdx into a user provided buffer. The function will copy all or numberOfElements into the output list. If input value of numberOfElements is 0 the function will not copy any data but only return the number of valid entries in the cache. EntryListPtr may be NULL_PTR in this case.	
<b>Available via</b>	Tcplp.h	

]

**[SWS\_Tcplp\_00271]** [Tcplp\_GetArpCacheEntries() shall only consider entryListPtr set to NULL\_PTR as valid if numberOfElements is set to zero.]

**[SWS\_Tcplp\_00272]** [If Tcplp\_GetArpCacheEntries() is called with numberOfElements set to zero, Tcplp shall set the parameter numberOfElements to the number of valid entries in the physical address cache related to ctrlIdx, leave the buffer provided by entryListPtr unchanged and return E\_OK.]

**[SWS\_Tcplp\_00273]** [If the numberOfElements is greater zero, Tcplp\_GetArpCacheEntries() shall copy up to that number of valid entries from the physical address cache related to ctrlIdx into the buffer provided by entryListPtr, set the parameter numberOfElements to the number of copied elements and return E\_OK.]

#### 8.4.3.16 Tcplp\_GetNdpCacheEntries

**[SWS\_TCPIP\_91001]** Definition of API function Tcplp\_GetNdpCacheEntries [

<b>Service Name</b>	Tcplp_GetNdpCacheEntries	
<b>Syntax</b>	Std_ReturnType Tcplp_GetNdpCacheEntries ( uint8 ctrlIdx, uint32* numberOfElements, Tcplp_NdpCacheEntryType* entryListPtr )	
<b>Service ID [hex]</b>	0x1c	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	ctrlIdx	EthIf controller index to identify the related NDP table.
<b>Parameters (inout)</b>	numberOfElements	In: Maximum number of entries that can be stored in output entry ListPtr. Out: Number of entries written to output entryListPtr (Number of all entries in the cache if input value is 0).
<b>Parameters (out)</b>	entryListPtr	Pointer to memory where the list of cache entries shall be stored.





<b>Return value</b>	Std_ReturnType	E_OK: physical address cache could be read. E_NOT_OK: physical address cache could not be read (i.e. no IPv6 instance active on this controller)
<b>Description</b>	Copies entries from the physical address cache of the IPv6 instance that is active on the EthIf controller specified by ctrlIdx into a user provided buffer. The function will copy all or numberOfElements into the output list. If input value of numberOfElements is 0 the function will not copy any data but only return the number of valid entries in the cache. EntryListPtr may be NULL_PTR in this case.	
<b>Available via</b>	Tcplp.h	

]

[SWS\_Tcplp\_00274] [TcpIp\_GetNdpCacheEntries() shall only consider entryListPtr set to NULL\_PTR as valid if numberOfElements is set to zero.]

[SWS\_Tcplp\_00275] [If TcpIp\_GetNdpCacheEntries() is called with numberOfElements set to zero, Tcplp shall set the parameter numberOfElements to the number of valid entries in the physical address cache related to ctrlIdx, leave the buffer provided by entryListPtr unchanged and return E\_OK.]

[SWS\_Tcplp\_00276] [If the numberOfElements is greater zero, TcpIp\_GetNdpCacheEntries() shall copy up to that number of valid entries from the physical address cache related to ctrlIdx into the buffer provided by entryListPtr, set the parameter numberOfElements to the number of copied elements and return E\_OK.]

#### 8.4.3.17 Tcplp\_GetAndResetMeasurementData

[SWS\_TCPIP\_91006] Definition of API function Tcplp\_GetAndResetMeasurementData [

<b>Service Name</b>	Tcplp_GetAndResetMeasurementData	
<b>Syntax</b>	Std_ReturnType TcpIp_GetAndResetMeasurementData ( TcpIp_MeasurementIdxType MeasurementIdx, boolean MeasurementResetNeeded, uint32* MeasurementDataPtr )	
<b>Service ID [hex]</b>	0x45	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	MeasurementIdx	Data index of measurement data
	MeasurementResetNeeded	Flag to trigger a reset of the measurement data
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	MeasurementDataPtr	Reference to data buffer, where to copy measurement data





<b>Return value</b>	Std_ReturnType	E_OK: successful E_NOT_OK: failed
<b>Description</b>	Allows to read and reset detailed measurement data for diagnostic purposes. Get all MeasurementIdx's at once is not supported. TCPIP_MEAS_ALL shall only be used to reset all MeasurementIdx's at once. A NULL_PTR shall be provided for MeasurementDataPtr in this case.	
<b>Available via</b>	Tcplp.h	

]

### [SWS\_Tcplp\_00284]

*Upstream requirements:* [SRS\\_Eth\\_00129](#)

[The function [TcpIp\\_GetAndResetMeasurementData\(\)](#) shall be pre compile time configurable On/Off by the configuration parameter: [TcpIpGetAndResetMeasurementDataApi](#).]

### [SWS\_Tcplp\_00295]

*Upstream requirements:* [SRS\\_Eth\\_00129](#)

[[TcpIp\\_GetAndResetMeasurementData\(\)](#) shall accept [MeasurementDataPtr](#) set to NULL\_PTR. In this case the measurement data shall not be copied.]

### [SWS\_Tcplp\_00286]

*Upstream requirements:* [SRS\\_Eth\\_00129](#)

[[TcpIp\\_GetAndResetMeasurementData\(\)](#) shall return measurement data for selected measurement index.]

### [SWS\_Tcplp\_00287]

*Upstream requirements:* [SRS\\_Eth\\_00129](#)

[For measurement index TCPIP\_MEAS\_DROP\_TCP [TcpIp\\_GetAndResetMeasurementData\(\)](#) shall return the number of all TCP datagrams which cannot be mapped to a valid local IP/Port.]

### [SWS\_Tcplp\_00288]

*Upstream requirements:* [SRS\\_Eth\\_00129](#)

[For measurement index TCPIP\_MEAS\_DROP\_UDP [TcpIp\\_GetAndResetMeasurementData\(\)](#) shall return the number of all UDP datagrams which cannot be mapped to a valid local IP/Port.]

**[SWS\_Tcplp\_00289]**

*Upstream requirements:* [SRS\\_Eth\\_00129](#)

[For measurement index `TCPIP_MEAS_DROP_IPV4` [TcpIp\\_GetAndResetMeasurementData\(\)](#) shall return the number of all dropped IPv4 datagrams, caused by invalid IP address.]

**[SWS\_Tcplp\_00290]**

*Upstream requirements:* [SRS\\_Eth\\_00129](#)

[For measurement index `TCPIP_MEAS_DROP_IPV6` [TcpIp\\_GetAndResetMeasurementData\(\)](#) shall return the number of all dropped IPv6 datagrams, caused by invalid IP address.]

**[SWS\_Tcplp\_00291]**

*Upstream requirements:* [SRS\\_Eth\\_00129](#)

[[TcpIp\\_GetAndResetMeasurementData\(\)](#) shall return `E_NOT_OK` if the requested measurement index is not supported.]

**[SWS\_Tcplp\_00292]**

*Upstream requirements:* [SRS\\_Eth\\_00129](#)

[[TcpIp\\_GetAndResetMeasurementData\(\)](#) shall additionally reset the measurement data to 0 if the [MeasurementResetNeeded](#) is true. The reset shall be applied after measurement data has been read.]

**[SWS\_Tcplp\_00293]**

*Upstream requirements:* [SRS\\_Eth\\_00129](#)

[[TcpIp\\_GetAndResetMeasurementData\(\)](#) shall reset all existing measurement data to 0, if [MeasurementResetNeeded](#) is true and measurement index is set to `TCPIP_MEAS_ALL`.]

**[SWS\_Tcplp\_00294]**

*Upstream requirements:* [SRS\\_Eth\\_00129](#)

[All measurement data which counts data shall not overrun.]

## 8.4.3.18 Tcplp\_IsConnectionReady

## [SWS\_TCPIP\_91016] Definition of API function Tcplp\_IsConnectionReady [

<b>Service Name</b>	Tcplp_IsConnectionReady	
<b>Syntax</b>	<pre>TcpIp_ReturnType Tcplp_IsConnectionReady (     TcpIp_SocketIdType SocketId,     const TcpIp_SockAddrType* RemoteAddrPtr )</pre>	
<b>Service ID [hex]</b>	0x46	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different SocketIds. Non reentrant for the same SocketId.	
<b>Parameters (in)</b>	SocketId	Socket handle identifying the local socket resource.
	RemoteAddrPtr	Pointer to the structure containing the requested remote IP address and port.
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	Tcplp_ReturnType	TCPIP_E_OK - SocketId is ready for communication. TCPIP_E_NOT_OK - Request was rejected. TCPIP_E_PENDING - Connection establishment in progress.
<b>Description</b>	API allows to check if a communication over this socket is possible for a dedicated remote address. It includes that the socket is bound, a physical address is available for the requested remote address and if a security association is configured that a secured connection is already established.	
<b>Available via</b>	Tcplp.h	

]

[SWS\_Tcplp\_00369] [If development error detection is enabled and the parameter `RemoteAddrPtr` equals `NULL_PTR`, the `Tcplp_IsConnectionReady()` function shall raise the development error code `TCPIP_E_PARAM_POINTER`.]

## 8.4.4 Transmission

## 8.4.4.1 Tcplp\_UdpTransmit

## [SWS\_TCPIP\_00025] Definition of API function Tcplp\_UdpTransmit [

<b>Service Name</b>	Tcplp_UdpTransmit	
<b>Syntax</b>	<pre>Std_ReturnType Tcplp_UdpTransmit (     TcpIp_SocketIdType SocketId,     const uint8* DataPtr,     const TcpIp_SockAddrType* RemoteAddrPtr,     uint16 TotalLength )</pre>	
<b>Service ID [hex]</b>	0x12	





<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different SocketIds. Non reentrant for the same SocketId.	
<b>Parameters (in)</b>	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.
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	Std_ReturnType	E_OK: Request to transmit the UDP message has been accepted. E_NOT_OK: UDP message could not be sent because of a permanent error, e.g. message is too long.
<b>Description</b>	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 Ethlf.	
<b>Available via</b>	Tcplp.h	

]

[SWS\_Tcplp\_00120] [With respect to [SWS\_Tcplp\_00191] and [SWS\_Tcplp\_00193], 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`.]

Note: Transmission stated in [SWS\_Tcplp\_00120] is done according to the sequence diagram specified in section [Section 9.5](#).

[SWS\_Tcplp\_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\_Tcplp\_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()`.]

#### 8.4.4.2 Tcplp\_TcpTransmit

##### [SWS\_TCPIP\_00050] Definition of API function Tcplp\_TcpTransmit [

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

]

[SWS\_Tcplp\_00123] [The service `TcpIp_TcpTransmit()` shall transmit data via TCP and the socket specified by `SocketId` to the connected remote socket.]

Note: Transmission stated in [SWS\_Tcplp\_00123] is done according to the sequence diagram specified in section [Section 9.4](#).

[SWS\_Tcplp\_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\_Tcplp\_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`, `Tcplp` may retrieve less data then available).]

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

## 8.5 Callback notifications

This is a list of functions provided for other modules.

### 8.5.1 Tcplp\_RxIndication

**[SWS\_TCPIP\_00029]** Definition of callback function `Tcplp_RxIndication` [

<b>Service Name</b>	Tcplp_RxIndication	
<b>Syntax</b>	<pre>void TcpIp_RxIndication (     PduIdType RxPduId,     const PduInfoType* PduInfoPtr )</pre>	
<b>Service ID [hex]</b>	0x14	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different PduIds. Non reentrant for the same PduId.	
<b>Parameters (in)</b>	RxPduId	ID of the received PDU.
	PduInfoPtr	Contains the length ( <code>SduLength</code> ) of the received PDU, a pointer to a buffer ( <code>SduDataPtr</code> ) containing the PDU, and the <code>MetaData</code> related to this PDU.
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	None	
<b>Description</b>	Indication of a received PDU from a lower layer communication interface module.	
<b>Available via</b>	Tcplp.h	

]



## [SWS\_Tcplp\_00415] Error reporting for invalid reception PDU during packet reception

Status: DRAFT

Upstream requirements: [SRS\\_Eth\\_00187](#)

[If development error detection is enabled: [TcpIp\\_RxIndication\(\)](#) shall check if the parameter [RxPduId](#) is valid. If the check fails, [TcpIp\\_RxIndication\(\)](#) shall refuse the request and raise the development error code `TCPIP_E_INVALID_PDU_SDU_ID` instead.]

## 8.5.2 Tcplp\_TriggerTransmit

### [SWS\_TCPIP\_91017] Definition of callback function Tcplp\_TriggerTransmit

Status: DRAFT

[

<b>Service Name</b>	Tcplp_TriggerTransmit (draft)	
<b>Syntax</b>	<pre>Std_ReturnType TcpIp_TriggerTransmit (     PduIdType TxPduId,     PduInfoType* PduInfoPtr )</pre>	
<b>Service ID [hex]</b>	0x41	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different PduIds. Non reentrant for the same PduId.	
<b>Parameters (in)</b>	TxPduId	ID of the SDU that is requested to be transmitted.
<b>Parameters (inout)</b>	PduInfoPtr	Contains a pointer to a buffer (SduDataPtr) to where the SDU data shall be copied, and the available buffer size in SduLength. On return, the service will indicate the length of the copied SDU data in SduLength.
<b>Parameters (out)</b>	None	
<b>Return value</b>	Std_ReturnType	<p><code>E_OK</code>: SDU has been copied and SduLength indicates the number of copied bytes.</p> <p><code>E_NOT_OK</code>: No SDU data has been copied. PduInfoPtr must not be used since it may contain a NULL pointer or point to invalid data.</p>
<b>Description</b>	<p>Within this API, the upper layer module (called module) shall check whether the available data fits into the buffer size reported by PduInfoPtr-&gt;SduLength. If it fits, it shall copy its data into the buffer provided by PduInfoPtr-&gt;SduDataPtr and update the length of the actual copied data in PduInfoPtr-&gt;SduLength. If not, it returns <code>E_NOT_OK</code> without changing PduInfoPtr.</p> <p><b>Tags:</b> atp.Status=draft</p>	
<b>Available via</b>	Tcplp.h	

]

**[SWS\_Tcplp\_00416] Error reporting for invalid transmission PDU during transmission triggering***Status:* DRAFT*Upstream requirements:* [SRS\\_Eth\\_00187](#)

[If development error detection is enabled: [TcpIp\\_TriggerTransmit\(\)](#) shall check if the parameter [TxPduId](#) is valid. If the check fails, [TcpIp\\_TriggerTransmit\(\)](#) shall refuse the request and raise the development error code [TCPIP\\_E\\_INVALID\\_PDU\\_SDU\\_ID](#) instead.]

**[SWS\_Tcplp\_00417] Error reporting for indirect data provision during trigger transmission***Status:* DRAFT*Upstream requirements:* [SRS\\_Eth\\_00187](#)

[If development error detection is enabled: [TcpIp\\_TriggerTransmit\(\)](#) shall check if the Tcplp module requested a transmission with indirect data provision for a specific [TxPduId](#). If the check fails, [TcpIp\\_TriggerTransmit\(\)](#) shall refuse the request and raise the development error code [TCPIP\\_E\\_PDU\\_DATA\\_FAILED](#) instead.]

**[SWS\_Tcplp\_00418] Error reporting for invalid transmission PDU state during transmission triggering***Status:* DRAFT*Upstream requirements:* [SRS\\_Eth\\_00187](#)

[If development error detection is enabled: [TcpIp\\_TriggerTransmit\(\)](#) shall check if the PDU is in state [PDU\\_IN\\_USE](#) for a specific [TxPduId](#). If the check fails, [TcpIp\\_TriggerTransmit\(\)](#) shall refuse the request and raise the development error code [TCPIP\\_E\\_PDU\\_DATA\\_FAILED](#) instead.]

**8.5.3 Tcplp\_TxConfirmation****[SWS\_TCPIP\_91018] Definition of callback function Tcplp\_TxConfirmation***Status:* DRAFT

[

<b>Service Name</b>	Tcplp_TxConfirmation (draft)
<b>Syntax</b>	<pre>void TcpIp_TxConfirmation (     PduIdType TxPduId,     Std_ReturnType result )</pre>
<b>Service ID [hex]</b>	0x40
<b>Sync/Async</b>	Synchronous





<b>Reentrancy</b>	Reentrant for different PduIds. Non reentrant for the same PduId.	
<b>Parameters (in)</b>	TxPduId	ID of the PDU that has been transmitted.
	result	E_OK: The PDU was transmitted. E_NOT_OK: Transmission of the PDU failed.
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	None	
<b>Description</b>	The lower layer communication interface module confirms the transmission of a PDU, or the failure to transmit a PDU. <b>Tags:</b> atp.Status=draft	
<b>Available via</b>	TcpIp.h	

]

### [SWS\_Tcplp\_00419] Error reporting for invalid transmission PDU during transmission confirmation

Status: DRAFT

Upstream requirements: [SRS\\_Eth\\_00187](#)

[If development error detection is enabled: [TcpIp\\_TxConfirmation\(\)](#) shall check if the parameter [TxPduId](#) is valid. If the check fails, [TcpIp\\_TxConfirmation\(\)](#) shall refuse the request and raise the development error code `TCPIP_E_INVALID_PDU_SDU_ID` instead.]

### [SWS\_Tcplp\_00420] Error reporting for invalid PDU state during transmission confirmation

Status: DRAFT

Upstream requirements: [SRS\\_Eth\\_00187](#)

[If development error detection is enabled: [TcpIp\\_TxConfirmation\(\)](#) shall check if the the PDU is in state `PDU_IN_USE` for a specific [TxPduId](#). If the check fails, [TcpIp\\_TxConfirmation\(\)](#) shall refuse the request and raise the development error code `TCPIP_E_PDU_DATA_FAILED` instead.]

## 8.6 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.

### 8.6.1 Terms and definitions

For details refer to the chapter 8.5 "Scheduled functions" in [SWS\\_BSWGeneral](#).

## 8.6.2 Tcplp\_MainFunction

### [SWS\_TCPIP\_00026] Definition of scheduled function Tcplp\_MainFunction [

<b>Service Name</b>	Tcplp_MainFunction
<b>Syntax</b>	void TcpIp_MainFunction ( void )
<b>Service ID [hex]</b>	0x15
<b>Description</b>	Schedules the TCP/IP stack. (Entry point for scheduling)
<b>Available via</b>	SchM_Tcplp.h

]

## 8.7 Expected interfaces

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

### 8.7.1 Mandatory interfaces

Note: This section defines all interfaces, which are required to fulfill the core functionality of the module.

### [SWS\_TCPIP\_00027] Definition of mandatory interfaces required by module Tcp Ip [

<b>API Function</b>	<b>Header File</b>	<b>Description</b>
Dem_SetEventStatus	Dem.h	Called by SW-Cs or BSW modules to report monitor status information to the Dem. BSW modules calling Dem_SetEventStatus can safely ignore the return value. This API will be available only if ({Dem/Dem ConfigSet/DemEventParameter/DemEvent ReportingType} == STANDARD_REPORTING)
Det_ReportRuntimeError	Det.h	Service to report runtime errors. If a callout has been configured then this callout shall be called.
EthIf_GetPhysAddr	EthIf.h	Obtains the physical source address used by the indexed controller
EthIf_SetPhysAddr	EthIf.h	Sets the physical source address used by the indexed controller.
EthSM_TcplpModelIndication	EthSM_Tcplp.h	This service is called by the Tcplp to report the actual Tcplp state (e.g. online, offline).
LSduR_TcplpTransmit (draft)	LSduR_<module>.h	Requests transmission of a PDU.

]

## 8.7.2 Optional interfaces

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

### [SWS\_TCPIP\_00028] Definition of optional interfaces requested by module TcpIp

API Function	Header File	Description
Csm_AEADDecrypt	Csm.h	Uses the given data to perform an AEAD encryption and stores the ciphertext and the MAC in the memory locations pointed by the ciphertext pointer and Tag pointer.
Csm_AEADEncrypt	Csm.h	Uses the given input data to perform a AEAD encryption and stores the ciphertext and the MAC in the memory locations pointed by the ciphertext pointer and Tag pointer.
Csm_Decrypt	Csm.h	Decrypts the given encrypted data and store the decrypted plaintext in the memory location pointed by the result pointer.
Csm_Encrypt	Csm.h	Encrypts the given data and store the ciphertext in the memory location pointed by the result pointer.
Csm_Hash	Csm.h	Uses the given data to perform the hash calculation and stores the hash.
Csm_KeyElementCopy	Csm.h	This function shall copy a key elements from one key to a target key.
Csm_KeyElementCopyPartial	Csm.h	Copies a key element to another key element in the same crypto driver. The keyElementSourceOffset and keyElementCopyLength allows to copy just a part of the source key element into the destination. The offset into the target key is also specified with this function.
Csm_KeyExchangeCalcPubVal	Csm.h	Calculates the public value of the current user for the key exchange and stores the public key in the memory location pointed by the public value pointer.
Csm_KeyExchangeCalcSecret	Csm.h	Calculates the shared secret key for the key exchange with the key material of the key identified by the keyId and the partner public key. The shared secret key is stored as a key element in the same key.
Csm_MacGenerate	Csm.h	Uses the given data to perform a MAC generation and stores the MAC in the memory location pointed to by the MAC pointer.
Csm_MacVerify	Csm.h	Verifies the given MAC by comparing if the MAC is generated with the given data.
Csm_RandomGenerate	Csm.h	Generate a random number and stores it in the memory location pointed by the result pointer.
Csm_SignatureGenerate	Csm.h	Uses the given data to perform the signature calculation and stores the signature in the memory location pointed by the result pointer.
Csm_SignatureVerify	Csm.h	Verifies the given signature by checking if it was generated with the given data.
Det_ReportError	Det.h	Service to report development errors.





API Function	Header File	Description
EthIf_UpdatePhysAddrFilter	EthIf.h	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.
IdsM_SetSecurityEvent (obsolete)	IdsM.h	This API is the application interface to report security events to the IdsM. <b>Tags:</b> atp.Status=obsolete
IdsM_SetSecurityEventWithContext Data (obsolete)	IdsM.h	This API is the application interface to report security events with context data to the IdsM. <b>Tags:</b> atp.Status=obsolete
KeyM_GetCertificate	KeyM.h	This function provides the DER encoded certificate data
KeyM_SetCertificate	KeyM.h	This function provides the certificate data to the key management module to temporarily store the certificate.
KeyM_VerifyCertificate	KeyM.h	This function verifies a certificate that was previously provided with KeyM_SetCertificate() against already stored and provided certificates stored with other certificate IDs.
KeyM_VerifyCertificateChain	KeyM.h	This function performs a certificate verification against a list of certificates. It is a pre-requisite that the certificate that shall be checked has already been written with KeyM_SetCertificate() and that the root certificate is either in the list or is already assigned to one of the other certificates.

]

### 8.7.3 Configurable interfaces

In this section, all interfaces are listed where the target function could be configured. The target function is usually a callback function. The names of this kind of interfaces are 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.

### 8.7.3.1 Tcplp\_<Up>GetSocket

#### [SWS\_TCPIP\_00018] Definition of API function Tcplp\_<Up>GetSocket

Upstream requirements: [SRS\\_Eth\\_00103](#)

[

<b>Service Name</b>	Tcplp_<Up>GetSocket	
<b>Syntax</b>	<pre>Std_ReturnType Tcplp_&lt;Up&gt;GetSocket (     TcpIp_DomainType Domain,     TcpIp_ProtocolType Protocol,     TcpIp_SocketIdType* SocketIdPtr )</pre>	
<b>Service ID [hex]</b>	0x03	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	Domain	IP address family.
	Protocol	Socket protocol as sub-family of parameter type.
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	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.
<b>Return value</b>	Std_ReturnType	Result of operation E_OK The request has been accepted E_NOT_OK The request has not been accepted: no free socket
<b>Description</b>	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.	
<b>Available via</b>	Tcplp.h	

]

[SWS\_Tcplp\_00128] [If development error detection is enabled, the service [TcpIp\\_GetSocket\(\)](#) shall check the parameter [Domain](#) for being valid and raise the development error TCP\_IP\_E\_AFNOSUPPORT if it is invalid.]

#### [SWS\_Tcplp\_00222]

Upstream requirements: [SRS\\_Eth\\_00103](#)

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

### 8.7.3.2 <Up\_PhysAddrTableChg>

**[SWS\_TCPIP\_00143] Definition of configurable interface <Up\_PhysAddrTableChg>** [

<b>Service Name</b>	<Up_PhysAddrTableChg>	
<b>Syntax</b>	<pre>void &lt;Up_PhysAddrTableChg&gt; (     uint8 CtrlIdx,     const TcpIp_SockAddrType* IpAddrPtr,     const uint8* PhysAddrPtr,     boolean valid )</pre>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	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)
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	None	
<b>Description</b>	This API is called by TcpIp in case of a change in the ARP/NDP table related to the controller specified by CtrlIdx.	
<b>Available via</b>	TcpIp_Externals.h	

]

### 8.7.3.3 SocketOwner functions

**[SWS\_Tcplp\_00220]**

*Upstream requirements:* [SRS\\_Eth\\_00103](#)

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

**[SWS\_Tcplp\_00221]**

*Upstream requirements:* [SRS\\_Eth\\_00103](#)

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



### 8.7.3.3.1 <Up\_RxIndication>

#### [SWS\_TCPIP\_00223] Definition of configurable interface <Up\_RxIndication>

Upstream requirements: [SRS\\_Eth\\_00103](#)

<b>Service Name</b>	<Up_RxIndication>	
<b>Syntax</b>	<pre>void &lt;Up_RxIndication&gt; (     TcpIp_SocketIdType SocketId,     const TcpIp_SockAddrType* RemoteAddrPtr,     const uint8* BufPtr,     uint16 Length )</pre>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different SocketIds. Non reentrant for the same SocketId.	
<b>Parameters (in)</b>	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.
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	None	
<b>Description</b>	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.	
<b>Available via</b>	SoAd.h if the respective TcpIpSocketOwnerUpperLayerType is SOAD, otherwise TcpIpSocketOwnerHeaderFileName	

### 8.7.3.3.2 <Up\_TcplpEvent>

#### [SWS\_TCPIP\_00224] Definition of configurable interface <Up\_TcplpEvent>

Upstream requirements: [SRS\\_Eth\\_00103](#)

<b>Service Name</b>	<Up_TcplpEvent>	
<b>Syntax</b>	<pre>void &lt;Up_TcplpEvent&gt; (     TcpIp_SocketIdType SocketId,     TcpIp_EventType Event )</pre>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	SocketId	Socket identifier of the related local socket resource.
	Event	This parameter contains a description of the event just encountered.





<b>Parameters (inout)</b>	None
<b>Parameters (out)</b>	None
<b>Return value</b>	None
<b>Description</b>	This service gets called if the stack encounters a condition described by the values in Event.
<b>Available via</b>	SoAd.h if the respective TcpIpSocketOwnerUpperLayerType is SOAD, otherwise TcpIpSocketOwnerHeaderFileName

### 8.7.3.3.3 <Up\_TxConfirmation>

#### [SWS\_TCPIP\_00225] Definition of configurable interface <Up\_TxConfirmation>

Upstream requirements: [SRS\\_Eth\\_00103](#)

<b>Service Name</b>	<Up_TxConfirmation>	
<b>Syntax</b>	<pre>void &lt;Up_TxConfirmation&gt; (     TcpIp_SocketIdType SocketId,     uint16 Length )</pre>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant for different SocketIds. Non reentrant for the same SocketId.	
<b>Parameters (in)</b>	SocketId	Socket identifier of the related local socket resource.
	Length	Number of transmitted data bytes.
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	None	
<b>Description</b>	<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>	
<b>Available via</b>	SoAd.h if the respective TcpIpSocketOwnerUpperLayerType is SOAD, otherwise TcpIpSocketOwnerHeaderFileName	

## 8.7.3.3.4 &lt;Up\_TcpAccepted&gt;

**[SWS\_TCPIP\_00226] Definition of configurable interface <Up\_TcpAccepted>**Upstream requirements: [SRS\\_Eth\\_00103](#)

[

<b>Service Name</b>	<Up_TcpAccepted>	
<b>Syntax</b>	<pre>Std_ReturnType &lt;Up_TcpAccepted&gt; (     TcpIp_SocketIdType SocketId,     TcpIp_SocketIdType SocketIdConnected,     const TcpIp_SockAddrType* RemoteAddrPtr )</pre>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	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.
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	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.
<b>Description</b>	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). 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.	
<b>Available via</b>	SoAd.h if the respective TcplpSocketOwnerUpperLayerType is SOAD, otherwise TcplpSocketOwnerHeaderFileName	

]

## 8.7.3.3.5 &lt;Up\_TcpConnected&gt;

**[SWS\_TCPIP\_00227] Definition of configurable interface <Up\_TcpConnected>**Upstream requirements: [SRS\\_Eth\\_00103](#)

[

<b>Service Name</b>	<Up_TcpConnected>	
<b>Syntax</b>	<pre>void &lt;Up_TcpConnected&gt; (     TcpIp_SocketIdType SocketId )</pre>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	SocketId	Socket identifier of the related local socket resource.





<b>Parameters (inout)</b>	None
<b>Parameters (out)</b>	None
<b>Return value</b>	None
<b>Description</b>	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 <code>Tcplp_TcpConnect()</code> and a TCP connection is confirmed by the peer. The parameter value of <code>SocketId</code> equals the <code>SocketId</code> value of the preceding <code>Tcplp_TcpConnect()</code> call.
<b>Available via</b>	SoAd.h if the respective <code>TcplpSocketOwnerUpperLayerType</code> is SOAD, otherwise <code>TcplpSocketOwnerHeaderFileName</code>

]

### 8.7.3.3.6 <Up\_CopyTxData>

#### [SWS\_TCPIP\_00228] Definition of configurable interface <Up\_CopyTxData>

Upstream requirements: [SRS\\_Eth\\_00103](#)

[

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

]

### 8.7.3.3.7 <Up\_LocalIpAddrAssignmentChg>

#### [SWS\_TCPIP\_00229] Definition of configurable interface <Up\_LocalIpAddrAssignmentChg>

Upstream requirements: [SRS\\_Eth\\_00103](#)

[

<b>Service Name</b>	<Up_LocalIpAddrAssignmentChg>	
<b>Syntax</b>	<pre>void &lt;Up_LocalIpAddrAssignmentChg&gt; (     TcpIp_LocalAddrIdType IpAddrId,     TcpIp_IpAddrStateType State )</pre>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	IpAddrId	IP address Identifier, representing an IP address specified in the TcpIp module configuraiton (e.g. static IPv4 address on EthIf controller 0).
	State	state of IP address assignment
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	None	
<b>Description</b>	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).	
<b>Available via</b>	SoAd.h if the respective TcpIpSocketOwnerUpperLayerType is SOAD, otherwise TcpIpSocket OwnerHeaderFileName	

]

### 8.7.3.4 <Up\_IcmpMsgHandler>

#### [SWS\_TCPIP\_00270] Definition of configurable interface <Up\_IcmpMsgHandler>

[

<b>Service Name</b>	<Up_IcmpMsgHandler>	
<b>Syntax</b>	<pre>void &lt;Up_IcmpMsgHandler&gt; (     TcpIp_LocalAddrIdType LocalAddrId,     const TcpIp_SockAddrType* RemoteAddrPtr,     uint8 Ttl,     uint8 Type,     uint8 Code,     uint16 DataLength,     uint8* DataPtr )</pre>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	LocalAddrId	Local address identifier representing the local IP address and Eth If controller where the ICMP message has been received.





	RemoteAddrPtr	pointer to struct representing the address of the ICMP sender
	Ttl	Time to live value of the received ICMPv4 message or Hop Limit value of the received ICMPv6 message.
	Type	type field value of the received ICMP message (Note: the value of the type field determines the format of the remaining ICMP message data)
	Code	code field value of the received ICMP message
	DataLength	length of ICMP message
	DataPtr	Pointer to the received ICMP message
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	None	
<b>Description</b>	By this API service the configured ICMP message handler function is called by the TCP/IP stack on reception of a ICMP message which is not handled by the TCP/IP stack.	
<b>Available via</b>	TcpIp_Externals.h	

」

### 8.7.3.5 <Up\_DADAddressConflict>

[SWS\_TCPIP\_91005] Definition of configurable interface <Up\_DADAddressConflict> 「

<b>Service Name</b>	<Up_DADAddressConflict>	
<b>Syntax</b>	<pre>void &lt;Up_DADAddressConflict&gt; (     TcpIp_LocalAddrIdType IpAddrId,     const TcpIp_SockAddrType* IpAddrPtr,     const uint8* LocalPhysAddrPtr,     const uint8* RemotePhysAddrPtr )</pre>	
<b>Service ID [hex]</b>	0x1e	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	IpAddrId	IP address Identifier, representing an IP address specified in the TcpIp module configuration.
	IpAddrPtr	Pointer to a struct where the conflicted IP address is stored.
	LocalPhysAddrPtr	Pointer to the memory where the local physical address (MAC address) related to the specified IP address is stored in network byte order.
	RemotePhysAddrPtr	Pointer to the memory where the remote physical address (MAC address) related to the specified IP address is stored in network byte order.
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	void	–





<b>Description</b>	This API is called by TcpIp in case the Duplicate Address Detection (DAD) is enabled and detecting a duplicate IP Address.
<b>Available via</b>	TcpIp_Externals.h

]

### [SWS\_Tcplp\_00283]

Upstream requirements: [SRS\\_Eth\\_00091](#), [SRS\\_BSW\\_00452](#)

[If the optional [TcpIpDuplicateAddressDetectionConfig](#) is defined and a duplicate IP address was found by the Duplicate Address Detection (DAD) algorithm, the TcpIp shall call the callout function specified by [TcpIpDuplicateAddressDetectionCalloutName](#).]

### 8.7.3.6 <Up\_TlsGetCurrentTime>

#### [SWS\_TCPIP\_91012] Definition of configurable interface <Up\_TlsGetCurrentTime> [

<b>Service Name</b>	<Up_TlsGetCurrentTime>	
<b>Syntax</b>	Std_ReturnType <Up_TlsGetCurrentTime> ( uint32* CurrentTimeUtc )	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	None	
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	CurrentTimeUtc	Pointer to uint32 to provide the GMT Unix time value.
<b>Return value</b>	Std_ReturnType	E_OK: Time stamp successfully provided. E_NOT_OK: Time stamp can currently not be provided. Data in CurrentTimeUtc not valid.
<b>Description</b>	This function queries the current time. This information will be requested when assembling the client hello message.	
<b>Available via</b>	TcpIp_Externals.h	

]

[SWS\_Tcplp\_00330] [If the optional parameter [TcpIpTlsConnectionGetTimeFunc](#) is defined the TLS\_CLIENT shall call the configured function to query the current time. The value 0 indicates that no time is available. The value 0 is also transmitted if the function returns E\_NOT\_OK.]

[SWS\_Tcplp\_00332] [The function [<Up\\_TlsGetCurrentTime>\(\)](#) shall provide the current UTC time. It is used to assemble the ClientHello handshake message. The

time is provided in big endian format and follows either the GMT Unix time format or can be 0 (See IETF RFC 5246, section 7.4.1.2, `gmt_unix_time` for details).]

### 8.7.3.7 <Up\_TlsServerGetPskIdentityHint>

#### [SWS\_TCPIP\_91013] Definition of configurable interface <Up\_TlsServerGetPskIdentityHint>

Upstream requirements: [SRS\\_Eth\\_00137](#)

[

<b>Service Name</b>	<Up_TlsServerGetPskIdentityHint>	
<b>Syntax</b>	<pre>Std_ReturnType &lt;Up_TlsServerGetPskIdentityHint&gt; (     TcpIp_SocketIdType SocketId,     TcpIp_TlsConnectionIdType TlsConnectionId,     uint16* IdentityHintLengthPtr,     uint8* IdentityHintPtr )</pre>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	SocketId	Socket identifier of the related local socket resource.
	TlsConnectionId	Provides the TLS connection identifier.
<b>Parameters (inout)</b>	IdentityHintLengthPtr	In: Provides the number of bytes available where identityHintPtr links to. Out: Provides the number of bytes that has been overwritten in identityHintPtr.
<b>Parameters (out)</b>	IdentityHintPtr	Ptr to buffer that is used to store the IdentityHint information.
<b>Return value</b>	Std_ReturnType	E_OK: IdentityHint successfully provided E_NOT_OK: IdentityHint could not be provided. Data in the pointer is invalid and shall not be used.
<b>Description</b>	Queries the Identity hint for a pre-shared key ciphersuite. This information is transmitted by the TLS Server to provide its identification to the TLS client.	
<b>Available via</b>	Tcplp_Externals.h	

]

[SWS\_Tcplp\_00333] [If the `TLS_SERVER` selects a PSK ciphersuite from the offered ciphersuite list and `TcpIpTlsPresharedKeyIdentityHint` is not defined but `<Up_TlsServerGetPskIdentityHint>()` is defined, then this function shall be called when the `TLS_SERVER` assembles the `ServerKeyExchange` message (according to RFC4279, Sect. 2) during the handshake to query the `psk_identity_hint`.]



### 8.7.3.8 <Up\_TlsClientGetPskIdentity >

#### [SWS\_TCPIP\_91014] Definition of configurable interface <Up\_TlsClientGetPskIdentity>

Upstream requirements: [SRS\\_Eth\\_00137](#)

<b>Service Name</b>	<Up_TlsClientGetPskIdentity>	
<b>Syntax</b>	<pre>Std_ReturnType &lt;Up_TlsClientGetPskIdentity&gt; (     TcpIp_SocketIdType SocketId,     TcpIp_TlsConnectionIdType TlsConnectionId,     uint16 PskIdentityHintLength,     const uint8* PskIdentityHintPtr,     uint16* PskKeyIdentityLengthPtr,     uint8* PskKeyIdentityPtr,     uint32* CsmKeyId )</pre>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	SocketId	Socket identifier of the related local socket resource.
	TlsConnectionId	Provides the TLS connection identifier.
	PskIdentityHintLength	Provides the number of bytes available in identityHintPtr.
	PskIdentityHintPtr	Pointer to the identity hint information from the server.
<b>Parameters (inout)</b>	PskKeyIdentityLengthPtr	In: Provides the number of bytes available in PskKeyIdentityPtr. Out: Provides the actual number of bytes that has been written to PskKeyIdentityPtr.
<b>Parameters (out)</b>	PskKeyIdentityPtr	Buffer that is used to store the pre-shared key identification.
	CsmKeyId	Provides the identifier of a CSM key.
<b>Return value</b>	Std_ReturnType	E_OK: Pre-Shared key selected properly. All output values are valid. E_NOT_OK: Pre-Shared key could not be selected. Key selection failed.
<b>Description</b>	This function is called on the TLS client side. It provides the key identification based on the identity hint provided by the TLS server. The TLS client selects the pre-shared key and returns the key identification name and the CSM key reference.	
<b>Available via</b>	TcpIp_Externals.h	

[SWS\_Tcplp\_00334] [If the TLS\_CLIENT receives a selected PSK ciphersuite and [TcpIpTlsPresharedKeyIdentityHint](#) or [TcpIpTlsPresharedKeyIdentity](#) or [TcpIpTlsPresharedKeyCsmKeyRef](#) is not defined but [<Up\\_TlsClientGetPskIdentity>\(\)](#) is defined, then this function shall be called when the TLS\_CLIENT assembles the ClientKeyExchange message (according to RFC4279, Sect. 2). The function provides the pre-shared key and the psk\_identity which is provided in the ClientKeyExchange message.]

### 8.7.3.9 <Up\_TlsServerGetPskIdentity>

#### [SWS\_TCPIP\_91015] Definition of configurable interface <Up\_TlsServerGetPskIdentity>

Upstream requirements: [SRS\\_Eth\\_00137](#)

[

<b>Service Name</b>	<Up_TlsServerGetPskIdentity>	
<b>Syntax</b>	<pre>Std_ReturnType &lt;Up_TlsServerGetPskIdentity&gt; (     TcpIp_SocketIdType SocketId,     TcpIp_TlsConnectionIdType TlsConnectionId,     uint16 PskKeyIdentityLength,     const uint8* PskKeyIdentityPtr,     uint32* CsmKeyId )</pre>	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	SocketId	Socket identifier of the related local socket resource.
	TlsConnectionId	Provides the TLS connection identifier.
	PskKeyIdentityLength	Provides the number of bytes available in PskKeyIdentityPtr.
	PskKeyIdentityPtr	Pointer to a buffer that provides the PSK key identification information.
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	CsmKeyId	Provides the identifier of a CSM key.
<b>Return value</b>	Std_ReturnType	E_OK: PSK key was identified and CsmKey reference provided properly.
		E_NOT_OK: Key identification or PSK key could not be identified.
<b>Description</b>	This callback is used for the TLS server to provide the CSM key name according to the key identification that was selected by the TLS client. The TLS server must provide a CsmKey reference to a key that matches this key identification name.	
<b>Available via</b>	Tcplp_Externals.h	

]

[SWS\_Tcplp\_00335] [If the TLS\_SERVER receives the ClientKeyExchange message during the handshake and [TcpIpTlsPresharedKeyIdentity](#) or [TcpIpTlsPresharedKeyCsmKeyRef](#) is not defined but [<Up\\_TlsServerGetPskIdentity>\(\)](#) is defined, then this function shall be called when the TLS\_CLIENT assembles the ClientKeyExchange message (according to RFC4279, Sect. 2). The function provides the pre-shared key and the psk\_identity which is provided in the ClientKeyExchange message.]

## 8.8 Service Interfaces

No service interfaces provided.

## 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

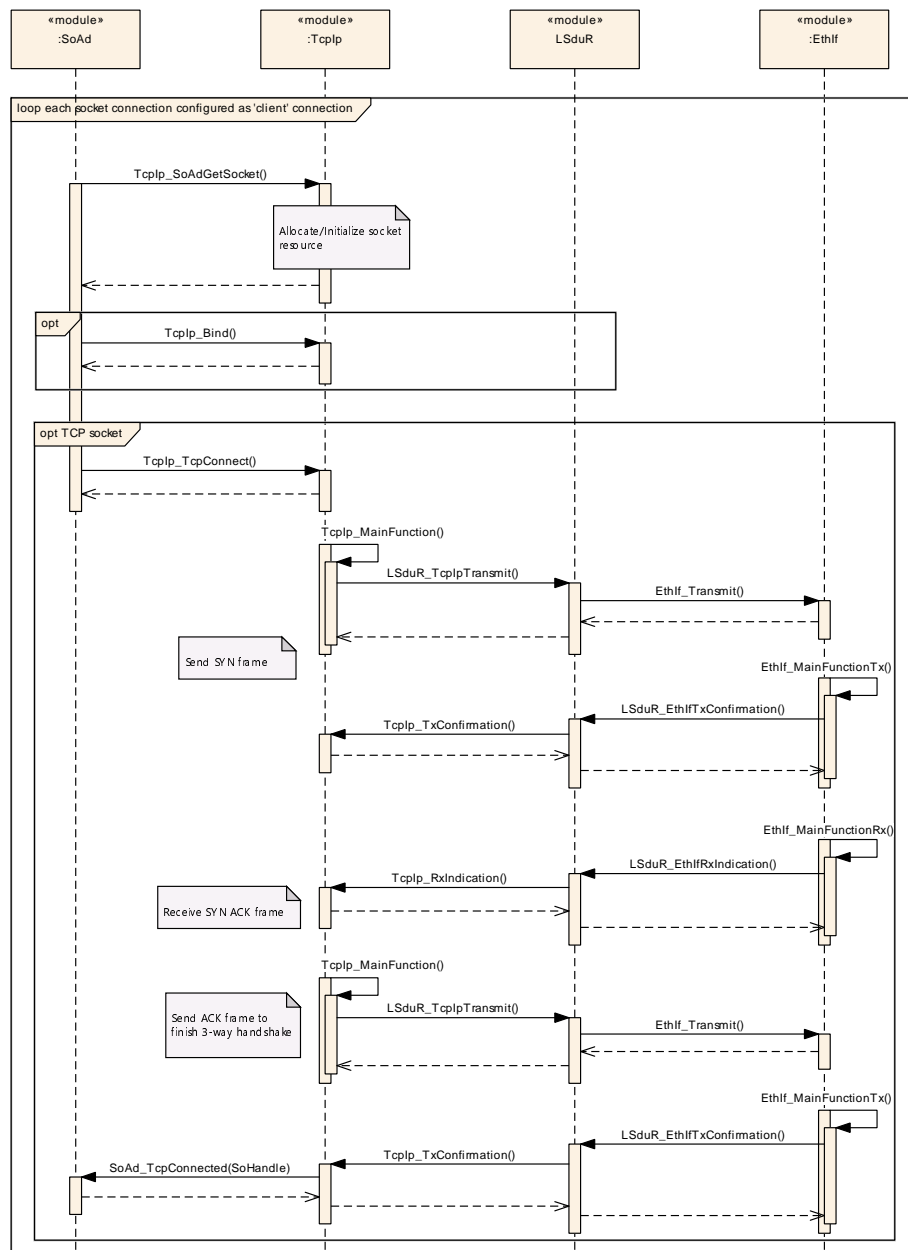
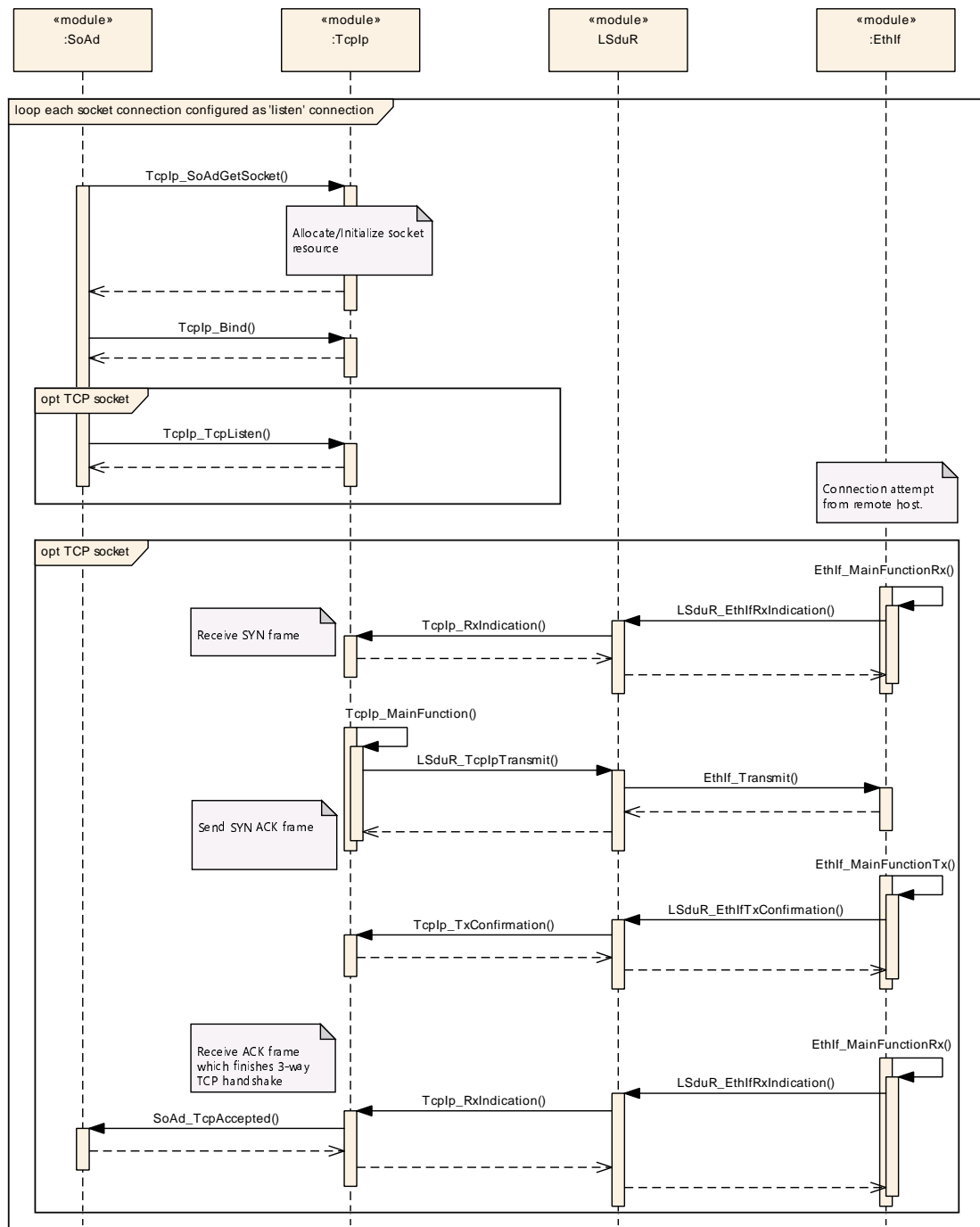


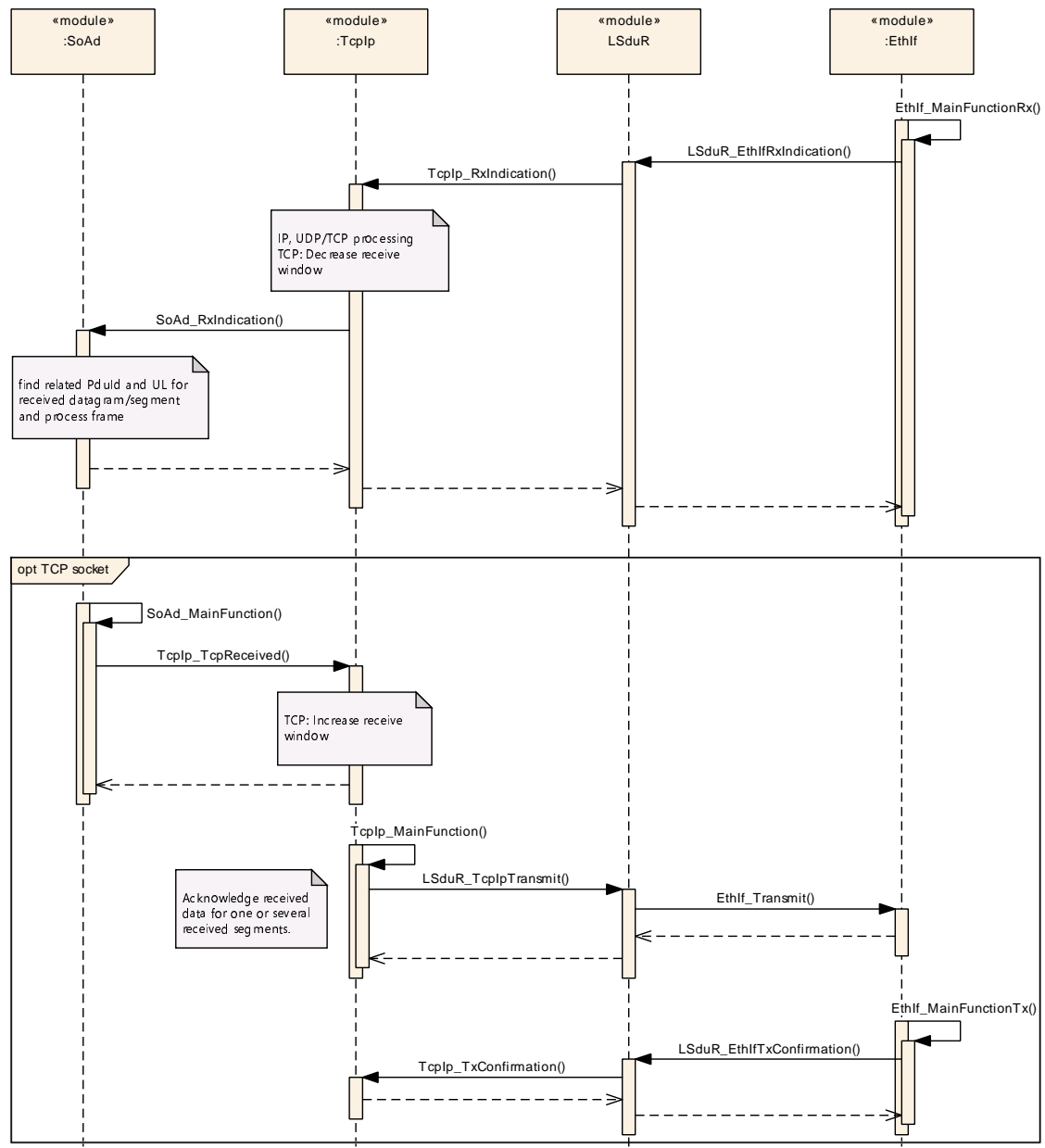
Figure 9.1: TcpIp TCP connection Setup Client

## 9.2 TCP Connection Setup - Server



**Figure 9.2: TcpIp TCP connection Setup Server**

### 9.3 Reception



**Figure 9.3: Tcplp Rx**

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

## 9.4 Transmission TCP

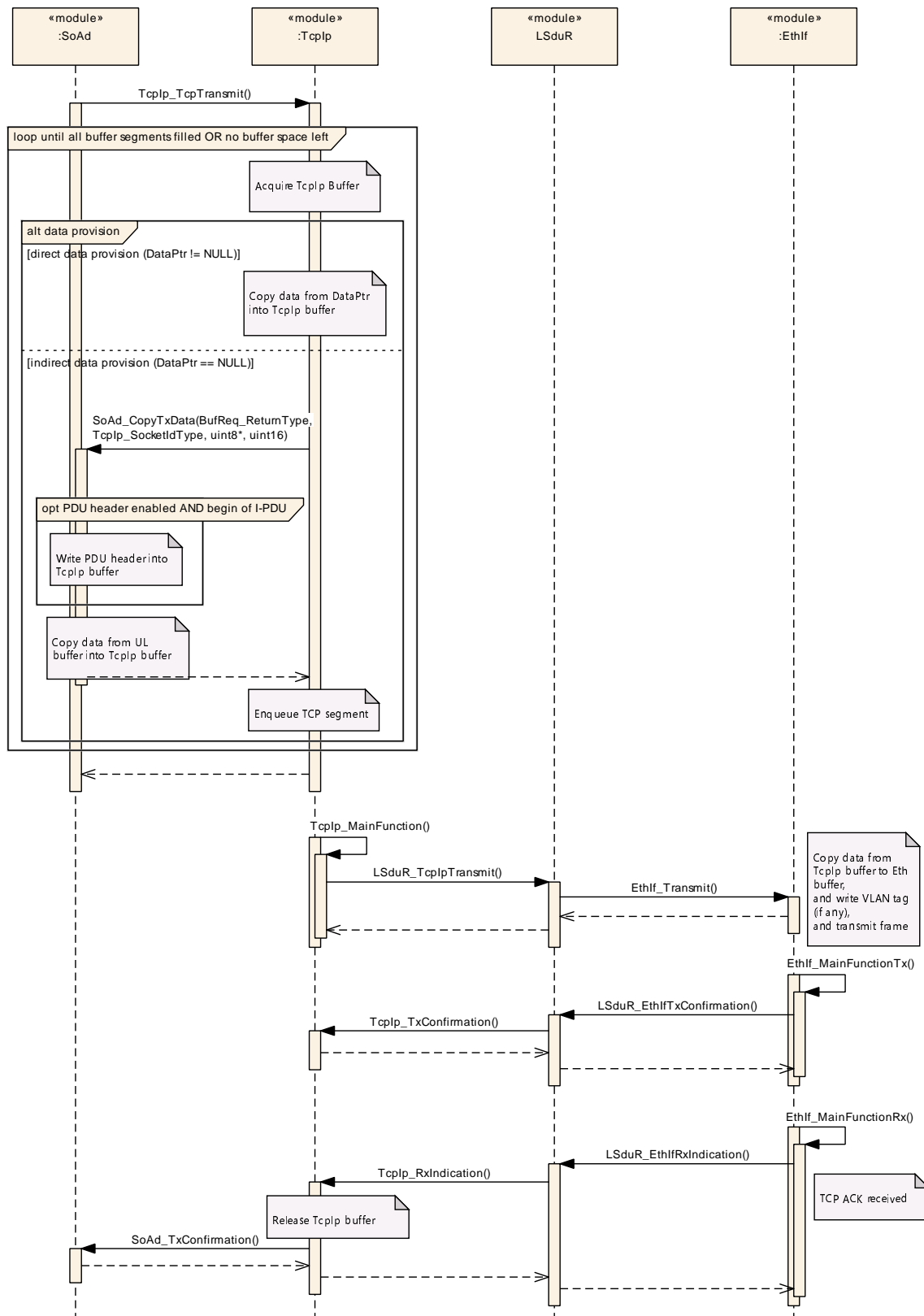


Figure 9.4: TcpIp TCP Tx

## 9.5 Transmission UDP

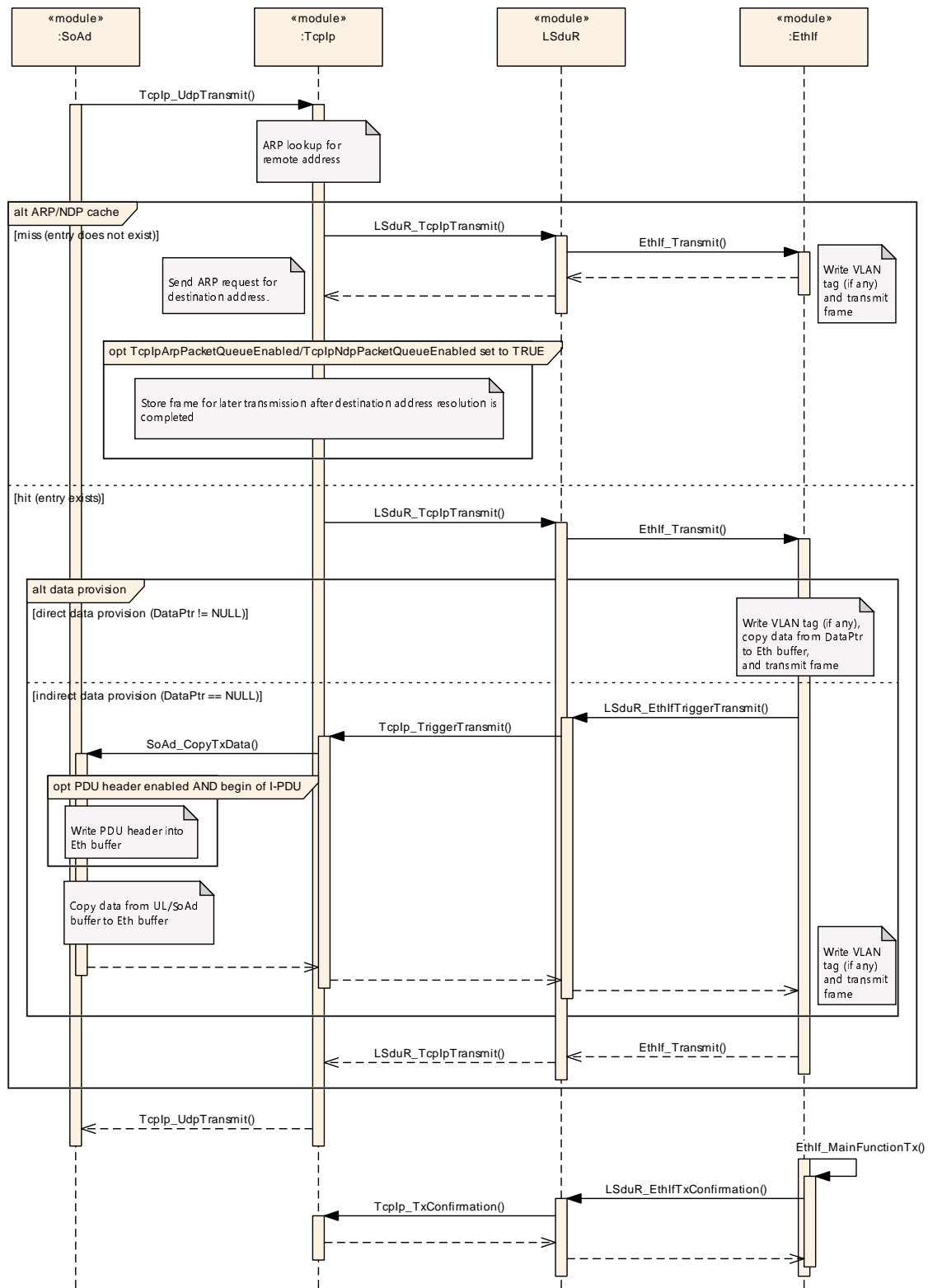
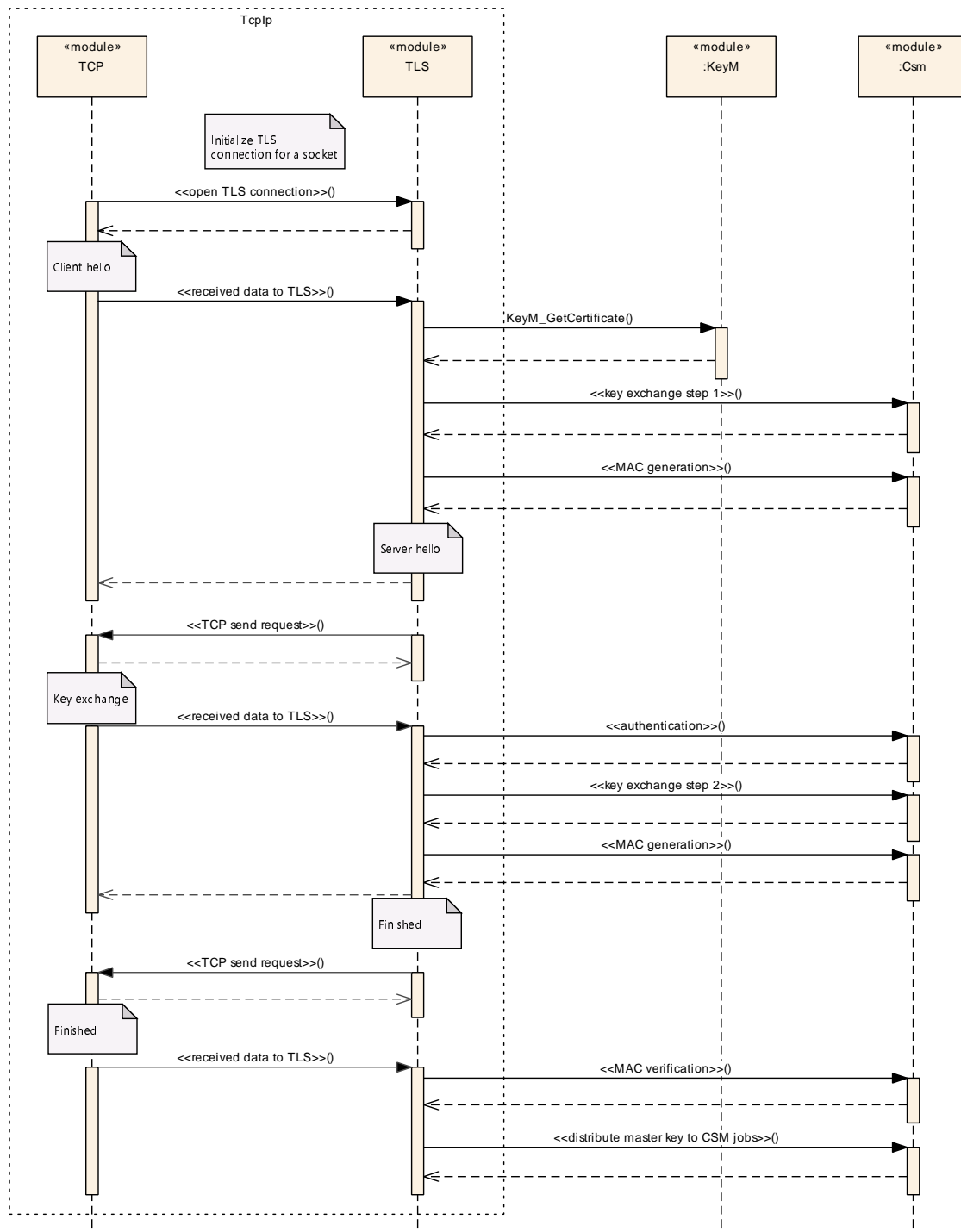


Figure 9.5: TcpIp UDP Tx

## 9.6 Connection setup for a TLS server



**Figure 9.6: TcpIp TLS server communication**



## 9.7 TLS connection assignment to socket

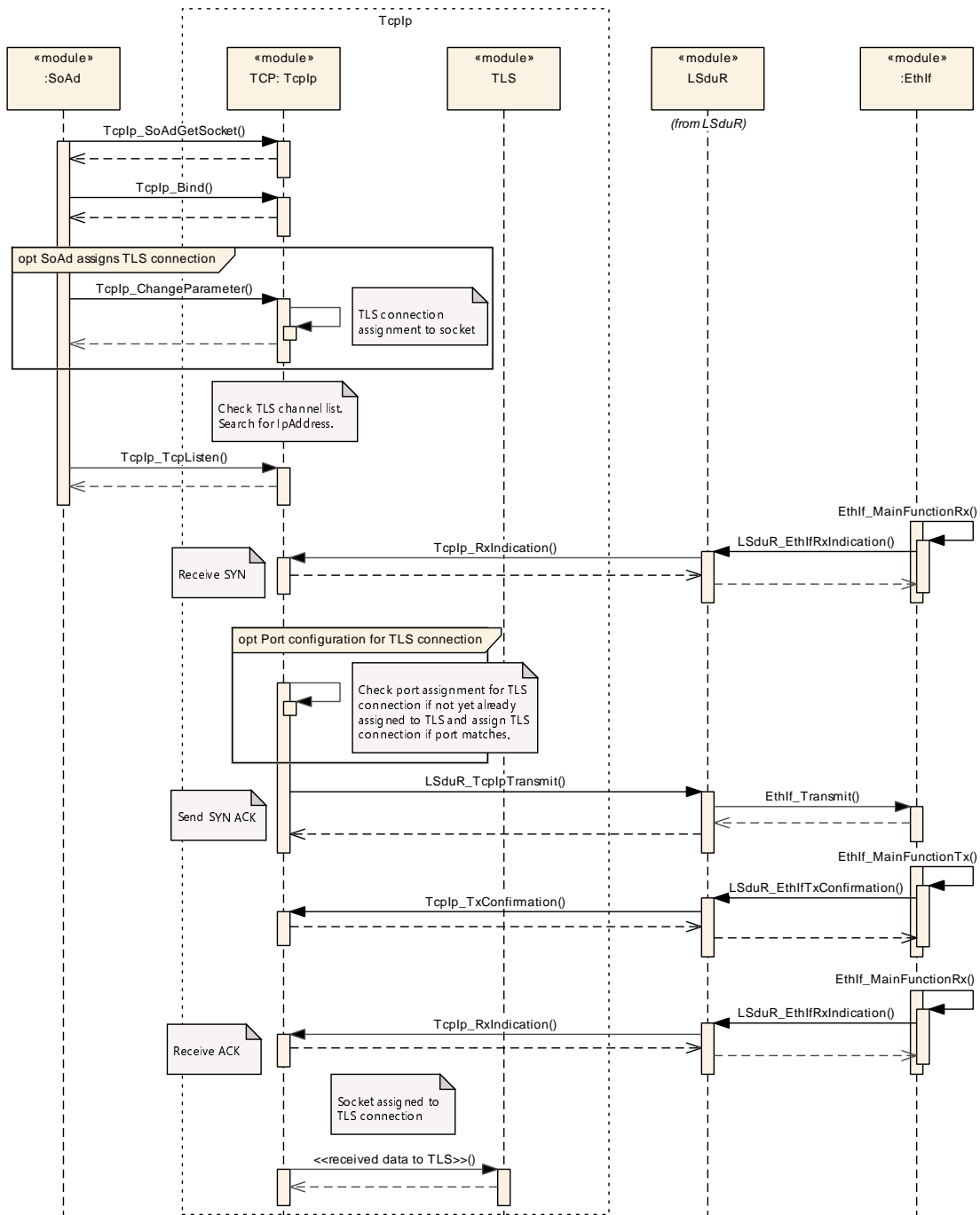


Figure 9.7: TcpIp TLS connection setup server

## 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 Tcp/lp.

Chapter 10.3 specifies published information of the module Tcp/lp.

### 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 Chapter 7 and Chapter 8.

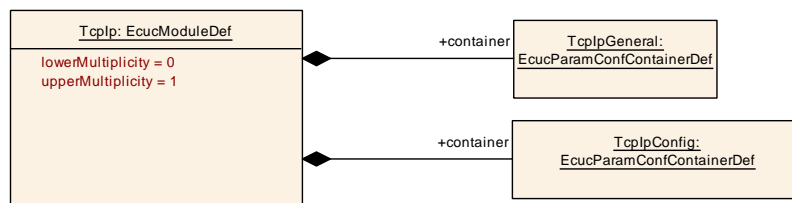


Figure 10.1: TcpIp

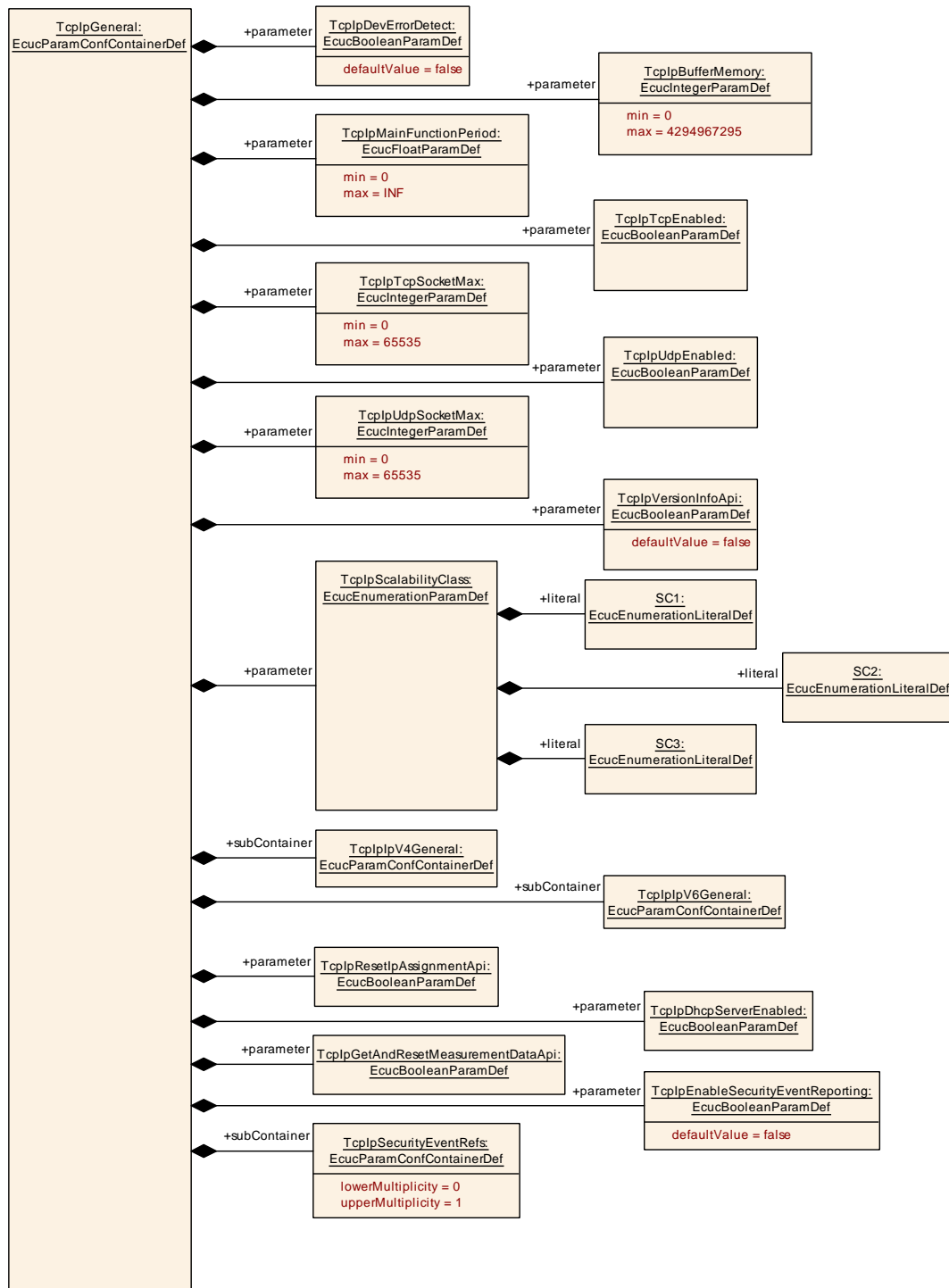
#### 10.2.1 TcpIp

##### [ECUC\_TcpIp\_00001] Definition of EcucModuleDef TcpIp [

Module Name	TcpIp
Description	Configuration of the TcpIp (TCP/IP stack) module.
Post-Build Variant Support	true
Supported Config Variants	VARIANT-LINK-TIME, VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

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

」



**Figure 10.2: TcpIpGeneral**

## 10.2.2 TcpIpGeneral

[ECUC\_TcpIp\_00002] Definition of EcucParamConfContainerDef TcpIpGeneral [

<b>Container Name</b>	TcplpGeneral
<b>Parent Container</b>	<a href="#">Tcplp</a>
<b>Description</b>	This container is a subcontainer of Tcplp and specifies the general configuration parameters of the TCP/IP stack.
<b>Configuration Parameters</b>	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpBufferMemory</a>	1	[ECUC_Tcplp_00016]
<a href="#">TcplpDevErrorDetect</a>	1	[ECUC_Tcplp_00004]
<a href="#">TcplpDhcpServerEnabled</a>	1	[ECUC_Tcplp_00183]
<a href="#">TcplpEnableSecurityEventReporting</a>	1	[ECUC_Tcplp_00319]
<a href="#">TcplpGetAndResetMeasurementDataApi</a>	1	[ECUC_Tcplp_00217]
<a href="#">TcplpMainFunctionPeriod</a>	1	[ECUC_Tcplp_00013]
<a href="#">TcplpResetIpAssignmentApi</a>	1	[ECUC_Tcplp_00182]
<a href="#">TcplpScalabilityClass</a>	1	[ECUC_Tcplp_00169]
<a href="#">TcplpTcpEnabled</a>	1	[ECUC_Tcplp_00008]
<a href="#">TcplpTcpSocketMax</a>	1	[ECUC_Tcplp_00014]
<a href="#">TcplpUdpEnabled</a>	1	[ECUC_Tcplp_00009]
<a href="#">TcplpUdpSocketMax</a>	1	[ECUC_Tcplp_00015]
<a href="#">TcplpVersionInfoApi</a>	1	[ECUC_Tcplp_00005]

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpIPv4General</a>	1	This container is a subcontainer of Tcplp and specifies the general configuration parameters of the TCP/IP stack for IPv4.
<a href="#">TcplpIPv6General</a>	1	This container is a subcontainer of Tcplp and specifies the general configuration parameters of the TCP/IP stack for IPv6.
<a href="#">TcplpSecurityEventRefs</a>	0..1	Container for the references to IdsMEEvent elements representing the security events that the Tcplp module shall report to the IdsM in case the corresponding security related event occurs (and if TcplpEnableSecurityEventReporting is set to "true"). The standardized security events in this container can be extended by vendor-specific security events. <b>Tags:</b> atp.Status=draft

## [ECUC\_Tcplp\_00016] Definition of EcucIntegerParamDef TcplpBufferMemory [

<b>Parameter Name</b>	TcplpBufferMemory	
<b>Parent Container</b>	<a href="#">TcplpGeneral</a>	
<b>Description</b>	Memory size in bytes reserved for TCP/IP buffers.	
<b>Multiplicity</b>	1	
<b>Type</b>	EcucIntegerParamDef	
<b>Range</b>	0 .. 4294967295	
<b>Default value</b>	—	





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		

└

#### [ECUC\_Tcplp\_00004] Definition of EcucBooleanParamDef TcplpDevErrorDetect

└

Parameter Name	TcplpDevErrorDetect		
Parent Container	<a href="#">TcplpGeneral</a>		
Description	Switches the development error detection and notification on or off. <ul style="list-style-type: none"> <li>• true: detection and notification is enabled.</li> <li>• false: detection and notification is disabled.</li> </ul>		
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		

└

#### [ECUC\_Tcplp\_00183] Definition of EcucBooleanParamDef TcplpDhcpServerEnabled

└

Parameter Name	TcplpDhcpServerEnabled		
Parent Container	<a href="#">TcplpGeneral</a>		
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		

└

## [ECUC\_Tcplp\_00319] Definition of EcucBooleanParamDef TcplpEnableSecurityEventReporting

Status: DRAFT

[

Parameter Name	TcplpEnableSecurityEventReporting		
Parent Container	<a href="#">TcplpGeneral</a>		
Description	Switches the reporting of security events to the IdsM: - true: reporting is enabled. - false: reporting is disabled. <b>Tags:</b> atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

]

## [ECUC\_Tcplp\_00217] Definition of EcucBooleanParamDef TcplpGetAndResetMeasurementDataApi

Parameter Name	TcplpGetAndResetMeasurementDataApi		
Parent Container	<a href="#">TcplpGeneral</a>		
Description	Enables / Disables the Get and Reset Measurement Data 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		

]

## [ECUC\_Tcplp\_00013] Definition of EcucFloatParamDef TcplpMainFunctionPeriod

Parameter Name	TcplpMainFunctionPeriod		
Parent Container	<a href="#">TcplpGeneral</a>		
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		

### [ECUC\_Tcplp\_00182] Definition of EcucBooleanParamDef TcplpResetIpAssignmentApi

Parameter Name	TcplpResetIpAssignmentApi		
Parent Container	<a href="#">TcplpGeneral</a>		
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		

### [ECUC\_Tcplp\_00169] Definition of EcucEnumerationParamDef TcplpScalabilityClass

Parameter Name	TcplpScalabilityClass		
Parent Container	<a href="#">TcplpGeneral</a>		
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		



### [ECUC\_Tcplp\_00008] Definition of EcucBooleanParamDef TcplpTcpEnabled [

Parameter Name	TcplpTcpEnabled		
Parent Container	<a href="#">TcplpGeneral</a>		
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		

]

### [ECUC\_Tcplp\_00014] Definition of EcucIntegerParamDef TcplpTcpSocketMax [

Parameter Name	TcplpTcpSocketMax		
Parent Container	<a href="#">TcplpGeneral</a>		
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	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

### [ECUC\_Tcplp\_00009] Definition of EcucBooleanParamDef TcplpUdpEnabled [

Parameter Name	TcplpUdpEnabled		
Parent Container	<a href="#">TcplpGeneral</a>		
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		

]

### [ECUC\_Tcplp\_00015] Definition of EcucIntegerParamDef TcplpUdpSocketMax [

Parameter Name	TcplpUdpSocketMax		
Parent Container	<a href="#">TcplpGeneral</a>		
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	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

### [ECUC\_Tcplp\_00005] Definition of EcucBooleanParamDef TcplpVersionInfoApi [

]

Parameter Name	TcplpVersionInfoApi		
Parent Container	<a href="#">TcplpGeneral</a>		
Description	If true the Tcplp_GetVersionInfo API is available.		
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		

]

## 10.2.3 TcplpV4General

### [ECUC\_Tcplp\_00163] Definition of EcucParamConfContainerDef TcplpV4General [

Container Name	TcplpV4General
Parent Container	<a href="#">TcplpGeneral</a>
Description	This container is a subcontainer of Tcplp and specifies the general configuration parameters of the TCP/IP stack for IPv4
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpArpEnabled</a>	1	[ECUC_Tcplp_00006]
<a href="#">TcplpAutolpEnabled</a>	1	[ECUC_Tcplp_00011]
<a href="#">TcplpDhcpClientEnabled</a>	1	[ECUC_Tcplp_00010]
<a href="#">TcplpIcmpEnabled</a>	1	[ECUC_Tcplp_00007]
<a href="#">TcplpIPv4Enabled</a>	1	[ECUC_Tcplp_00088]
<a href="#">TcplpLocalAddrIpv4EntriesMax</a>	1	[ECUC_Tcplp_00018]
<a href="#">TcplpPathMtuDiscoveryEnabled</a>	1	[ECUC_Tcplp_00012]

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

]

### [ECUC\_Tcplp\_00006] Definition of EcucBooleanParamDef TcplpArpEnabled [

Parameter Name	TcplpArpEnabled		
Parent Container	<a href="#">TcplpV4General</a>		
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	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

### [ECUC\_Tcplp\_00011] Definition of EcucBooleanParamDef TcplpAutolpEnabled [

Parameter Name	TcplpAutolpEnabled		
Parent Container	<a href="#">TcplpV4General</a>		
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	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

## [ECUC\_Tcplp\_00010] Definition of EcucBooleanParamDef TcplpDhcpClientEnabled

Parameter Name	TcplpDhcpClientEnabled		
Parent Container	<a href="#">TcplpV4General</a>		
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	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

## [ECUC\_Tcplp\_00007] Definition of EcucBooleanParamDef TcplpIcmpEnabled

Parameter Name	TcplpIcmpEnabled		
Parent Container	<a href="#">TcplpV4General</a>		
Description	Enables (TRUE) or disables (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		

## [ECUC\_Tcplp\_00088] Definition of EcucBooleanParamDef TcplpV4Enabled

Parameter Name	TcplpV4Enabled		
Parent Container	<a href="#">TcplpV4General</a>		
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
--------------------	--------------

### [ECUC\_Tcplp\_00018] Definition of EcucIntegerParamDef TcplpLocalAddrIpv4EntriesMax

Parameter Name	TcplpLocalAddrIpv4EntriesMax		
Parent Container	<a href="#">TcplpV4General</a>		
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	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

### [ECUC\_Tcplp\_00012] Definition of EcucBooleanParamDef TcplpPathMtuDiscoveryEnabled

Parameter Name	TcplpPathMtuDiscoveryEnabled		
Parent Container	<a href="#">TcplpV4General</a>		
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		

## 10.2.4 TcplpV6General

### [ECUC\_Tcplp\_00164] Definition of EcucParamConfContainerDef TcplpV6General

<b>Container Name</b>	TcplpV6General
<b>Parent Container</b>	<a href="#">TcplpGeneral</a>
<b>Description</b>	This container is a subcontainer of Tcplp and specifies the general configuration parameters of the TCP/IP stack for IPv6.
<b>Configuration Parameters</b>	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpDhcpV6ClientEnabled</a>	1	[ECUC_Tcplp_00093]
<a href="#">TcplpV6Enabled</a>	1	[ECUC_Tcplp_00089]
<a href="#">TcplpV6PathMtuDiscoveryEnabled</a>	1	[ECUC_Tcplp_00090]
<a href="#">TcplpLocalAddrIpv6EntriesMax</a>	1	[ECUC_Tcplp_00017]
<a href="#">TcplpNdpAddressResolutionUnreachabilityDetectionEnabled</a>	1	[ECUC_Tcplp_00091]
<a href="#">TcplpNdpPrefixAndRouterDiscoveryEnabled</a>	1	[ECUC_Tcplp_00092]

<b>No Included Containers</b>
-------------------------------

## [ECUC\_Tcplp\_00093] Definition of EcucBooleanParamDef TcplpDhcpV6Client Enabled [

<b>Parameter Name</b>	TcplpDhcpV6ClientEnabled		
<b>Parent Container</b>	<a href="#">TcplpV6General</a>		
<b>Description</b>	Enables (TRUE) or disables (FALSE) the DHCPv6 (Dynamic Host Configuration Protocol for IPv6) Client.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	–		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

## [ECUC\_Tcplp\_00089] Definition of EcucBooleanParamDef TcplpV6Enabled [

<b>Parameter Name</b>	TcplpV6Enabled
<b>Parent Container</b>	<a href="#">TcplpV6General</a>
<b>Description</b>	Enables (TRUE) or disables (FALSE) support of IPv6 (Internet Protocol version 6).
<b>Multiplicity</b>	1
<b>Type</b>	EcucBooleanParamDef
<b>Default value</b>	–





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

### [ECUC\_Tcplp\_00090] Definition of EcucBooleanParamDef TcplpV6PathMtuDiscoveryEnabled

Parameter Name	TcplpV6PathMtuDiscoveryEnabled		
Parent Container	<a href="#">TcplpV6General</a>		
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		

### [ECUC\_Tcplp\_00017] Definition of EcucIntegerParamDef TcplpLocalAddrIpv6EntriesMax

Parameter Name	TcplpLocalAddrIpv6EntriesMax		
Parent Container	<a href="#">TcplpV6General</a>		
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	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

## [ECUC\_Tcplp\_00091] Definition of EcucBooleanParamDef TcplpNdpAddressResolutionUnreachabilityDetectionEnabled [

Parameter Name	TcplpNdpAddressResolutionUnreachabilityDetectionEnabled		
Parent Container	<a href="#">TcplpV6General</a>		
Description	Enables (TRUE) or disables (FALSE) support of Address Resolution and Neighbor Unreachability Detection 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		

]

## [ECUC\_Tcplp\_00092] Definition of EcucBooleanParamDef TcplpNdpPrefixAndRouterDiscoveryEnabled [

Parameter Name	TcplpNdpPrefixAndRouterDiscoveryEnabled		
Parent Container	<a href="#">TcplpV6General</a>		
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		

]



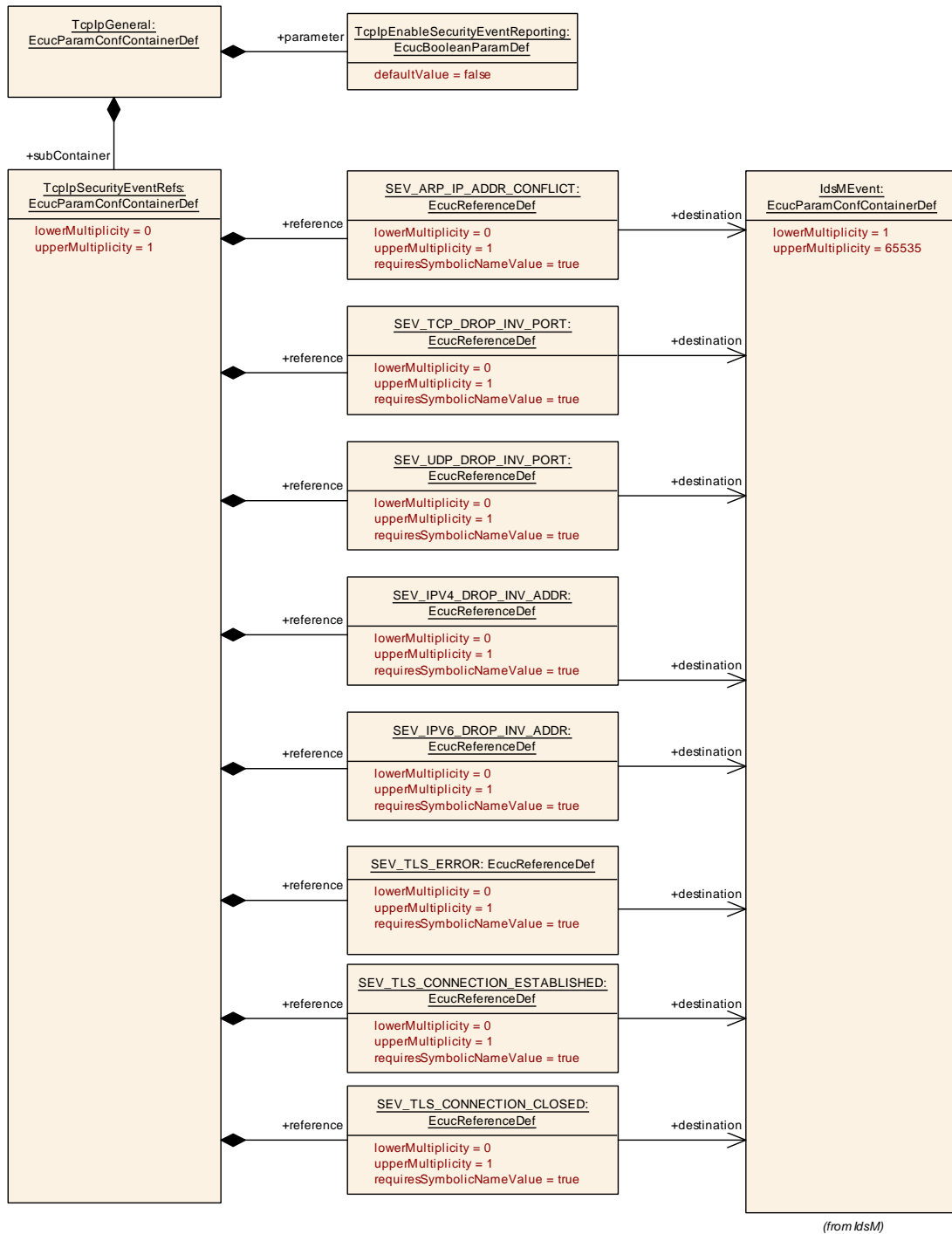


Figure 10.3: TcplpSecurityEventRefs

## 10.2.5 TcplpSecurityEventRefs

### [ECUC\_Tcplp\_00320] Definition of EcucParamConfContainerDef TcplpSecurityEventRefs

Status: DRAFT

[

Container Name	TcplpSecurityEventRefs		
Parent Container	<a href="#">TcplpGeneral</a>		
Description	<p>Container for the references to IdsMEvent elements representing the security events that the Tcplp module shall report to the IdsM in case the corresponding security related event occurs (and if TcplpEnableSecurityEventReporting is set to "true"). The standardized security events in this container can be extended by vendor-specific security events.</p> <p><b>Tags:</b> atp.Status=draft</p>		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">SEV_ARP_IP_ADDR_CONFLICT</a>	0..1	<a href="#">[ECUC_Tcplp_00321]</a>
<a href="#">SEV_IPV4_DROP_INV_ADDR</a>	0..1	<a href="#">[ECUC_Tcplp_00324]</a>
<a href="#">SEV_IPV6_DROP_INV_ADDR</a>	0..1	<a href="#">[ECUC_Tcplp_00325]</a>
<a href="#">SEV_TCP_DROP_INV_PORT</a>	0..1	<a href="#">[ECUC_Tcplp_00322]</a>
<a href="#">SEV_TLS_CONNECTION_CLOSED</a>	0..1	<a href="#">[ECUC_Tcplp_00339]</a>
<a href="#">SEV_TLS_CONNECTION_ESTABLISHED</a>	0..1	<a href="#">[ECUC_Tcplp_00338]</a>
<a href="#">SEV_TLS_ERROR</a>	0..1	<a href="#">[ECUC_Tcplp_00337]</a>
<a href="#">SEV_UDP_DROP_INV_PORT</a>	0..1	<a href="#">[ECUC_Tcplp_00323]</a>

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

]

## [ECUC\_Tcplp\_00321] Definition of EcucReferenceDef SEV\_ARP\_IP\_ADDR\_CONFLICT

Status: DRAFT

[

Parameter Name	SEV_ARP_IP_ADDR_CONFLICT		
Parent Container	<a href="#">TcplpSecurityEventRefs</a>		
Description	Received local IP address in ARP reply for different MAC. <b>Tags:</b> atp.Status=draft		
Multiplicity	0..1		
Type	Symbolic name reference to IdsMEvent		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

## [ECUC\_Tcplp\_00324] Definition of EcucReferenceDef SEV\_IPV4\_DROP\_INV\_ADDR

Status: DRAFT

[

Parameter Name	SEV_IPV4_DROP_INV_ADDR		
Parent Container	<a href="#">TcplpSecurityEventRefs</a>		
Description	Dropped datagram because of invalid IPv4 address. <b>Tags:</b> atp.Status=draft		
Multiplicity	0..1		
Type	Symbolic name reference to IdsMEvent		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

## [ECUC\_Tcplp\_00325] Definition of EcucReferenceDef SEV\_IPV6\_DROP\_INV\_ADDR

Status: DRAFT

[

Parameter Name	SEV_IPV6_DROP_INV_ADDR		
Parent Container	<a href="#">TcplpSecurityEventRefs</a>		
Description	Dropped datagram because of invalid IPV6 address. <b>Tags:</b> atp.Status=draft		
Multiplicity	0..1		
Type	Symbolic name reference to IdsMEvent		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

## [ECUC\_Tcplp\_00322] Definition of EcucReferenceDef SEV\_TCP\_DROP\_INV\_PORT

Status: DRAFT

[

Parameter Name	SEV_TCP_DROP_INV_PORT		
Parent Container	<a href="#">TcplpSecurityEventRefs</a>		
Description	Dropped TCP packet because of invalid destination TCP-Port. <b>Tags:</b> atp.Status=draft		
Multiplicity	0..1		
Type	Symbolic name reference to IdsMEvent		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

## [ECUC\_Tcplp\_00339] Definition of EcucReferenceDef SEV\_TLS\_CONNECTION\_CLOSED

Status: DRAFT

[

Parameter Name	SEV_TLS_CONNECTION_CLOSED		
Parent Container	<a href="#">TcplpSecurityEventRefs</a>		
Description	A TLS connection was closed normally. <b>Tags:</b> atp.Status=draft		
Multiplicity	0..1		
Type	Symbolic name reference to IdsMEvent		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

## [ECUC\_Tcplp\_00338] Definition of EcucReferenceDef SEV\_TLS\_CONNECTION\_ESTABLISHED

Status: DRAFT

[

Parameter Name	SEV_TLS_CONNECTION_ESTABLISHED		
Parent Container	<a href="#">TcplpSecurityEventRefs</a>		
Description	A TLS connection was successfully established. <b>Tags:</b> atp.Status=draft		
Multiplicity	0..1		
Type	Symbolic name reference to IdsMEvent		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

## [ECUC\_Tcplp\_00337] Definition of EcucReferenceDef SEV\_TLS\_ERROR

Status: DRAFT

[

Parameter Name	SEV_TLS_ERROR		
Parent Container	<a href="#">TcplpSecurityEventRefs</a>		
Description	An alert message was detected (either received or generated) by TLS. <b>Tags:</b> atp.Status=draft		
Multiplicity	0..1		
Type	Symbolic name reference to IdsMEvent		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

## [ECUC\_Tcplp\_00323] Definition of EcucReferenceDef SEV\_UDP\_DROP\_INV\_PORT

Status: DRAFT

[

Parameter Name	SEV_UDP_DROP_INV_PORT		
Parent Container	<a href="#">TcplpSecurityEventRefs</a>		
Description	Dropped UDP packet because of invalid destination UDP-Port. <b>Tags:</b> atp.Status=draft		
Multiplicity	0..1		
Type	Symbolic name reference to IdsMEvent		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

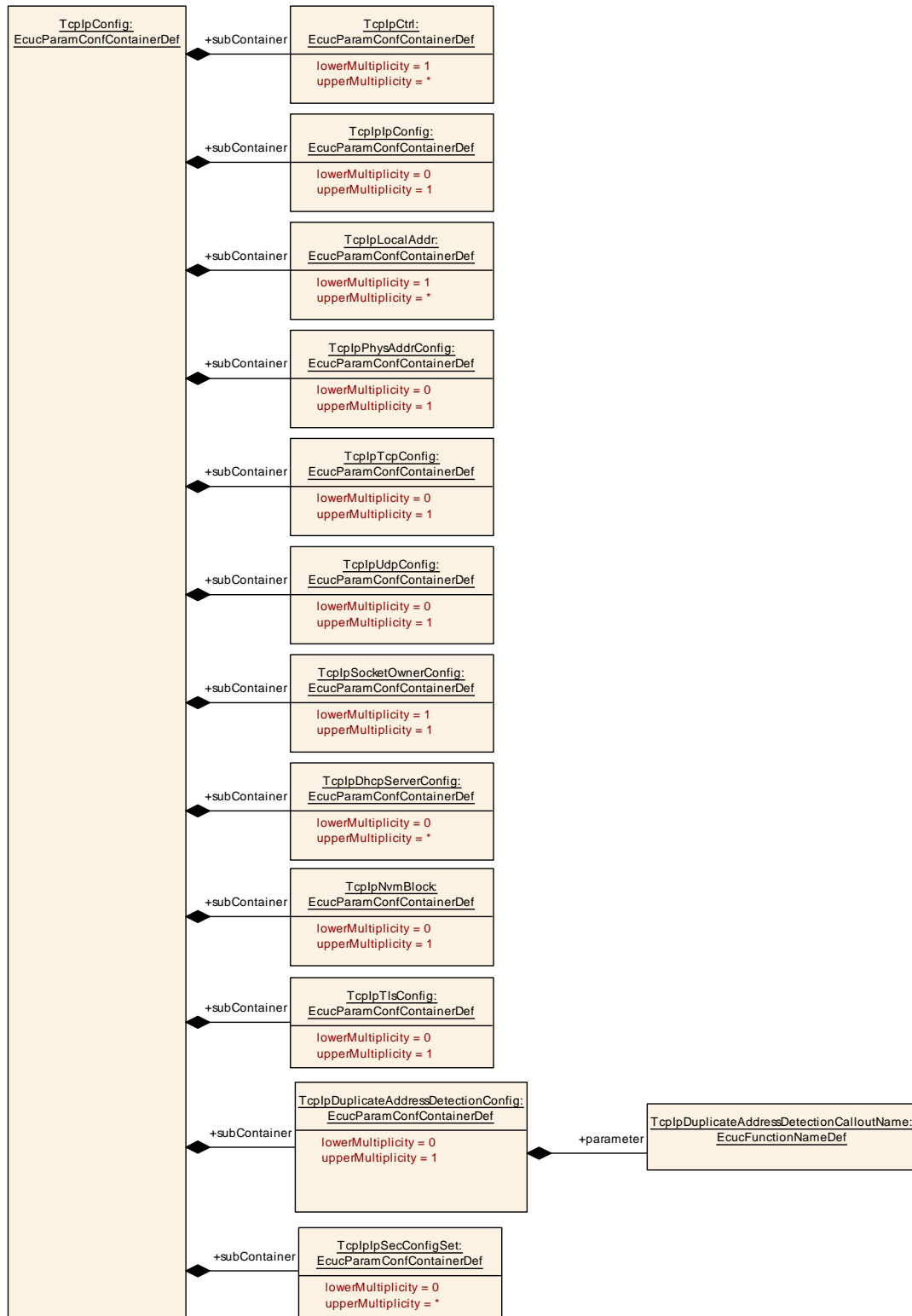


Figure 10.4: TcpIpConfig

## 10.2.6 TcplpConfig

### [ECUC\_Tcplp\_00003] Definition of EcucParamConfContainerDef TcplpConfig [

<b>Container Name</b>	TcplpConfig
<b>Parent Container</b>	<a href="#">Tcplp</a>
<b>Description</b>	This container contains the configuration parameters and sub containers of the AUTOSAR Tcplp module.
<b>Configuration Parameters</b>	

<b>No Included Parameters</b>
-------------------------------

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpCtrl</a>	1..*	Specifies the EthIf controller used for IP communication.
<a href="#">TcplpDhcpServerConfig</a>	0..*	Specifies the configuration parameters of the DHCP Server sub-module.
<a href="#">TcplpDuplicateAddressDetection Config</a>	0..1	Specifies the DAD callout function.
<a href="#">TcplpIpConfig</a>	0..1	Specifies the configuration parameters of the IP (Internet Protocol) sub-module
<a href="#">TcplpIpSecConfigSet</a>	0..*	Specifies the IPsec configuration.
<a href="#">TcplpLocalAddr</a>	1..*	Specifies the local IP (Internet Protocol) addresses used for IP communication.
<a href="#">TcplpNvmBlock</a>	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).
<a href="#">TcplpPhysAddrConfig</a>	0..1	Specifies the physical address configuration.
<a href="#">TcplpSocketOwnerConfig</a>	1	Specifies the upper layer modules of Tcplp using the socket API.
<a href="#">TcplpTcpConfig</a>	0..1	Specifies the configuration parameters of the TCP (Transmission Control Protocol) sub-module.
<a href="#">TcplpTlsConfig</a>	0..1	Specifies the configuration parameters of the TLS (Transport Layer Security) sub module.
<a href="#">TcplpUdpConfig</a>	0..1	Specifies the configuration parameters of the UDP (User Datagram Protocol) sub-module

]



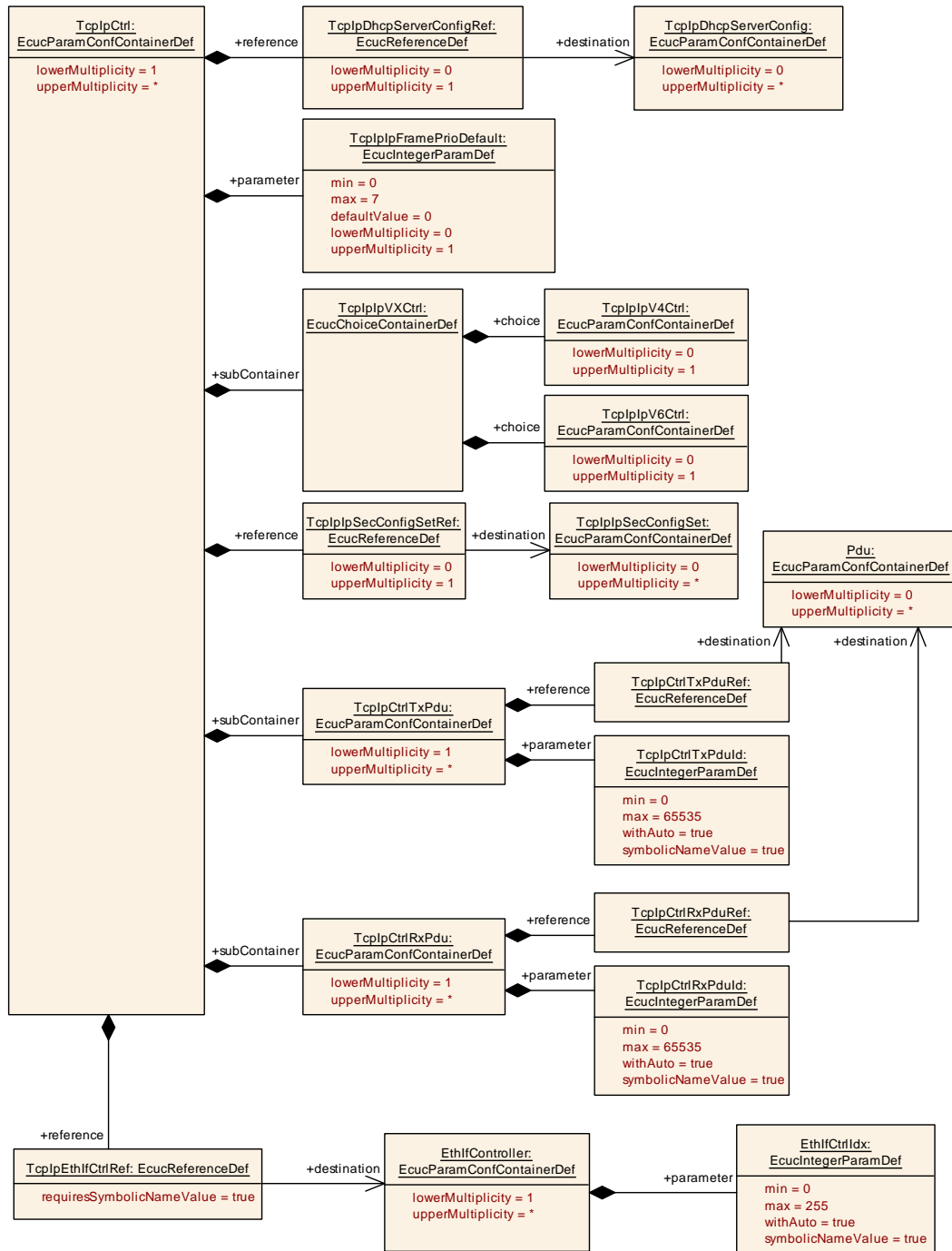


Figure 10.5: TcpIpCtrl

## 10.2.7 TcpIpCtrl

[ECUC\_TcpIp\_00021] Definition of EcucParamConfContainerDef TcpIpCtrl [

<b>Container Name</b>	TcplpCtrl
<b>Parent Container</b>	<a href="#">TcplpConfig</a>
<b>Description</b>	Specifies the EthIf controller used for IP communication.
<b>Configuration Parameters</b>	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpFramePrioDefault</a>	0..1	[ECUC_Tcplp_00081]
<a href="#">TcplpDhcpServerConfigRef</a>	0..1	[ECUC_Tcplp_00195]
<a href="#">TcplpEthIfCtrlRef</a>	1	[ECUC_Tcplp_00041]
<a href="#">TcplpSecConfigSetRef</a>	0..1	[ECUC_Tcplp_00315]

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpCtrlRxPdu</a>	1..*	PDU used for reception of Ethernet frames. Supported MetaDataItemTypes: <ul style="list-style-type: none"> <li>• ETHERNET_MAC_64</li> <li>• BROADCAST_8</li> </ul> <b>Tags:</b> atp.Status=draft
<a href="#">TcplpCtrlTxPdu</a>	1..*	PDU used for transmission of Ethernet frames. Supported MetaDataItemTypes: <ul style="list-style-type: none"> <li>• ETHERNET_MAC_64</li> <li>• PRIORITY_8</li> <li>• LISTELEM_PTR</li> </ul> <b>Tags:</b> atp.Status=draft
<a href="#">TcplpVXCtrl</a>	1	Specifies whether this controller is an Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) instance.

## [ECUC\_Tcplp\_00081] Definition of EcucIntegerParamDef TcplpFramePrioDefault

<b>Parameter Name</b>	TcplpFramePrioDefault		
<b>Parent Container</b>	<a href="#">TcplpCtrl</a>		
<b>Description</b>	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.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 7		
<b>Default value</b>	0		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	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		

### [ECUC\_Tcplp\_00195] Definition of EcucReferenceDef TcplpDhcpServerConfig Ref [

Parameter Name	TcplpDhcpServerConfigRef		
Parent Container	<a href="#">TcplpCtrl</a>		
Description	Reference to a TcplpDhcpServerConfig which shall be used for this controller setting (VLAN).		
Multiplicity	0..1		
Type	Reference to <a href="#">TcplpDhcpServerConfig</a>		
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		

### [ECUC\_Tcplp\_00041] Definition of EcucReferenceDef TcplpEthIfCtrlRef [

Parameter Name	TcplpEthIfCtrlRef		
Parent Container	<a href="#">TcplpCtrl</a>		
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		

### [ECUC\_Tcplp\_00315] Definition of EcucReferenceDef TcplpIpSecConfigSetRef [

Parameter Name	TcplpIpSecConfigSetRef		
Parent Container	<a href="#">TcplpCtrl</a>		
Description	Reference to set of SDP entries which shall be used for IPsec.		





<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to <a href="#">TcpIpSecConfigSet</a>		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

]

## 10.2.8 TcpIpCtrlRxPdu

### [ECUC\_TcpIp\_00344] Definition of EcucParamConfContainerDef TcpIpCtrlRxPdu

Status: DRAFT

[

<b>Container Name</b>	TcpIpCtrlRxPdu		
<b>Parent Container</b>	<a href="#">TcpIpCtrl</a>		
<b>Description</b>	PDU used for reception of Ethernet frames. Supported MetaDataItemTypes: <ul style="list-style-type: none"> <li>ETHERNET_MAC_64</li> <li>BROADCAST_8</li> </ul> <b>Tags:</b> atp.Status=draft		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Configuration Parameters</b>			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcpIpCtrlRxPduId</a>	1	[ECUC_TcpIp_00345]
<a href="#">TcpIpCtrlRxPduRef</a>	1	[ECUC_TcpIp_00346]

<b>No Included Containers</b>
-------------------------------

]

## [ECUC\_Tcplp\_00345] Definition of EcucIntegerParamDef TcplpCtrlRxPduId

Status: DRAFT

[

Parameter Name	TcplpCtrlRxPduId		
Parent Container	<a href="#">TcplpCtrlRxPdu</a>		
Description	The PDU identifier used for RxIndication from LSduR. <b>Tags:</b> atp.Status=draft		
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 withAuto = true		

]

## [ECUC\_Tcplp\_00346] Definition of EcucReferenceDef TcplpCtrlRxPduRef

Status: DRAFT

[

Parameter Name	TcplpCtrlRxPduRef		
Parent Container	<a href="#">TcplpCtrlRxPdu</a>		
Description	Reference to the global PDU. <b>Tags:</b> atp.Status=draft		
Multiplicity	1		
Type	Reference to Pdu		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

]

## 10.2.9 TcplpCtrlTxPdu

## [ECUC\_Tcplp\_00341] Definition of EcucParamConfContainerDef TcplpCtrlTxPdu

Status: DRAFT

[

Container Name	TcplpCtrlTxPdu		
Parent Container	<a href="#">TcplpCtrl</a>		
Description	PDU used for transmission of Ethernet frames. Supported MetaDataItemTypes: <ul style="list-style-type: none"> <li>• ETHERNET_MAC_64</li> <li>• PRIORITY_8</li> <li>• LISTELEM_PTR</li> </ul> <b>Tags:</b> atp.Status=draft		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpCtrlTxPduld</a>	1	[ <a href="#">ECUC_Tcplp_00342</a> ]
<a href="#">TcplpCtrlTxPduRef</a>	1	[ <a href="#">ECUC_Tcplp_00343</a> ]

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

## [ECUC\_Tcplp\_00342] Definition of EcucIntegerParamDef TcplpCtrlTxPduld

Status: DRAFT

Parameter Name	TcplpCtrlTxPduld		
Parent Container	<a href="#">TcplpCtrlTxPdu</a>		
Description	The PDU identifier used for TxConfirmation from LSduR. <b>Tags:</b> atp.Status=draft		
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 withAuto = true		

**[ECUC\_Tcplp\_00343] Definition of EcucReferenceDef TcplpCtrlTxPduRef***Status:* DRAFT

[

<b>Parameter Name</b>	TcplpCtrlTxPduRef		
<b>Parent Container</b>	<a href="#">TcplpCtrlTxPdu</a>		
<b>Description</b>	Reference to the global PDU. <b>Tags:</b> atp.Status=draft		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to Pdu		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: ECU		

]

**10.2.10 TcplpVXCtrl****[ECUC\_Tcplp\_00094] Definition of EcucChoiceContainerDef TcplpVXCtrl** [

<b>Choice Container Name</b>	TcplpVXCtrl
<b>Parent Container</b>	<a href="#">TcplpCtrl</a>
<b>Description</b>	Specifies whether this controller is an Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) instance.

<b>No Included Parameters</b>
-------------------------------

Container Choices		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpV4Ctrl</a>	0..1	Specifies an Internet Protocol version 4 (IPv4) instance.
<a href="#">TcplpV6Ctrl</a>	0..1	Specifies an Internet Protocol version 6 (IPv6) instance.

]

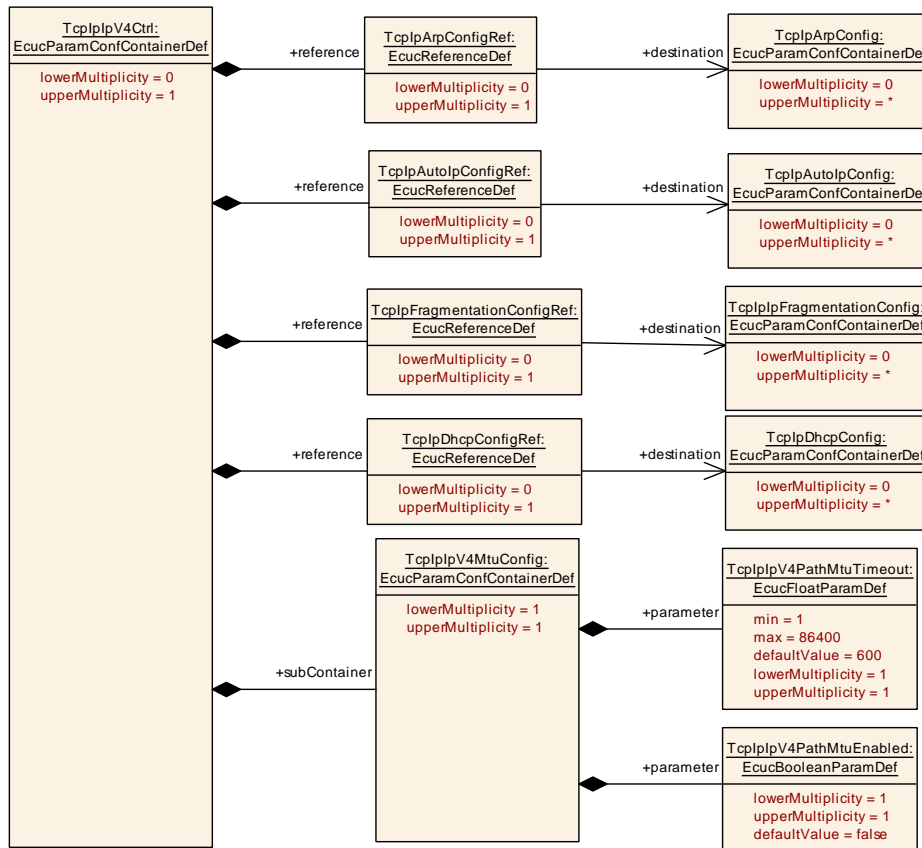


Figure 10.6: TcpIpV4Ctrl

### 10.2.11 TcpIpV4Ctrl

[ECUC\_TcpIp\_00166] Definition of EcucParamConfContainerDef TcpIpV4Ctrl [

Container Name	TcpIpV4Ctrl
Parent Container	<a href="#">TcpIpVXCtrl</a>
Description	Specifies an Internet Protocol version 4 (IPv4) instance.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcpIpArpConfigRef</a>	0..1	[ECUC_TcpIp_00097]
<a href="#">TcpIpAutopConfigRef</a>	0..1	[ECUC_TcpIp_00098]
<a href="#">TcpIpDhcpConfigRef</a>	0..1	[ECUC_TcpIp_00100]
<a href="#">TcpIpFragmentationConfigRef</a>	0..1	[ECUC_TcpIp_00099]

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcpIpV4MtuConfig</a>	1	This container specifies the Maximum Transmission Unit parameters for this IPv4 instance.



## [ECUC\_Tcplp\_00097] Definition of EcucReferenceDef TcplpArpConfigRef [

Parameter Name	TcplpArpConfigRef		
Parent Container	<a href="#">TcplpV4Ctrl</a>		
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 <a href="#">TcplpArpConfig</a>		
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		

## [ECUC\_Tcplp\_00098] Definition of EcucReferenceDef TcplpAutolpConfigRef [

Parameter Name	TcplpAutolpConfigRef		
Parent Container	<a href="#">TcplpV4Ctrl</a>		
Description	Reference to Autolp configuration for this IPv4 instance. (Multiple IPv4 instances may use the same configuration container but will operate independently)		
Multiplicity	0..1		
Type	Reference to <a href="#">TcplpAutolpConfig</a>		
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		

## [ECUC\_Tcplp\_00100] Definition of EcucReferenceDef TcplpDhcpConfigRef [

Parameter Name	TcplpDhcpConfigRef		
Parent Container	<a href="#">TcplpV4Ctrl</a>		
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 <a href="#">TcplpDhcpConfig</a>		
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		

]

## [ECUC\_Tcplp\_00099] Definition of EcucReferenceDef TcplpFragmentationConfigRef [

Parameter Name	TcplpFragmentationConfigRef		
Parent Container	<a href="#">TcplpV4Ctrl</a>		
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 <a href="#">TcplpFragmentationConfig</a>		
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		

]

### 10.2.12 TcplpV4MtuConfig

## [ECUC\_Tcplp\_00209] Definition of EcucParamConfContainerDef TcplpV4MtuConfig [

Container Name	TcpIplpV4MtuConfig
Parent Container	<a href="#">TcpIplpV4Ctrl</a>
Description	This container specifies the Maximum Transmission Unit parameters for this IPv4 instance.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcpIplpV4PathMtuEnabled</a>	1	[ECUC_TcpIp_00211]
<a href="#">TcpIplpV4PathMtuTimeout</a>	1	[ECUC_TcpIp_00210]

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

## [ECUC\_TcpIp\_00211] Definition of EcucBooleanParamDef TcpIplpV4PathMtuEnabled

Parameter Name	TcpIplpV4PathMtuEnabled		
Parent Container	<a href="#">TcpIplpV4MtuConfig</a>		
Description	If enabled the IPv4 processes incoming ICMPv4 "Packet Too Big" messages and stores a MTU value for each destination address.		
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		

## [ECUC\_TcpIp\_00210] Definition of EcucFloatParamDef TcpIplpV4PathMtuTimeout

Parameter Name	TcpIplpV4PathMtuTimeout		
Parent Container	<a href="#">TcpIplpV4MtuConfig</a>		
Description	If this value is >0 the IpV4 will reset the MTU value stored for each destination after n seconds. see [RFC1191 6.3. 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		

└

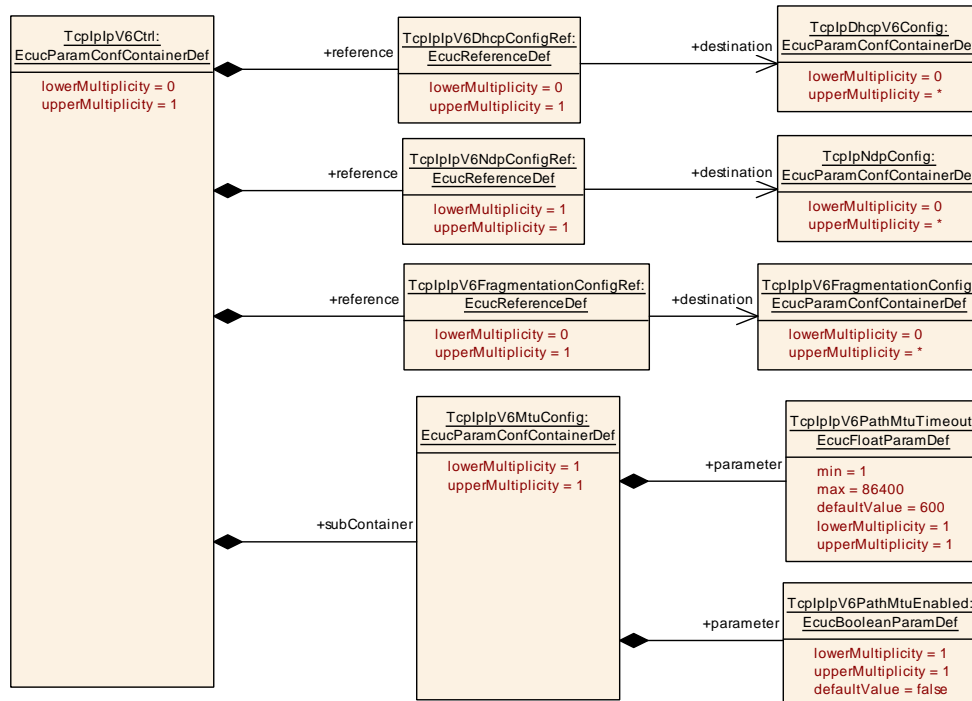


Figure 10.7: TcpIpV6Ctrl

### 10.2.13 TcpIpV6Ctrl

#### [ECUC\_TcpIp\_00096] Definition of EcucParamConfContainerDef TcpIpV6Ctrl

Container Name	TcpIpV6Ctrl
Parent Container	<a href="#">TcpIpVXCtrl</a>
Description	Specifies an Internet Protocol version 6 (IPv6) instance.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcpIpV6DhcpConfigRef</a>	0..1	[ECUC_TcpIp_00101]
<a href="#">TcpIpV6FragmentationConfigRef</a>	0..1	[ECUC_TcpIp_00103]
<a href="#">TcpIpV6NdpConfigRef</a>	1	[ECUC_TcpIp_00102]

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpV6MtuConfig</a>	1	This container specifies the Maximum Transmission Unit parameters for this IPv6 instance.

## [ECUC\_Tcplp\_00101] Definition of EcucReferenceDef TcplpV6DhcpConfigRef

Parameter Name	TcplpV6DhcpConfigRef		
Parent Container	<a href="#">TcplpV6Ctrl</a>		
Description	Reference to DHCPv6 configuration. (Multiple IPv6 instances may use the same configuration container but will operate independently)		
Multiplicity	0..1		
Type	Reference to <a href="#">TcplpDhcpV6Config</a>		
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		

## [ECUC\_Tcplp\_00103] Definition of EcucReferenceDef TcplpV6FragmentationConfigRef

Parameter Name	TcplpV6FragmentationConfigRef		
Parent Container	<a href="#">TcplpV6Ctrl</a>		
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 <a href="#">TcplpV6FragmentationConfig</a>		
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		

**[ECUC\_Tcplp\_00102] Definition of EcucReferenceDef TcplpV6NdpConfigRef** [

Parameter Name	TcplpV6NdpConfigRef		
Parent Container	<a href="#">TcplpV6Ctrl</a>		
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 <a href="#">TcplpNdpConfig</a>		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

**10.2.14 TcplpV6MtuConfig****[ECUC\_Tcplp\_00104] Definition of EcucParamConfContainerDef TcplpV6MtuConfig** [

Container Name	TcplpV6MtuConfig
Parent Container	<a href="#">TcplpV6Ctrl</a>
Description	This container specifies the Maximum Transmission Unit parameters for this IPv6 instance.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpV6PathMtuEnabled</a>	1	[ECUC_Tcplp_00107]
<a href="#">TcplpV6PathMtuTimeout</a>	1	[ECUC_Tcplp_00105]

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

]

**[ECUC\_Tcplp\_00107] Definition of EcucBooleanParamDef TcplpV6PathMtuEnabled** [

Parameter Name	TcplpV6PathMtuEnabled		
Parent Container	<a href="#">TcplpV6MtuConfig</a>		
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.		





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

]

### [ECUC\_Tcplp\_00105] Definition of EcucFloatParamDef TcplpV6PathMtuTimeout [

<b>Parameter Name</b>	TcplpV6PathMtuTimeout		
<b>Parent Container</b>	<a href="#">TcplpV6MtuConfig</a>		
<b>Description</b>	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)		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	[1 .. 86400]		
<b>Default value</b>	600		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

]

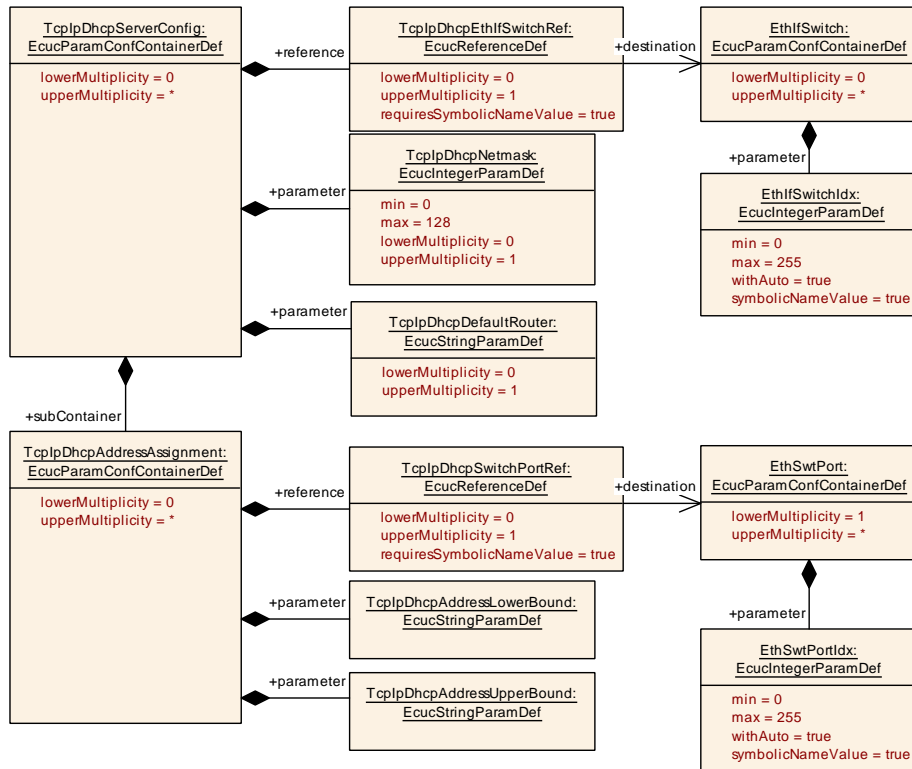


Figure 10.8: TcpIpDhcpServer

### 10.2.15 TcpIpDhcpServerConfig

[ECUC\_Tcplp\_00187] Definition of EcucParamConfContainerDef TcpIpDhcpServerConfig

Container Name	TcpIpDhcpServerConfig		
Parent Container	TcpIpConfig		
Description	Specifies the configuration parameters of the DHCP Server sub-module.		
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			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
TcpIpDhcpDefaultRouter	0..1	[ECUC_Tcplp_00190]
TcpIpDhcpNetmask	0..1	[ECUC_Tcplp_00189]
TcpIpDhcpEthIfSwitchRef	0..1	[ECUC_Tcplp_00188]



Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpDhcpAddressAssignment</a>	0..*	Defines a Ethernet Switch port based IP address assignment.

## [ECUC\_Tcplp\_00190] Definition of EcucStringParamDef TcplpDhcpDefault Router

Parameter Name	TcplpDhcpDefaultRouter		
Parent Container	<a href="#">TcplpDhcpServerConfig</a>		
Description	IP address of default router (gateway).		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		
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		

## [ECUC\_Tcplp\_00189] Definition of EcucIntegerParamDef TcplpDhcpNetmask

Parameter Name	TcplpDhcpNetmask		
Parent Container	<a href="#">TcplpDhcpServerConfig</a>		
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
--------------------	--------------

]

## [ECUC\_Tcplp\_00188] Definition of EcucReferenceDef TcplpDhcpEthIfSwitchRef

[

Parameter Name	TcplpDhcpEthIfSwitchRef		
Parent Container	<a href="#">TcplpDhcpServerConfig</a>		
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		

]

## 10.2.16 TcplpDhcpAddressAssignment

## [ECUC\_Tcplp\_00191] Definition of EcucParamConfContainerDef TcplpDhcpAddressAssignment

[

Container Name	TcplpDhcpAddressAssignment		
Parent Container	<a href="#">TcplpDhcpServerConfig</a>		
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			
Included Parameters			
Parameter Name	Multiplicity	ECUC ID	
<a href="#">TcplpDhcpAddressLowerBound</a>	1	<a href="#">[ECUC_Tcplp_00193]</a>	
<a href="#">TcplpDhcpAddressUpperBound</a>	1	<a href="#">[ECUC_Tcplp_00194]</a>	
<a href="#">TcplpDhcpSwitchPortRef</a>	0..1	<a href="#">[ECUC_Tcplp_00192]</a>	

No Included Containers

## [ECUC\_Tcplp\_00193] Definition of EcucStringParamDef TcplpDhcpAddressLowerBound

Parameter Name	TcplpDhcpAddressLowerBound		
Parent Container	<a href="#">TcplpDhcpAddressAssignment</a>		
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	–		
Regular Expression	–		
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		

## [ECUC\_Tcplp\_00194] Definition of EcucStringParamDef TcplpDhcpAddressUpperBound

Parameter Name	TcplpDhcpAddressUpperBound		
Parent Container	<a href="#">TcplpDhcpAddressAssignment</a>		
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	–		
Regular Expression	–		
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		

## [ECUC\_Tcplp\_00192] Definition of EcucReferenceDef TcplpDhcpSwitchPortRef

Parameter Name	TcplpDhcpSwitchPortRef		
Parent Container	<a href="#">TcplpDhcpAddressAssignment</a>		
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 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		

## 10.2.17 TcplpDuplicateAddressDetectionConfig

## [ECUC\_Tcplp\_00214] Definition of EcucParamConfContainerDef TcplpDuplicateAddressDetectionConfig

Container Name	TcplpDuplicateAddressDetectionConfig		
Parent Container	<a href="#">TcplpConfig</a>		
Description	Specifies the DAD callout function.		
Configuration Parameters			
Included Parameters			
Parameter Name		Multiplicity	ECUC ID
<a href="#">TcplpDuplicateAddressDetectionCalloutName</a>		1	[ECUC_Tcplp_00216]
No Included Containers			

## [ECUC\_Tcplp\_00216] Definition of EcucFunctionNameDef TcplpDuplicateAddressDetectionCalloutName

Parameter Name	TcplpDuplicateAddressDetectionCalloutName		
Parent Container	<a href="#">TcplpDuplicateAddressDetectionConfig</a>		
Description	This parameter defines the name of the DAD callout function <Up_DADAddressConflict>.		
Multiplicity	1		
Type	EcucFunctionNameDef		
Default value	–		
Regular Expression	–		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

]

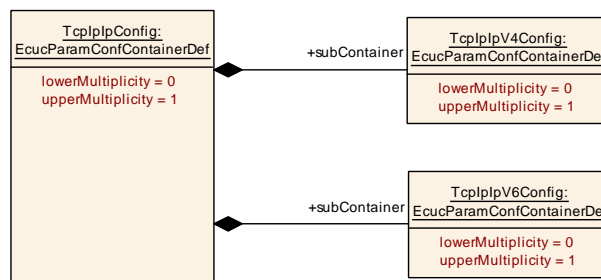


Figure 10.9: TcplpIpConfig

### 10.2.18 TcplpIpConfig

## [ECUC\_Tcplp\_00022] Definition of EcucParamConfContainerDef TcplpIpConfig

Container Name	TcplpIpConfig
Parent Container	<a href="#">TcplpConfig</a>
Description	Specifies the configuration parameters of the IP (Internet Protocol) sub-module
Configuration Parameters	
No Included 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.

└

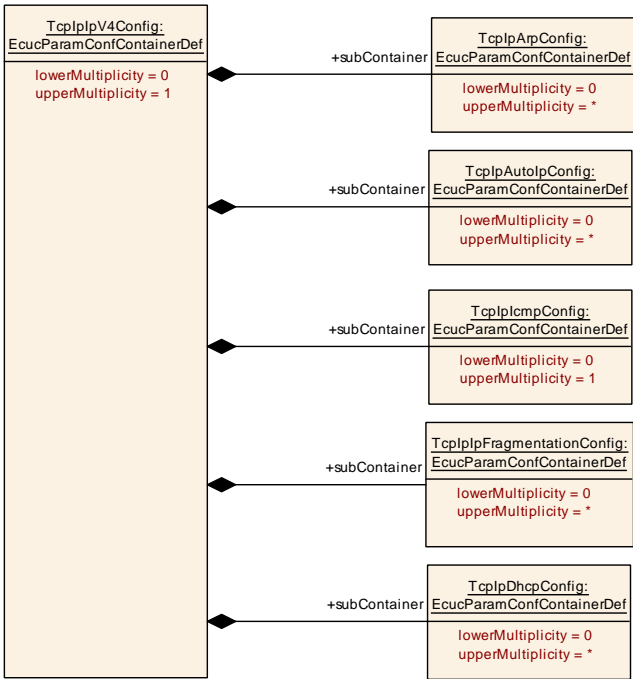


Figure 10.10: TcplpV4Config

10.2.19 TcplpV4Config

[ECUC\_Tcplp\_00095] Definition of EcucParamConfContainerDef TcplpV4Config

Container Name	TcplpV4Config
Parent Container	TcplpConfig
Description	Specifies the configuration parameters of the IPv4 (Internet Protocol version 4) sub-module.
Configuration Parameters	
No Included Parameters	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpArpConfig</a>	0..*	Specifies the configuration parameters of the ARP (Address Resolution Protocol) sub-module.
<a href="#">TcplpAutolpConfig</a>	0..*	Specifies the configuration parameters of the Auto-IP (automatic private IP addressing) sub-module.
<a href="#">TcplpDhcpConfig</a>	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.
<a href="#">TcplpIcmpConfig</a>	0..1	Specifies the configuration parameters of the ICMP (Internet Control Message Protocol) sub-module.
<a href="#">TcplpIpFragmentationConfig</a>	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.

]

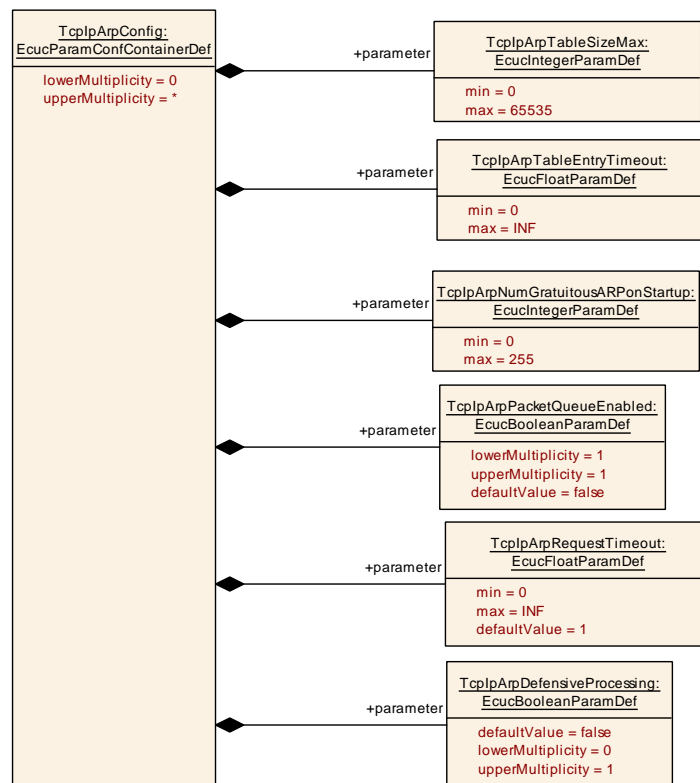


Figure 10.11: TcplpArpConfig

## 10.2.20 TcplpArpConfig

### [ECUC\_Tcplp\_00023] Definition of EcucParamConfContainerDef TcplpArpConfig

Container Name	TcplpArpConfig	
Parent Container	<a href="#">TcplpV4Config</a>	
Description	Specifies the configuration parameters of the ARP (Address Resolution Protocol) sub-module.	
Configuration Parameters		
Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpArpDefensiveProcessing</a>	0..1	[ECUC_Tcplp_00326]
<a href="#">TcplpArpNumGratuitousARPonStartup</a>	1	[ECUC_Tcplp_00054]
<a href="#">TcplpArpPacketQueueEnabled</a>	1	[ECUC_Tcplp_00170]
<a href="#">TcplpArpRequestTimeout</a>	1	[ECUC_Tcplp_00218]
<a href="#">TcplpArpTableEntryTimeout</a>	1	[ECUC_Tcplp_00053]
<a href="#">TcplpArpTableSizeMax</a>	1	[ECUC_Tcplp_00052]
No Included Containers		

### [ECUC\_Tcplp\_00326] Definition of EcucBooleanParamDef TcplpArpDefensive Processing

<b>Parameter Name</b>	TcplpArpDefensiveProcessing		
<b>Parent Container</b>	<a href="#">TcplpArpConfig</a>		
<b>Description</b>	If enabled the ARP shall only process ARP replies which are received in reaction to a previously transmitted ARP request as well as skipping updates to the ARP table based on received Gratuitous ARP packets. If disabled all ARP packets shall be processed as specified in IETF RFC 826.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	false		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		



## [ECUC\_Tcplp\_00054] Definition of EcucIntegerParamDef TcplpArpNumGratuitousARPonStartup

Parameter Name	TcplpArpNumGratuitousARPonStartup		
Parent Container	<a href="#">TcplpArpConfig</a>		
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		

## [ECUC\_Tcplp\_00170] Definition of EcucBooleanParamDef TcplpArpPacketQueueEnabled

Parameter Name	TcplpArpPacketQueueEnabled		
Parent Container	<a href="#">TcplpArpConfig</a>		
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		

## [ECUC\_Tcplp\_00218] Definition of EcucFloatParamDef TcplpArpRequestTimeout

Parameter Name	TcplpArpRequestTimeout		
Parent Container	<a href="#">TcplpArpConfig</a>		
Description	Specifies a timeout in seconds for the validity of ARP requests. After the transmission of an ARP request the Tcplp shall skip the transmission of any further ARP requests to the same destination within a duration of TcplpArpRequestTimeout seconds. (IETF RFC 1122, section 2.3.2.1) The value for this parameter shall be an integral multiple of TcplpMainFunctionPeriod or 0. If this parameter set to 0 this features is disabled and no delay between ARP requests is enforced.		





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

### [ECUC\_Tcplp\_00053] Definition of EcucFloatParamDef TcplpArpTableEntry Timeout

<b>Parameter Name</b>	TcplpArpTableEntryTimeout		
<b>Parent Container</b>	<a href="#">TcplpArpConfig</a>		
<b>Description</b>	Timeout in seconds after which an unused ARP entry is removed.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	]0 .. INF]		
<b>Default value</b>	–		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

### [ECUC\_Tcplp\_00052] Definition of EcucIntegerParamDef TcplpArpTableSizeMax

<b>Parameter Name</b>	TcplpArpTableSizeMax		
<b>Parent Container</b>	<a href="#">TcplpArpConfig</a>		
<b>Description</b>	Maximum number of entries in the ARP table.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 65535		
<b>Default value</b>	–		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

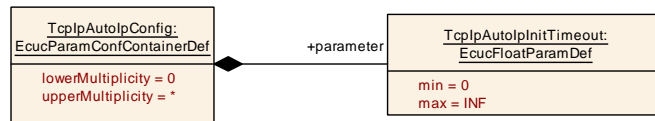


Figure 10.12: TcpIpAutoIpConfig

## 10.2.21 TcpIpAutoIpConfig

### [ECUC\_TcpIp\_00028] Definition of EcucParamConfContainerDef TcpIpAutoIpConfig

Container Name	TcpIpAutoIpConfig
Parent Container	<a href="#">TcpIpV4Config</a>
Description	Specifies the configuration parameters of the Auto-IP (automatic private IP addressing) sub-module.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcpIpAutoIpInitTimeout</a>	1	[ <a href="#">ECUC_TcpIp_00074</a> ]

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

### [ECUC\_TcpIp\_00074] Definition of EcucFloatParamDef TcpIpAutoIpInitTimeout

Parameter Name	TcpIpAutoIpInitTimeout		
Parent Container	<a href="#">TcpIpAutoIpConfig</a>		
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		

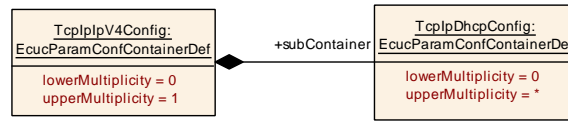


Figure 10.13: TcpIpDhcpConfig

## 10.2.22 TcpIpDhcpConfig

[ECUC\_TcpIp\_00167] Definition of EcucParamConfContainerDef TcpIpDhcpConfig [

Container Name	TcpIpDhcpConfig
Parent Container	<a href="#">TcpIpV4Config</a>
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 Parameters	
No Included Containers	

]

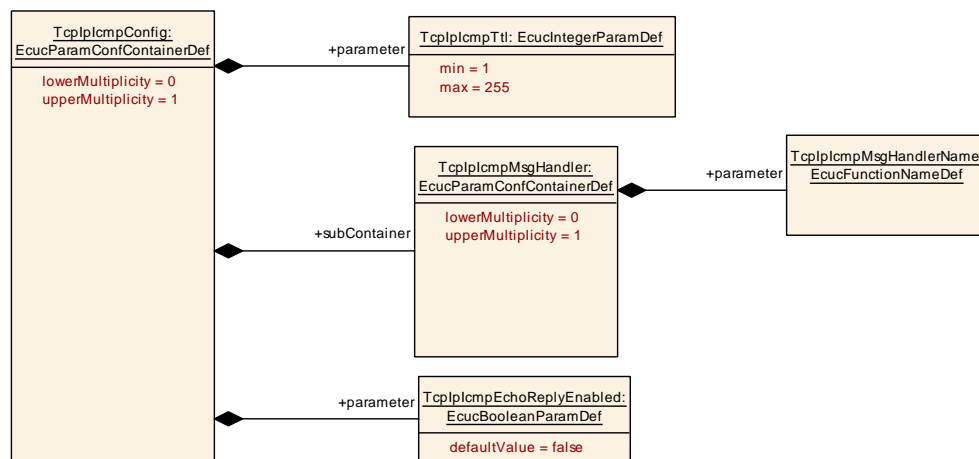


Figure 10.14: TcpIpLcmpConfig

### 10.2.23 TcplplcmpConfig

#### [ECUC\_Tcplp\_00024] Definition of EcucParamConfContainerDef TcplplcmpConfig

Container Name	TcplplcmpConfig
Parent Container	<a href="#">TcplplV4Config</a>
Description	Specifies the configuration parameters of the ICMP (Internet Control Message Protocol) sub-module.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplplcmpEchoReplyEnabled</a>	1	[ECUC_Tcplp_00213]
<a href="#">TcplplcmpTtl</a>	1	[ECUC_Tcplp_00055]

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplplcmpMsgHandler</a>	0..1	This container is a subcontainer of TcplplcmpConfig and specifies the configuration parameters for the ICMP message handler.

#### [ECUC\_Tcplp\_00213] Definition of EcucBooleanParamDef TcplplcmpEchoReplyEnabled

Parameter Name	TcplplcmpEchoReplyEnabled		
Parent Container	<a href="#">TcplplcmpConfig</a>		
Description	Enables or disables transmission of ICMP echo reply message in case of a ICMP echo reception.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

### [ECUC\_Tcplp\_00055] Definition of EcucIntegerParamDef TcplplcmpTtl [

Parameter Name	TcplplcmpTtl		
Parent Container	<a href="#">TcplplcmpConfig</a>		
Description	Default Time-to-live value of outgoing ICMP packets.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 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		

]

## 10.2.24 TcplplcmpMsgHandler

### [ECUC\_Tcplp\_00056] Definition of EcucParamConfContainerDef TcplplcmpMsg Handler [

Container Name	TcplplcmpMsgHandler		
Parent Container	<a href="#">TcplplcmpConfig</a>		
Description	This container is a subcontainer of TcplplcmpConfig and specifies the configuration parameters for the ICMP message handler.		
Configuration Parameters			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplplcmpMsgHandlerName</a>	1	[ECUC_Tcplp_00057]

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

]

### [ECUC\_Tcplp\_00057] Definition of EcucFunctionNameDef TcplplcmpMsgHandlerName [

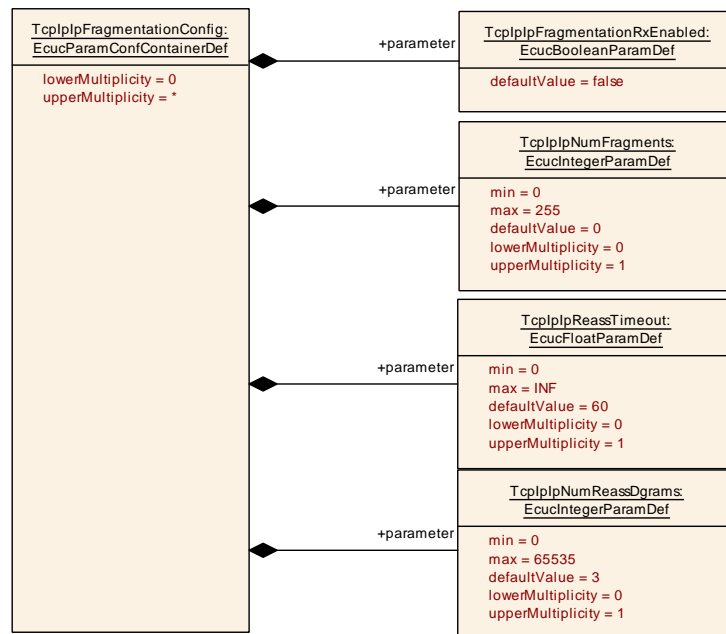
Parameter Name	TcplplcmpMsgHandlerName		
Parent Container	<a href="#">TcplplcmpMsgHandler</a>		
Description	This parameter defines the name of the ICMP message handler function <Up_Icmp MsgHandler>.		
Multiplicity	1		





Type	EcucFunctionNameDef		
Default value	–		
Regular Expression	–		
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		

└



**Figure 10.15: TcpIplpFragmentationConfig**

## 10.2.25 TcpIplpFragmentationConfig

**[ECUC\_TcpIplp\_00108] Definition of EcucParamConfContainerDef TcpIplpFragmentationConfig** ┌

Container Name	TcpIplpFragmentationConfig
Parent Container	<a href="#">TcpIplpV4Config</a>
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	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcpIpFragmentationRxEnabled</a>	1	[ECUC_TcpIp_00077]
<a href="#">TcpIpNumFragments</a>	0..1	[ECUC_TcpIp_00078]
<a href="#">TcpIpNumReassDgrams</a>	0..1	[ECUC_TcpIp_00080]
<a href="#">TcpIpReassTimeout</a>	0..1	[ECUC_TcpIp_00079]

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

### [ECUC\_TcpIp\_00077] Definition of EcucBooleanParamDef TcpIpFragmentationRxEnabled

Parameter Name	TcpIpFragmentationRxEnabled		
Parent Container	<a href="#">TcpIpFragmentationConfig</a>		
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		

### [ECUC\_TcpIp\_00078] Definition of EcucIntegerParamDef TcpIpNumFragments

Parameter Name	TcpIpNumFragments		
Parent Container	<a href="#">TcpIpFragmentationConfig</a>		
Description	Specifies the maximum number of IP fragments per datagram. Note: this parameter is only relevant if TcpIpFragmentationRxEnabled 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







	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local dependency: TcplpFragmentationRxEnabled		

]

## [ECUC\_Tcplp\_00080] Definition of EcucIntegerParamDef TcplpNumReassDgrams [

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

]

## [ECUC\_Tcplp\_00079] Definition of EcucFloatParamDef TcplpReassTimeout [

<b>Parameter Name</b>	TcplpReassTimeout		
<b>Parent Container</b>	<a href="#">TcplpFragmentationConfig</a>		
<b>Description</b>	Specifies the timeout in [s] after which an incomplete datagram gets discarded. Note: this parameter is only relevant if TcplpFragmentationRxEnabled is TRUE.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	[0 .. INF]		
<b>Default value</b>	60		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	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: TcpIpV6FragmentationRxEnabled		

└

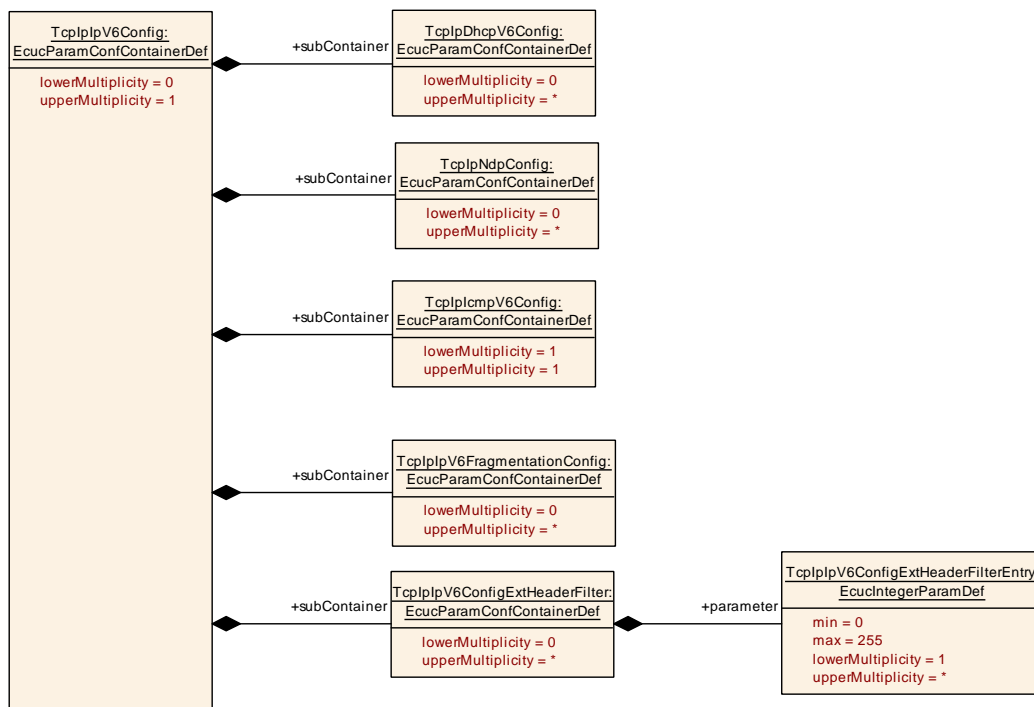


Figure 10.16: TcpIpV6Config

## 10.2.26 TcpIpV6Config

[ECUC\_TcpIp\_00168] Definition of EcucParamConfContainerDef TcpIpV6Config

Container Name	TcpIpV6Config
Parent Container	<a href="#">TcpIpConfig</a>
Description	Specifies the configuration parameters of the IPv6 (Internet Protocol version 6) sub-module.
Configuration Parameters	

No Included Parameters

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcpIpDhcpV6Config</a>	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.
<a href="#">TcpIpIcmpV6Config</a>	1	Specifies the configuration parameters of the ICMPv6 (Internet Control Message Protocol for IPv6) sub-module.
<a href="#">TcpIpIpV6ConfigExtHeaderFilter</a>	0..*	This container describes the white list for the filtering of IPv6 extension headers, i.e. frames containing IPv6 extension headers not listed here shall be silently dropped.
<a href="#">TcpIpIpV6FragmentationConfig</a>	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.
<a href="#">TcpIpNdpConfig</a>	0..*	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.

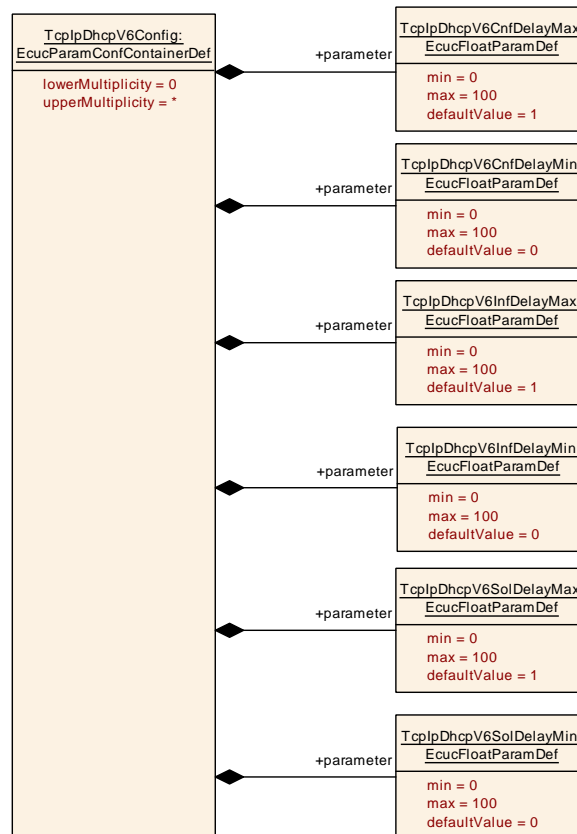


Figure 10.17: TcpIpDhcpV6Config

## 10.2.27 TcplpDhcpV6Config

### [ECUC\_Tcplp\_00110] Definition of EcucParamConfContainerDef TcplpDhcpV6Config

<b>Container Name</b>	TcplpDhcpV6Config
<b>Parent Container</b>	<a href="#">TcplpV6Config</a>
<b>Description</b>	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.
<b>Configuration Parameters</b>	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpDhcpV6CnfDelayMax</a>	1	[ECUC_Tcplp_00116]
<a href="#">TcplpDhcpV6CnfDelayMin</a>	1	[ECUC_Tcplp_00117]
<a href="#">TcplpDhcpV6InfDelayMax</a>	1	[ECUC_Tcplp_00118]
<a href="#">TcplpDhcpV6InfDelayMin</a>	1	[ECUC_Tcplp_00119]
<a href="#">TcplpDhcpV6SolDelayMax</a>	1	[ECUC_Tcplp_00120]
<a href="#">TcplpDhcpV6SolDelayMin</a>	1	[ECUC_Tcplp_00121]

<b>No Included Containers</b>
-------------------------------

### [ECUC\_Tcplp\_00116] Definition of EcucFloatParamDef TcplpDhcpV6CnfDelayMax

<b>Parameter Name</b>	TcplpDhcpV6CnfDelayMax		
<b>Parent Container</b>	<a href="#">TcplpDhcpV6Config</a>		
<b>Description</b>	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.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	[0 .. 100]		
<b>Default value</b>	1		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

### [ECUC\_Tcplp\_00117] Definition of EcucFloatParamDef TcplpDhcpV6CnfDelayMin

Parameter Name	TcplpDhcpV6CnfDelayMin		
Parent Container	<a href="#">TcplpDhcpV6Config</a>		
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		

### [ECUC\_Tcplp\_00118] Definition of EcucFloatParamDef TcplpDhcpV6InfDelayMax

Parameter Name	TcplpDhcpV6InfDelayMax		
Parent Container	<a href="#">TcplpDhcpV6Config</a>		
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		

### [ECUC\_Tcplp\_00119] Definition of EcucFloatParamDef TcplpDhcpV6InfDelayMin

Parameter Name	TcplpDhcpV6InfDelayMin		
Parent Container	<a href="#">TcplpDhcpV6Config</a>		
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		

### [ECUC\_Tcplp\_00120] Definition of EcucFloatParamDef TcplpDhcpV6SolDelay Max

Parameter Name	TcplpDhcpV6SolDelayMax		
Parent Container	<a href="#">TcplpDhcpV6Config</a>		
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		

### [ECUC\_Tcplp\_00121] Definition of EcucFloatParamDef TcplpDhcpV6SolDelay Min

Parameter Name	TcplpDhcpV6SolDelayMin		
Parent Container	<a href="#">TcplpDhcpV6Config</a>		
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		

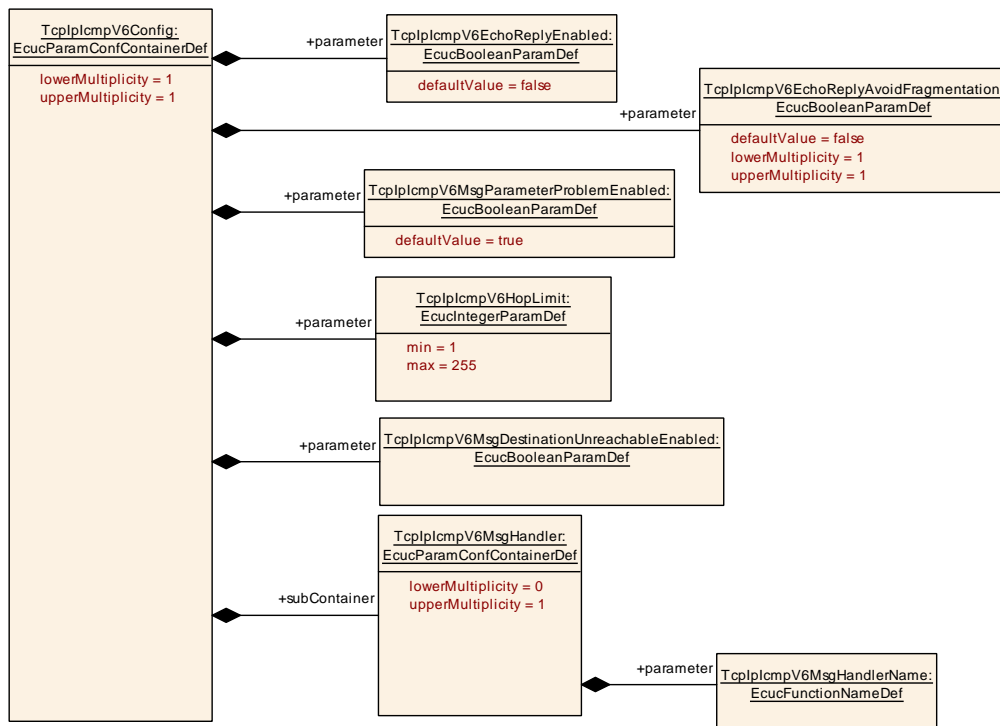


Figure 10.18: TcpIplcmpV6Config

## 10.2.28 TcpIplcmpV6Config

### [ECUC\_Tcplp\_00113] Definition of EcucParamConfContainerDef TcpIplcmpV6Config

Container Name	TcpIplcmpV6Config
Parent Container	<a href="#">TcplpV6Config</a>
Description	Specifies the configuration parameters of the ICMPv6 (Internet Control Message Protocol for IPv6) sub-module.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcpIplcmpV6EchoReplyAvoidFragmentation</a>	1	[ECUC_Tcplp_00212]
<a href="#">TcpIplcmpV6EchoReplyEnabled</a>	1	[ECUC_Tcplp_00149]
<a href="#">TcpIplcmpV6HopLimit</a>	1	[ECUC_Tcplp_00152]
<a href="#">TcpIplcmpV6MsgDestinationUnreachableEnabled</a>	1	[ECUC_Tcplp_00153]
<a href="#">TcpIplcmpV6MsgParameterProblemEnabled</a>	1	[ECUC_Tcplp_00151]

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcpIplcmpV6MsgHandler</a>	0..1	This container is a subcontainer of TcpIplcmpConfig and specifies the configuration parameters for the ICMPv6 message handler.

]

**[ECUC\_Tcplp\_00212] Definition of EcucBooleanParamDef TcplplcmpV6EchoReplyAvoidFragmentation** [

<b>Parameter Name</b>	TcplplcmpV6EchoReplyAvoidFragmentation		
<b>Parent Container</b>	<a href="#">TcplplcmpV6Config</a>		
<b>Description</b>	If enabled, the stack will respond only to incoming ICMPv6 Echo Requests (Pings) that fit the MTU of the respective interface, i.e. can be transmitted without IPv6 fragmentation. Only relevant if TcplplcmpV6EchoReplyEnabled is enabled.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local dependency: TcplplcmpV6EchoReplyEnabled		

]

**[ECUC\_Tcplp\_00149] Definition of EcucBooleanParamDef TcplplcmpV6EchoReplyEnabled** [

<b>Parameter Name</b>	TcplplcmpV6EchoReplyEnabled		
<b>Parent Container</b>	<a href="#">TcplplcmpV6Config</a>		
<b>Description</b>	If enabled, the stack will respond to incoming ICMPv6 Echo Requests (Pings).		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

]

**[ECUC\_Tcplp\_00152] Definition of EcucIntegerParamDef TcplplcmpV6HopLimit** [

<b>Parameter Name</b>	TcplplcmpV6HopLimit		
<b>Parent Container</b>	<a href="#">TcplplcmpV6Config</a>		
<b>Description</b>	Default Hop-Limit value of outgoing ICMPv6 packets.		
<b>Multiplicity</b>	1		







Type	EcucIntegerParamDef		
Range	1 .. 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		

### [ECUC\_Tcplp\_00153] Definition of EcucBooleanParamDef TcplplcmpV6MsgDestinationUnreachableEnabled

Parameter Name	TcplplcmpV6MsgDestinationUnreachableEnabled		
Parent Container	<a href="#">TcplplcmpV6Config</a>		
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		

### [ECUC\_Tcplp\_00151] Definition of EcucBooleanParamDef TcplplcmpV6MsgParameterProblemEnabled

Parameter Name	TcplplcmpV6MsgParameterProblemEnabled		
Parent Container	<a href="#">TcplplcmpV6Config</a>		
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. [RFC8200 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		

## 10.2.29 TcplplcmpV6MsgHandler

### [ECUC\_Tcplp\_00154] Definition of EcucParamConfContainerDef TcplplcmpV6MsgHandler

Container Name	TcplplcmpV6MsgHandler
Parent Container	<a href="#">TcplplcmpV6Config</a>
Description	This container is a subcontainer of TcplplcmpConfig and specifies the configuration parameters for the ICMPv6 message handler.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplplcmpV6MsgHandlerName</a>	1	[ECUC_Tcplp_00156]

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

### [ECUC\_Tcplp\_00156] Definition of EcucFunctionNameDef TcplplcmpV6MsgHandlerName

Parameter Name	TcplplcmpV6MsgHandlerName		
Parent Container	<a href="#">TcplplcmpV6MsgHandler</a>		
Description	This parameter defines the name of the ICMP message handler function <Up_Icmp MsgHandler>.		
Multiplicity	1		
Type	EcucFunctionNameDef		
Default value	–		
Regular Expression	–		
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		

## 10.2.30 TcplplpV6ConfigExtHeaderFilter

### [ECUC\_Tcplp\_00198] Definition of EcucParamConfContainerDef TcplplpV6ConfigExtHeaderFilter

<b>Container Name</b>	TcpIplpV6ConfigExtHeaderFilter
<b>Parent Container</b>	<a href="#">TcpIplpV6Config</a>
<b>Description</b>	This container describes the white list for the filtering of IPv6 extension headers, i.e. frames containing IPv6 extension headers not listed here shall be silently dropped.
<b>Post-Build Variant Multiplicity</b>	false
<b>Configuration Parameters</b>	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcpIplpV6ConfigExtHeaderFilterEntry</a>	1..*	[ECUC_TcpIp_00199]

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

## [ECUC\_TcpIp\_00199] Definition of EcucIntegerParamDef TcpIplpV6ConfigExtHeaderFilterEntry [

<b>Parameter Name</b>	TcpIplpV6ConfigExtHeaderFilterEntry		
<b>Parent Container</b>	<a href="#">TcpIplpV6ConfigExtHeaderFilter</a>		
<b>Description</b>	IPv6 Extension Header type allowed by this filter.		
<b>Multiplicity</b>	1..*		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default value</b>	–		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

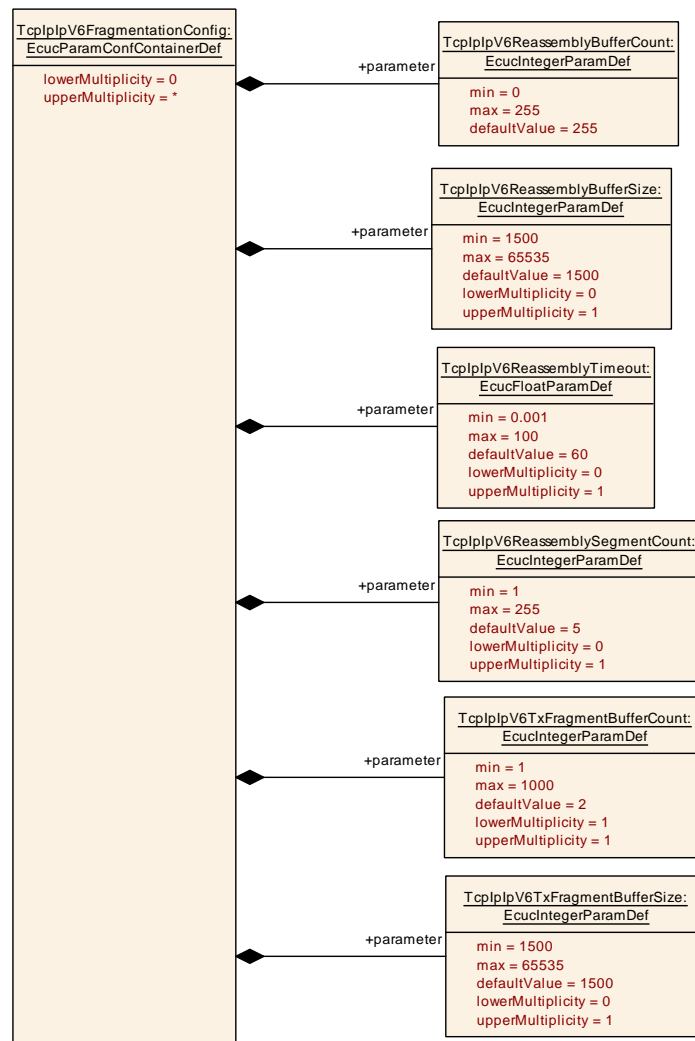


Figure 10.19: TcplpV6FragmentationConfig

### 10.2.31 TcplpV6FragmentationConfig

[ECUC\_Tcplp\_00114] Definition of EcucParamConfContainerDef TcplpV6FragmentationConfig

Container Name	TcplpV6FragmentationConfig
Parent Container	<a href="#">TcplpV6Config</a>
Description	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.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpV6ReassemblyBufferCount</a>	1	<a href="#">[ECUC_Tcplp_00157]</a>
<a href="#">TcplpV6ReassemblyBufferSize</a>	0..1	<a href="#">[ECUC_Tcplp_00158]</a>
<a href="#">TcplpV6ReassemblySegmentCount</a>	0..1	<a href="#">[ECUC_Tcplp_00160]</a>
<a href="#">TcplpV6ReassemblyTimeout</a>	0..1	<a href="#">[ECUC_Tcplp_00159]</a>
<a href="#">TcplpV6TxFragmentBufferCount</a>	1	<a href="#">[ECUC_Tcplp_00161]</a>
<a href="#">TcplpV6TxFragmentBufferSize</a>	0..1	<a href="#">[ECUC_Tcplp_00162]</a>

No Included Containers

## [ECUC\_Tcplp\_00157] Definition of EcucIntegerParamDef TcplpV6ReassemblyBufferCount

Parameter Name	TcplpV6ReassemblyBufferCount		
Parent Container	<a href="#">TcplpV6FragmentationConfig</a>		
Description	<p>Number of buffers that can be used for fragment reassembly. In case of a reassembly error or if not all fragments are received in time this buffer will be blocked until the specified "Fragment Reassembly Timeout" has been exceeded.</p> <p>A value of 0 disables fragment reassembly.</p> <p>[RFC8200 5. Packet Size Issues] "In order to send a packet larger than a path's MTU, a node may use the IPv6 Fragment header to fragment the packet at the source and have it reassembled at the destination(s). However, the use of such fragmentation is discouraged in any application that is able to adjust its packets to fit the measured path MTU (i.e., down to 1280 octets)."</p>		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	255		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

## [ECUC\_Tcplp\_00158] Definition of EcucIntegerParamDef TcplpV6ReassemblyBufferSize

Parameter Name	TcplpV6ReassemblyBufferSize		
Parent Container	<a href="#">TcplpV6FragmentationConfig</a>		
Description	<p>[RFC8200 5. Packet Size Issues] "A node must be able to accept a fragmented packet that, after reassembly, is as large as 1500 octets. A node is permitted to accept fragmented packets that reassemble to more than 1500 octets."the measured path MTU (i.e., down to 1280 octets)."</p>		





<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	1500 .. 65535		
<b>Default value</b>	1500		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

## [ECUC\_Tcplp\_00160] Definition of EcucIntegerParamDef TcplpV6Reassembly SegmentCount

<b>Parameter Name</b>	TcplpV6ReassemblySegmentCount		
<b>Parent Container</b>	<a href="#">TcplpV6FragmentationConfig</a>		
<b>Description</b>	<p>Specifies the maximum number of consecutive data segments that can be managed in each reassembly buffer. If all fragments are received in order, only one segment will be needed.</p> <p>To deal with fragments received out of order this value should be configured bigger than 1.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	1 .. 255		
<b>Default value</b>	5		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

## [ECUC\_Tcplp\_00159] Definition of EcucFloatParamDef TcplpV6ReassemblyTimeout

Parameter Name	TcplpV6ReassemblyTimeout		
Parent Container	<a href="#">TcplpV6FragmentationConfig</a>		
Description	[RFC8200 4.5 Fragment Header] Default: 60 seconds		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0.001 .. 100]		
Default value	60		
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		

## [ECUC\_Tcplp\_00161] Definition of EcucIntegerParamDef TcplpV6TxFragmentBufferCount

Parameter Name	TcplpV6TxFragmentBufferCount		
Parent Container	<a href="#">TcplpV6FragmentationConfig</a>		
Description	<p>These buffers will be used if the IpV6 receives packets from the upper layer that do not fit into the MTU and thus must be fragmented.</p> <p>A value of 0 disables tx fragmentation.</p> <p>If the upper layer transmits packets that do not fit into the link or path MTU, the IpV6 will split-up the packet into fragments.</p> <p>see "Enable Fragment Reassembly"</p>		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 1000		
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		

## [ECUC\_Tcplp\_00162] Definition of EcucIntegerParamDef TcplpV6TxFragmentBufferSize

Parameter Name	TcplpV6TxFragmentBufferSize		
Parent Container	<a href="#">TcplpV6FragmentationConfig</a>		
Description	Size of each fragment tx buffer in bytes		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1500 .. 65535		
Default value	1500		
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		



Figure 10.20: TcplpNdpConfig

### 10.2.32 TcplpNdpConfig

## [ECUC\_Tcplp\_00112] Definition of EcucParamConfContainerDef TcplpNdpConfig

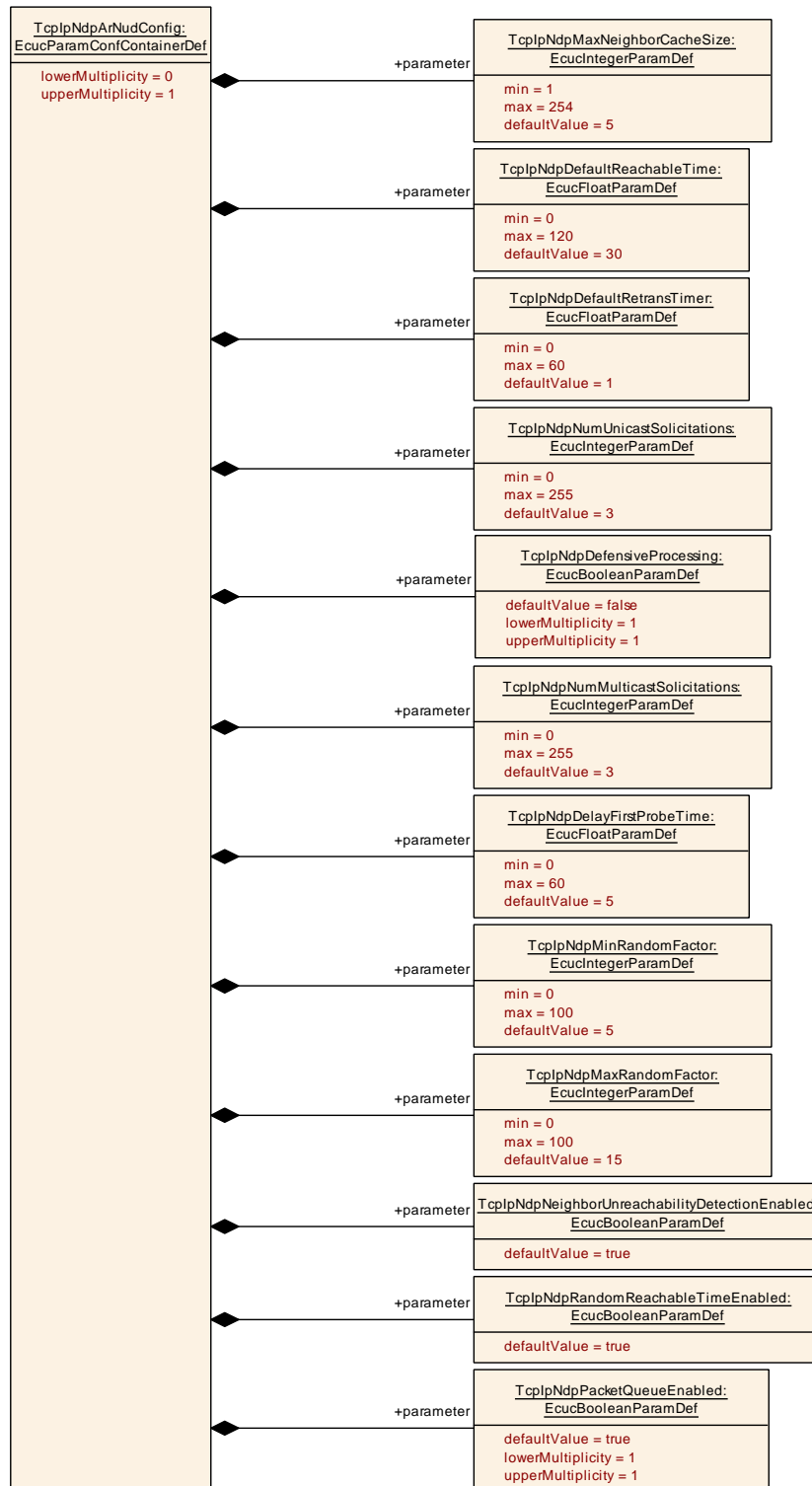


<b>Container Name</b>	TcpIpNdpConfig
<b>Parent Container</b>	<a href="#">TcpIpV6Config</a>
<b>Description</b>	Specifies the configuration parameters of the Neighbor Discovery Protocol for IPv6  This container may be referenced by multiple IPv6 instances if they shall use the same configuration. This container may have multiple instances if different configurations are required for different IPv6 instances.
<b>Configuration Parameters</b>	

<b>No Included Parameters</b>
-------------------------------

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcpIpNdpArNudConfig</a>	0..1	Specifies the configuration parameters for NDP Address Resolution and Neighbor Unreachability Detection.
<a href="#">TcpIpNdpPrefixRouterDiscoveryConfig</a>	0..1	Specifies the configuration parameters for NDP Prefix and Router Discovery.
<a href="#">TcpIpNdpSlaacConfig</a>	0..1	Specifies the configuration parameters for Stateless Address AutoConfiguration.

]



**Figure 10.21: TcpIpNdpArNudConfig**

## 10.2.33 TcplpNdpArNudConfig

**[ECUC\_Tcplp\_00123] Definition of EcucParamConfContainerDef TcplpNdpArNudConfig** [

<b>Container Name</b>	TcplpNdpArNudConfig
<b>Parent Container</b>	<a href="#">TcplpNdpConfig</a>
<b>Description</b>	Specifies the configuration parameters for NDP Address Resolution and Neighbor Unreachability Detection.
<b>Configuration Parameters</b>	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpNdpDefaultReachableTime</a>	1	[ECUC_Tcplp_00130]
<a href="#">TcplpNdpDefaultRetransTimer</a>	1	[ECUC_Tcplp_00165]
<a href="#">TcplpNdpDefensiveProcessing</a>	1	[ECUC_Tcplp_00201]
<a href="#">TcplpNdpDelayFirstProbeTime</a>	1	[ECUC_Tcplp_00133]
<a href="#">TcplpNdpMaxNeighborCacheSize</a>	1	[ECUC_Tcplp_00129]
<a href="#">TcplpNdpMaxRandomFactor</a>	1	[ECUC_Tcplp_00135]
<a href="#">TcplpNdpMinRandomFactor</a>	1	[ECUC_Tcplp_00134]
<a href="#">TcplpNdpNeighborUnreachabilityDetectionEnabled</a>	1	[ECUC_Tcplp_00136]
<a href="#">TcplpNdpNumMulticastSolicitations</a>	1	[ECUC_Tcplp_00132]
<a href="#">TcplpNdpNumUnicastSolicitations</a>	1	[ECUC_Tcplp_00131]
<a href="#">TcplpNdpPacketQueueEnabled</a>	1	[ECUC_Tcplp_00171]
<a href="#">TcplpNdpRandomReachableTimeEnabled</a>	1	[ECUC_Tcplp_00137]

<b>No Included Containers</b>
-------------------------------

]

**[ECUC\_Tcplp\_00130] Definition of EcucFloatParamDef TcplpNdpDefaultReachableTime** [

<b>Parameter Name</b>	TcplpNdpDefaultReachableTime	
<b>Parent Container</b>	<a href="#">TcplpNdpArNudConfig</a>	
<b>Description</b>	<p>Configuration of the ReachableTime (s) specified in [RFC4861 6.3.2. Host Variables].</p> <p>"The time a neighbor is considered reachable after receiving a reachability confirmation."</p> <p>If "TcplpNdpDynamicReachableTimeEnabled" is checked, this value may be reconfigured based on received Router Advertisements.</p> <p>Default: REACHABLE_TIME = 30 seconds</p>	
<b>Multiplicity</b>	1	
<b>Type</b>	EcucFloatParamDef	
<b>Range</b>	[0 .. 120]	
<b>Default value</b>	30	
<b>Post-Build Variant Value</b>	false	





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

## [ECUC\_Tcplp\_00165] Definition of EcucFloatParamDef TcplpNdpDefaultRetrans Timer

Parameter Name	TcplpNdpDefaultRetransTimer		
Parent Container	<a href="#">TcplpNdpArNudConfig</a>		
Description	<p>Configures the default value (s) for the RetransTimer variable specified in [RFC4861 6.3.2. Host Variables].</p> <p>"The time between retransmissions of Neighbor Solicitation messages to a neighbor when resolving the address or when probing the reachability of a neighbor."</p> <p>If "TcplpNdpDynamicRetransTimeEnabled" is checked, this value may be reconfigured based on received Router Advertisements.</p> <p>Default: RETRANS_TIMER = 1 second</p>		
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		

## [ECUC\_Tcplp\_00201] Definition of EcucBooleanParamDef TcplpNdpDefensive Processing

Parameter Name	TcplpNdpDefensiveProcessing		
Parent Container	<a href="#">TcplpNdpArNudConfig</a>		
Description	<p>If enabled the NDP shall only process Neighbor Advertisements which are received in reaction to a previously transmitted Neighbor Solicitation as well as skipping updates to the Neighbor Cache based on received Neighbor Solicitations. If disabled all Neighbor Advertisements and Solicitations shall be processed as specified in RFC4861. [RFC4861 7.2.5. Receipt of Neighbor Advertisements]</p>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	





	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

└

### [ECUC\_Tcplp\_00133] Definition of EcucFloatParamDef TcplpNdpDelayFirstProbeTime

<b>Parameter Name</b>	TcplpNdpDelayFirstProbeTime		
<b>Parent Container</b>	<a href="#">TcplpNdpArNudConfig</a>		
<b>Description</b>	Delay before sending the first NUD probe in (s). [RFC4861 7.3.3. Node Behavior] Default: DELAY_FIRST_PROBE_TIME = 5 seconds		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	[0 .. 60]		
<b>Default value</b>	5		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

└

### [ECUC\_Tcplp\_00129] Definition of EcucIntegerParamDef TcplpNdpMaxNeighborCacheSize

<b>Parameter Name</b>	TcplpNdpMaxNeighborCacheSize		
<b>Parent Container</b>	<a href="#">TcplpNdpArNudConfig</a>		
<b>Description</b>	Maximum number of entries in the neighbor cache. [RFC4861 5.1. Conceptual Data Structures]		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	1 .. 254		
<b>Default value</b>	5		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

└

## [ECUC\_Tcplp\_00135] Definition of EcucIntegerParamDef TcplpNdpMaxRandom Factor

Parameter Name	TcplpNdpMaxRandomFactor		
Parent Container	<a href="#">TcplpNdpArNudConfig</a>		
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		

## [ECUC\_Tcplp\_00134] Definition of EcucIntegerParamDef TcplpNdpMinRandom Factor

Parameter Name	TcplpNdpMinRandomFactor		
Parent Container	<a href="#">TcplpNdpArNudConfig</a>		
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		

## [ECUC\_Tcplp\_00136] Definition of EcucBooleanParamDef TcplpNdpNeighborUnreachabilityDetectionEnabled

Parameter Name	TcplpNdpNeighborUnreachabilityDetectionEnabled		
Parent Container	<a href="#">TcplpNdpArNudConfig</a>		
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		

## [ECUC\_Tcplp\_00132] Definition of EcucIntegerParamDef TcplpNdpNumMulticastSolicitations

Parameter Name	TcplpNdpNumMulticastSolicitations		
Parent Container	<a href="#">TcplpNdpArNudConfig</a>		
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		

## [ECUC\_Tcplp\_00131] Definition of EcucIntegerParamDef TcplpNdpNumUnicast Solicitations

Parameter Name	TcplpNdpNumUnicastSolicitations		
Parent Container	<a href="#">TcplpNdpArNudConfig</a>		
Description	Maximum number of unicast solicitations that will be sent when performig 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		

## [ECUC\_Tcplp\_00171] Definition of EcucBooleanParamDef TcplpNdpPacket QueueEnabled

Parameter Name	TcplpNdpPacketQueueEnabled		
Parent Container	<a href="#">TcplpNdpArNudConfig</a>		
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		



# [ECUC\_Tcplp\_00137] Definition of EcucBooleanParamDef TcplpNdpRandomReachableTimeEnabled [

Parameter Name	TcplpNdpRandomReachableTimeEnabled		
Parent Container	<a href="#">TcplpNdpArNudConfig</a>		
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		

]

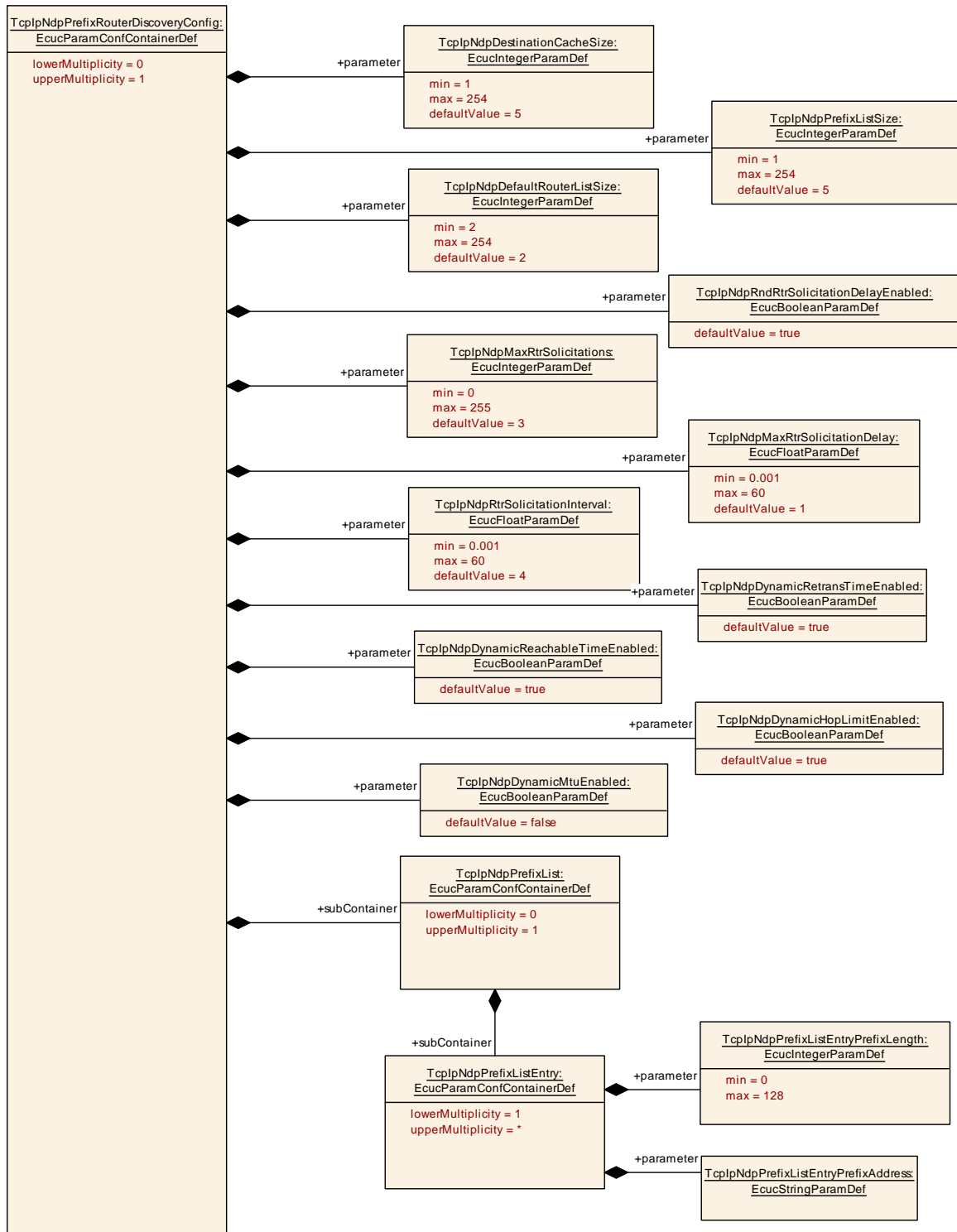


Figure 10.22: TcpIpNdpPrefixRouterDiscoveryConfig

### 10.2.34 TcplpNdpPrefixRouterDiscoveryConfig

#### [ECUC\_Tcplp\_00124] Definition of EcucParamConfContainerDef TcplpNdpPrefixRouterDiscoveryConfig

Container Name	TcplpNdpPrefixRouterDiscoveryConfig
Parent Container	<a href="#">TcplpNdpConfig</a>
Description	Specifies the configuration parameters for NDP Prefix and Router Discovery.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpNdpDefaultRouterListSize</a>	1	[ECUC_Tcplp_00139]
<a href="#">TcplpNdpDestinationCacheSize</a>	1	[ECUC_Tcplp_00138]
<a href="#">TcplpNdpDynamicHopLimitEnabled</a>	1	[ECUC_Tcplp_00147]
<a href="#">TcplpNdpDynamicMtuEnabled</a>	1	[ECUC_Tcplp_00148]
<a href="#">TcplpNdpDynamicReachableTimeEnabled</a>	1	[ECUC_Tcplp_00146]
<a href="#">TcplpNdpDynamicRetransTimeEnabled</a>	1	[ECUC_Tcplp_00145]
<a href="#">TcplpNdpMaxRtrSolicitationDelay</a>	1	[ECUC_Tcplp_00143]
<a href="#">TcplpNdpMaxRtrSolicitations</a>	1	[ECUC_Tcplp_00142]
<a href="#">TcplpNdpPrefixListSize</a>	1	[ECUC_Tcplp_00140]
<a href="#">TcplpNdpRndRtrSolicitationDelayEnabled</a>	1	[ECUC_Tcplp_00141]
<a href="#">TcplpNdpRtrSolicitationInterval</a>	1	[ECUC_Tcplp_00144]

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpNdpPrefixList</a>	0..1	Specifies a list of prefixes to be treated as "on-link" according to IETF RFC 4861 Section 5.1.

#### [ECUC\_Tcplp\_00139] Definition of EcucIntegerParamDef TcplpNdpDefaultRouterListSize

Parameter Name	TcplpNdpDefaultRouterListSize		
Parent Container	<a href="#">TcplpNdpPrefixRouterDiscoveryConfig</a>		
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
--------------------	--------------

]

## [ECUC\_Tcplp\_00138] Definition of EcucIntegerParamDef TcplpNdpDestinationCacheSize [

Parameter Name	TcplpNdpDestinationCacheSize		
Parent Container	<a href="#">TcplpNdpPrefixRouterDiscoveryConfig</a>		
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		

]

## [ECUC\_Tcplp\_00147] Definition of EcucBooleanParamDef TcplpNdpDynamicHopLimitEnabled [

Parameter Name	TcplpNdpDynamicHopLimitEnabled		
Parent Container	<a href="#">TcplpNdpPrefixRouterDiscoveryConfig</a>		
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		

]

### [ECUC\_Tcplp\_00148] Definition of EcucBooleanParamDef TcplpNdpDynamicMtuEnabled

Parameter Name	TcplpNdpDynamicMtuEnabled		
Parent Container	<a href="#">TcplpNdpPrefixRouterDiscoveryConfig</a>		
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		

### [ECUC\_Tcplp\_00146] Definition of EcucBooleanParamDef TcplpNdpDynamicReachableTimeEnabled

Parameter Name	TcplpNdpDynamicReachableTimeEnabled		
Parent Container	<a href="#">TcplpNdpPrefixRouterDiscoveryConfig</a>		
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		

## [ECUC\_Tcplp\_00145] Definition of EcucBooleanParamDef TcplpNdpDynamicRetransTimeEnabled [

Parameter Name	TcplpNdpDynamicRetransTimeEnabled		
Parent Container	<a href="#">TcplpNdpPrefixRouterDiscoveryConfig</a>		
Description	<p>If enabled the default Retransmit Timer value may be reconfigured based on received Router Advertisements.</p> <p>[RFC4861 6.3.4. Processing Received Router Advertisements]</p> <p>Default: Enabled</p>		
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		

]

## [ECUC\_Tcplp\_00143] Definition of EcucFloatParamDef TcplpNdpMaxRtrSolicitationDelay [

Parameter Name	TcplpNdpMaxRtrSolicitationDelay		
Parent Container	<a href="#">TcplpNdpPrefixRouterDiscoveryConfig</a>		
Description	<p>Maximum delay before the first Router Solicitation will be sent after interface initialization in (s).</p> <p>[RFC4861 6.3.7. Sending Router Solicitations]</p> <p>Default: MAX_RTR_SOLICITATION_DELAY = 1 second</p>		
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		

]

## [ECUC\_Tcplp\_00142] Definition of EcucIntegerParamDef TcplpNdpMaxRtrSolicitations

Parameter Name	TcplpNdpMaxRtrSolicitations		
Parent Container	<a href="#">TcplpNdpPrefixRouterDiscoveryConfig</a>		
Description	<p>Maximum number of Router Solicitations that will be sent before the first Router Advertisement has been received.</p> <p>0 = No Router Solicitations will be sent. This has no impact on handling Router Advertisements.</p> <p>[RFC4861 6.3.7. Sending Router Solicitations]</p> <p>Default: MAX_RTR_SOLICITATIONS = 3 transmissions</p>		
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		

## [ECUC\_Tcplp\_00140] Definition of EcucIntegerParamDef TcplpNdpPrefixListSize

Parameter Name	TcplpNdpPrefixListSize		
Parent Container	<a href="#">TcplpNdpPrefixRouterDiscoveryConfig</a>		
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		

### [ECUC\_Tcplp\_00141] Definition of EcucBooleanParamDef TcplpNdpRndRtrSolicitationDelayEnabled [

Parameter Name	TcplpNdpRndRtrSolicitationDelayEnabled		
Parent Container	<a href="#">TcplpNdpPrefixRouterDiscoveryConfig</a>		
Description	<p>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.</p> <p>[RFC4861 6.3.7. Sending Router Solicitations]</p> <p>Default: Enabled</p>		
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		

]

### [ECUC\_Tcplp\_00144] Definition of EcucFloatParamDef TcplpNdpRtrSolicitationInterval [

Parameter Name	TcplpNdpRtrSolicitationInterval		
Parent Container	<a href="#">TcplpNdpPrefixRouterDiscoveryConfig</a>		
Description	<p>Interval between consecutive Router Solicitations in (s).</p> <p>[RFC4861 6.3.7. Sending Router Solicitations]</p> <p>Default: RTR_SOLICITATION_INTERVAL = 4 seconds</p>		
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		

]

## 10.2.35 TcplpNdpPrefixList

### [ECUC\_Tcplp\_00205] Definition of EcucParamConfContainerDef TcplpNdpPrefixList [



Container Name	TcplpNdpPrefixList
Parent Container	<a href="#">TcplpNdpPrefixRouterDiscoveryConfig</a>
Description	Specifies a list of prefixes to be treated as "on-link" according to IETF RFC 4861 Section 5.1.
Configuration Parameters	

No Included Parameters
------------------------

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpNdpPrefixListEntry</a>	1..*	Single entry in the prefix list.

]

### 10.2.36 TcplpNdpPrefixListEntry

#### [ECUC\_Tcplp\_00206] Definition of EcucParamConfContainerDef TcplpNdpPrefixListEntry [

Container Name	TcplpNdpPrefixListEntry
Parent Container	<a href="#">TcplpNdpPrefixList</a>
Description	Single entry in the prefix list.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpNdpPrefixListEntryPrefixAddress</a>	1	[ECUC_Tcplp_00208]
<a href="#">TcplpNdpPrefixListEntryPrefixLength</a>	1	[ECUC_Tcplp_00207]

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

]

#### [ECUC\_Tcplp\_00208] Definition of EcucStringParamDef TcplpNdpPrefixListEntryPrefixAddress [

Parameter Name	TcplpNdpPrefixListEntryPrefixAddress
Parent Container	<a href="#">TcplpNdpPrefixListEntry</a>
Description	The prefix of an IP address. This prefix can be used for on-link determination.
Multiplicity	1
Type	EcucStringParamDef
Default value	–





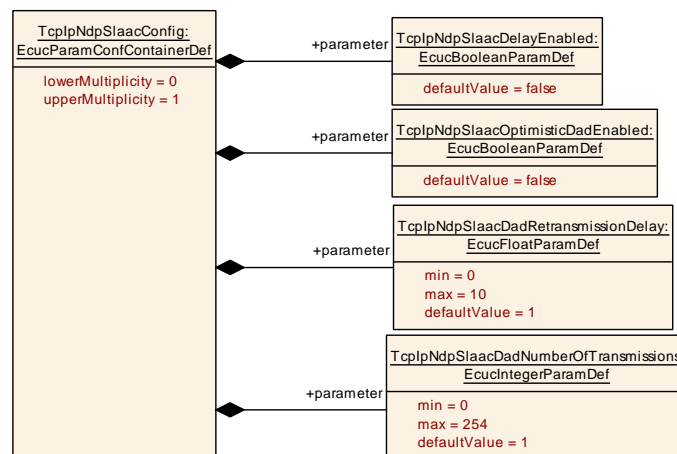
Regular Expression	–		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

## [ECUC\_Tcplp\_00207] Definition of EcucIntegerParamDef TcplpNdpPrefixListEntryPrefixLength

Parameter Name	TcplpNdpPrefixListEntryPrefixLength		
Parent Container	<a href="#">TcplpNdpPrefixListEntry</a>		
Description	The number of leading bits in the Prefix that are valid.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 128		
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		

]



**Figure 10.23: TcplpNdpSlaacConfig**

### 10.2.37 TcplpNdpSlaacConfig

#### [ECUC\_Tcplp\_00122] Definition of EcucParamConfContainerDef TcplpNdpSlaacConfig

Container Name	TcplpNdpSlaacConfig
Parent Container	<a href="#">TcplpNdpConfig</a>
Description	Specifies the configuration parameters for StateLess Address AutoConfiguration.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpNdpSlaacDadNumberOfTransmissions</a>	1	<a href="#">[ECUC_Tcplp_00128]</a>
<a href="#">TcplpNdpSlaacDadRetransmissionDelay</a>	1	<a href="#">[ECUC_Tcplp_00127]</a>
<a href="#">TcplpNdpSlaacDelayEnabled</a>	1	<a href="#">[ECUC_Tcplp_00125]</a>
<a href="#">TcplpNdpSlaacOptimisticDadEnabled</a>	1	<a href="#">[ECUC_Tcplp_00126]</a>

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

#### [ECUC\_Tcplp\_00128] Definition of EcucIntegerParamDef TcplpNdpSlaacDadNumberOfTransmissions

Parameter Name	TcplpNdpSlaacDadNumberOfTransmissions		
Parent Container	<a href="#">TcplpNdpSlaacConfig</a>		
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		

## [ECUC\_Tcplp\_00127] Definition of EcucFloatParamDef TcplpNdpSlaacDadRetransmissionDelay

Parameter Name	TcplpNdpSlaacDadRetransmissionDelay		
Parent Container	<a href="#">TcplpNdpSlaacConfig</a>		
Description	<p>Sets the maximum value for the address configuration delay (s).</p> <p>According to [RFC4861 5.4.2. Sending Neighbor Solicitation Messages] this value should be the same as MAX_RTR_SOLICITATION_DELAY.</p> <p>Default: MAX_RTR_SOLICITATION_DELAY = 1 second</p>		
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		

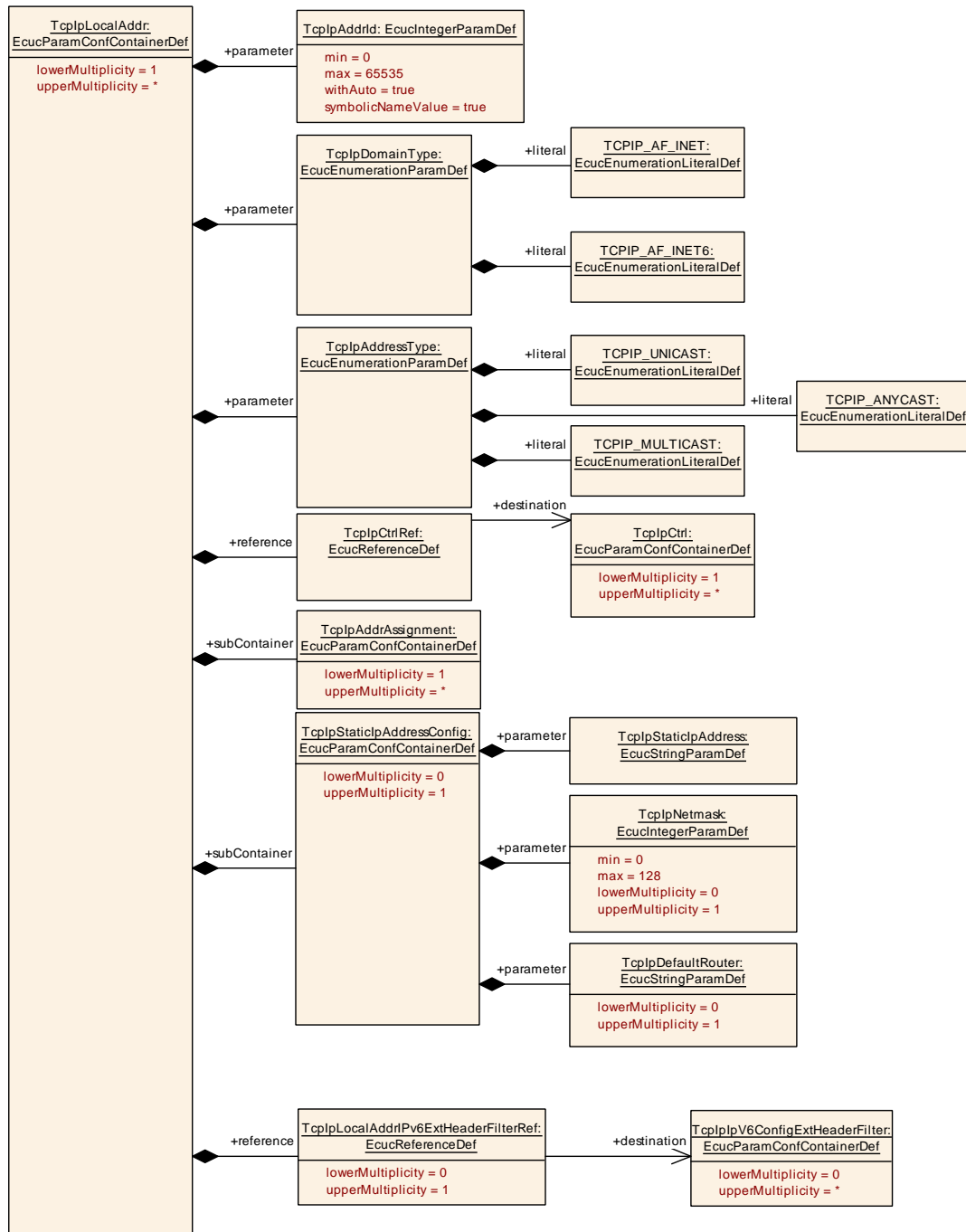
## [ECUC\_Tcplp\_00125] Definition of EcucBooleanParamDef TcplpNdpSlaacDelayEnabled

Parameter Name	TcplpNdpSlaacDelayEnabled		
Parent Container	<a href="#">TcplpNdpSlaacConfig</a>		
Description	<p>If enabled transmission of the first DAD Neighbor Solicitation will be delayed by a random value from [0...MAX_DAD_DELAY].</p> <p>"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."</p> <p>"The delay will avoid similar congestion when multiple nodes are going to configure addresses by receiving the same single multicast router advertisement."</p> <p>[RFC4861 5.4.2. Sending Neighbor Solicitation Messages]</p> <p>Default: True</p>		
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		

# [ECUC\_Tcplp\_00126] Definition of EcucBooleanParamDef TcplpNdpSlaacOptimisticDadEnabled [

Parameter Name	TcplpNdpSlaacOptimisticDadEnabled		
Parent Container	<a href="#">TcplpNdpSlaacConfig</a>		
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		

]



**Figure 10.24: TcpIpLocalAddr**

### 10.2.38 TcpIpLocalAddr

**[ECUC\_TcpIp\_00020] Definition of EcucParamConfContainerDef TcpIpLocalAddr**

<b>Container Name</b>	TcplpLocalAddr
<b>Parent Container</b>	<a href="#">TcplpConfig</a>
<b>Description</b>	Specifies the local IP (Internet Protocol) addresses used for IP communication.
<b>Configuration Parameters</b>	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpAddressType</a>	1	[ECUC_Tcplp_00031]
<a href="#">TcplpAddrId</a>	1	[ECUC_Tcplp_00029]
<a href="#">TcplpDomainType</a>	1	[ECUC_Tcplp_00030]
<a href="#">TcplpCtrlRef</a>	1	[ECUC_Tcplp_00032]
<a href="#">TcplpLocalAddrIPv6ExtHeaderFilterRef</a>	0..1	[ECUC_Tcplp_00200]

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

## [ECUC\_Tcplp\_00031] Definition of EcucEnumerationParamDef TcplpAddress Type 「

Parameter Name	TcplpAddressType		
Parent Container	<a href="#">TcplpLocalAddr</a>		
Description	Address type.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	TCPIP_ANYCAST	Anycast address	
	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		

### [ECUC\_Tcplp\_00029] Definition of EcucIntegerParamDef TcplpAddrId [

Parameter Name	TcplpAddrId		
Parent Container	<a href="#">TcplpLocalAddr</a>		
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 withAuto = true		

]

### [ECUC\_Tcplp\_00030] Definition of EcucEnumerationParamDef TcplpDomain Type [

Parameter Name	TcplpDomainType		
Parent Container	<a href="#">TcplpLocalAddr</a>		
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		

]

### [ECUC\_Tcplp\_00032] Definition of EcucReferenceDef TcplpCtrlRef [

Parameter Name	TcplpCtrlRef		
Parent Container	<a href="#">TcplpLocalAddr</a>		
Description	Reference to a TcplpCtrl specifying the EthIf Controller where the IP address shall be assigned.		
Multiplicity	1		
Type	Reference to <a href="#">TcplpCtrl</a>		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME







	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

└

## [ECUC\_Tcplp\_00200] Definition of EcucReferenceDef TcplpLocalAddrIPv6ExtHeaderFilterRef

<b>Parameter Name</b>	TcplpLocalAddrIPv6ExtHeaderFilterRef		
<b>Parent Container</b>	<a href="#">TcplpLocalAddr</a>		
<b>Description</b>	Reference to a set of IPv6 Extension Headers which are allowed for this local IPv6 address. Note: this parameter is only relevant if the related TcplpDomainType is TCPIP_AF_INET6.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Reference to <a href="#">TcplpV6ConfigExtHeaderFilter</a>		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	dependency: only relevant if TcplpDomainType = TCPIP_AF_INET6		

└

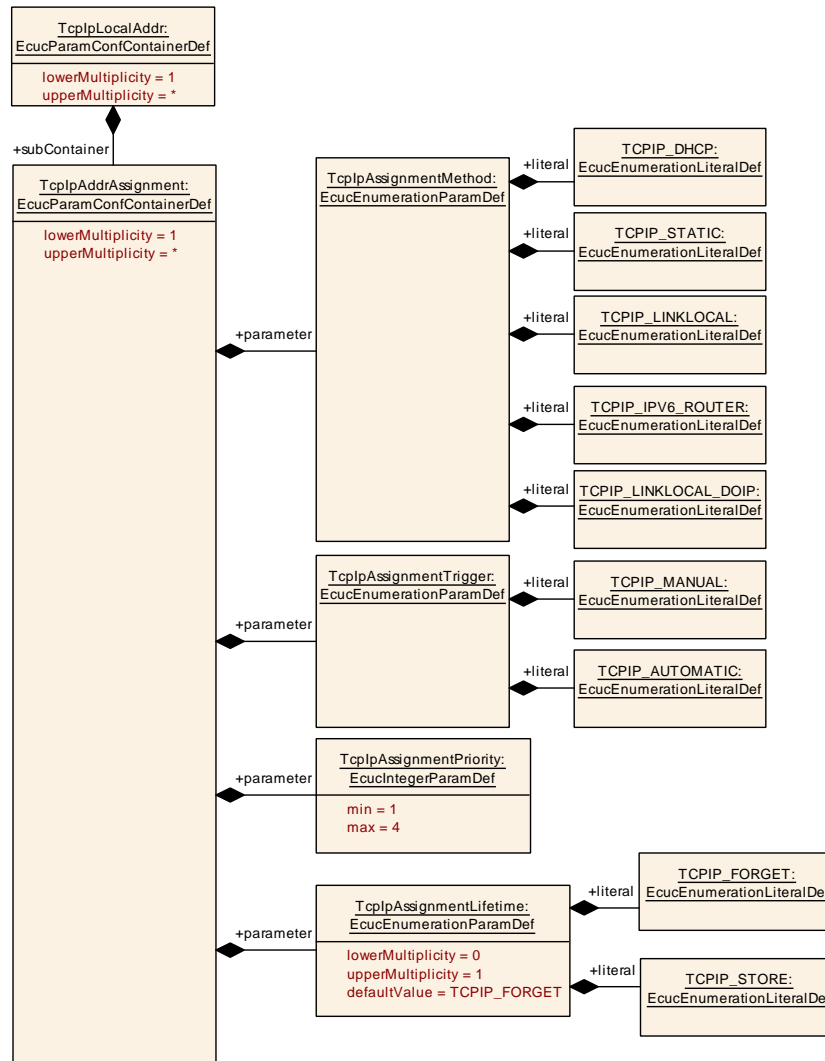


Figure 10.25: TcplpAddrAssignment

### 10.2.39 TcplpAddrAssignment

**[ECUC\_Tcplp\_00033] Definition of EcucParamConfContainerDef TcplpAddrAssignment**

Container Name	TcplpAddrAssignment
Parent Container	<a href="#">TcplpLocalAddr</a>
Description	This container is a subcontainer of TcplpLocalAddr and specifies the assignment policy for the IP address.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpAssignmentLifetime</a>	0..1	[ECUC_Tcplp_00186]
<a href="#">TcplpAssignmentMethod</a>	1	[ECUC_Tcplp_00035]
<a href="#">TcplpAssignmentPriority</a>	1	[ECUC_Tcplp_00037]
<a href="#">TcplpAssignmentTrigger</a>	1	[ECUC_Tcplp_00036]

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

]

### [ECUC\_Tcplp\_00186] Definition of EcucEnumerationParamDef TcplpAssignmentLifetime [

Parameter Name	TcplpAssignmentLifetime		
Parent Container	<a href="#">TcplpAddrAssignment</a>		
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	<a href="#">TCPIP_FORGET</a>		
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		

]

### [ECUC\_Tcplp\_00035] Definition of EcucEnumerationParamDef TcplpAssignmentMethod [

Parameter Name	TcplpAssignmentMethod		
Parent Container	<a href="#">TcplpAddrAssignment</a>		
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		

### [ECUC\_Tcplp\_00037] Definition of EcucIntegerParamDef TcplpAssignmentPriority

Parameter Name	TcplpAssignmentPriority		
Parent Container	<a href="#">TcplpAddrAssignment</a>		
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 .. 4		
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		

### [ECUC\_Tcplp\_00036] Definition of EcucEnumerationParamDef TcplpAssignmentTrigger

Parameter Name	TcplpAssignmentTrigger		
Parent Container	<a href="#">TcplpAddrAssignment</a>		
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
--------------------	--------------

## 10.2.40 TcplpStaticIpAddressConfig

### [ECUC\_Tcplp\_00034] Definition of EcucParamConfContainerDef TcplpStaticIpAddressConfig [

Container Name	TcplpStaticIpAddressConfig
Parent Container	<a href="#">TcplpLocalAddr</a>
Description	This container is a subcontainer of TcplpLocalAddr and specifies a static IP address including directly related parameters.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpDefaultRouter</a>	0..1	[ECUC_Tcplp_00040]
<a href="#">TcplpNetmask</a>	0..1	[ECUC_Tcplp_00039]
<a href="#">TcplpStaticIpAddress</a>	1	[ECUC_Tcplp_00038]

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

### [ECUC\_Tcplp\_00040] Definition of EcucStringParamDef TcplpDefaultRouter [

Parameter Name	TcplpDefaultRouter		
Parent Container	<a href="#">TcplpStaticIpAddressConfig</a>		
Description	IP address of default router (gateway)		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		
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
--------------------	--------------

### [ECUC\_Tcplp\_00039] Definition of EcucIntegerParamDef TcplpNetmask [

Parameter Name	TcplpNetmask		
Parent Container	<a href="#">TcplpStaticIpAddressConfig</a>		
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		

### [ECUC\_Tcplp\_00038] Definition of EcucStringParamDef TcplpStaticIpAddress [

Parameter Name	TcplpStaticIpAddress		
Parent Container	<a href="#">TcplpStaticIpAddressConfig</a>		
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	–		
Regular Expression	–		
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		

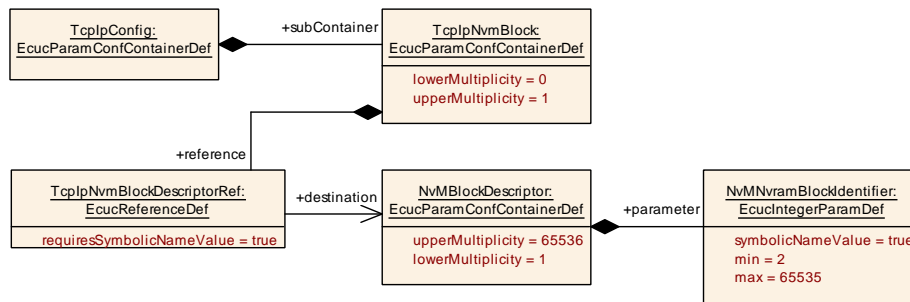


Figure 10.26: TcpIpNvm

### 10.2.41 TcpIpNvmBlock

#### [ECUC\_Tcplp\_00184] Definition of EcucParamConfContainerDef TcpIpNvmBlock

Container Name	TcpIpNvmBlock		
Parent Container	TcpIpConfig		
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			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
TcpIpNvmBlockDescriptorRef	1	[ECUC_Tcplp_00185]

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

#### [ECUC\_Tcplp\_00185] Definition of EcucReferenceDef TcpIpNvmBlockDescriptorRef

Parameter Name	TcpIpNvmBlockDescriptorRef		
Parent Container	TcpIpNvmBlock		
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		

]

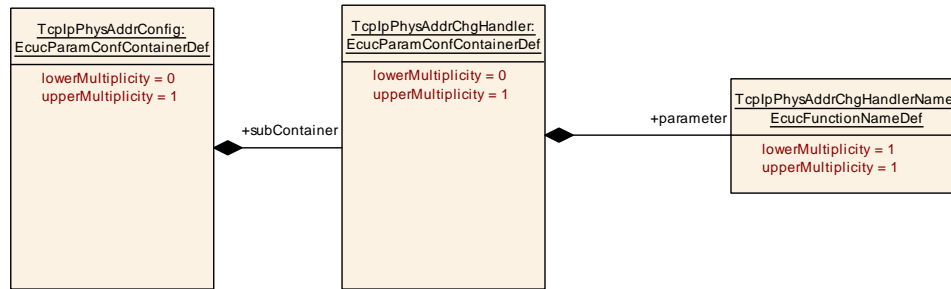


Figure 10.27: TcplpPhysAddrConfig

## 10.2.42 TcplpPhysAddrConfig

### [ECUC\_Tcplp\_00083] Definition of EcucParamConfContainerDef TcplpPhysAddr Config

Container Name	TcplpPhysAddrConfig
Parent Container	<a href="#">TcplpConfig</a>
Description	Specifies the physical address configuration.
Configuration Parameters	

#### No Included Parameters

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpPhysAddrChgHandler</a>	0..1	This container is a subcontainer of TcplpPhysAddrConfig and specifies the configuration parameters for physical address change handler.

]

## 10.2.43 TcplpPhysAddrChgHandler

### [ECUC\_Tcplp\_00084] Definition of EcucParamConfContainerDef TcplpPhysAddr ChgHandler



<b>Container Name</b>	TcplpPhysAddrChgHandler
<b>Parent Container</b>	<a href="#">TcplpPhysAddrConfig</a>
<b>Description</b>	This container is a subcontainer of TcplpPhysAddrConfig and specifies the configuration parameters for physical address change handler.
<b>Configuration Parameters</b>	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpPhysAddrChgHandlerName</a>	1	[ECUC_Tcplp_00086]

<b>No Included Containers</b>
-------------------------------

]

## [ECUC\_Tcplp\_00086] Definition of EcucFunctionNameDef TcplpPhysAddrChgHandlerName [

<b>Parameter Name</b>	TcplpPhysAddrChgHandlerName		
<b>Parent Container</b>	<a href="#">TcplpPhysAddrChgHandler</a>		
<b>Description</b>	This parameter defines the name of the physical address change function <Up>_PhysAddrTableChg.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default value</b>	–		
<b>Regular Expression</b>	–		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: ECU		

]

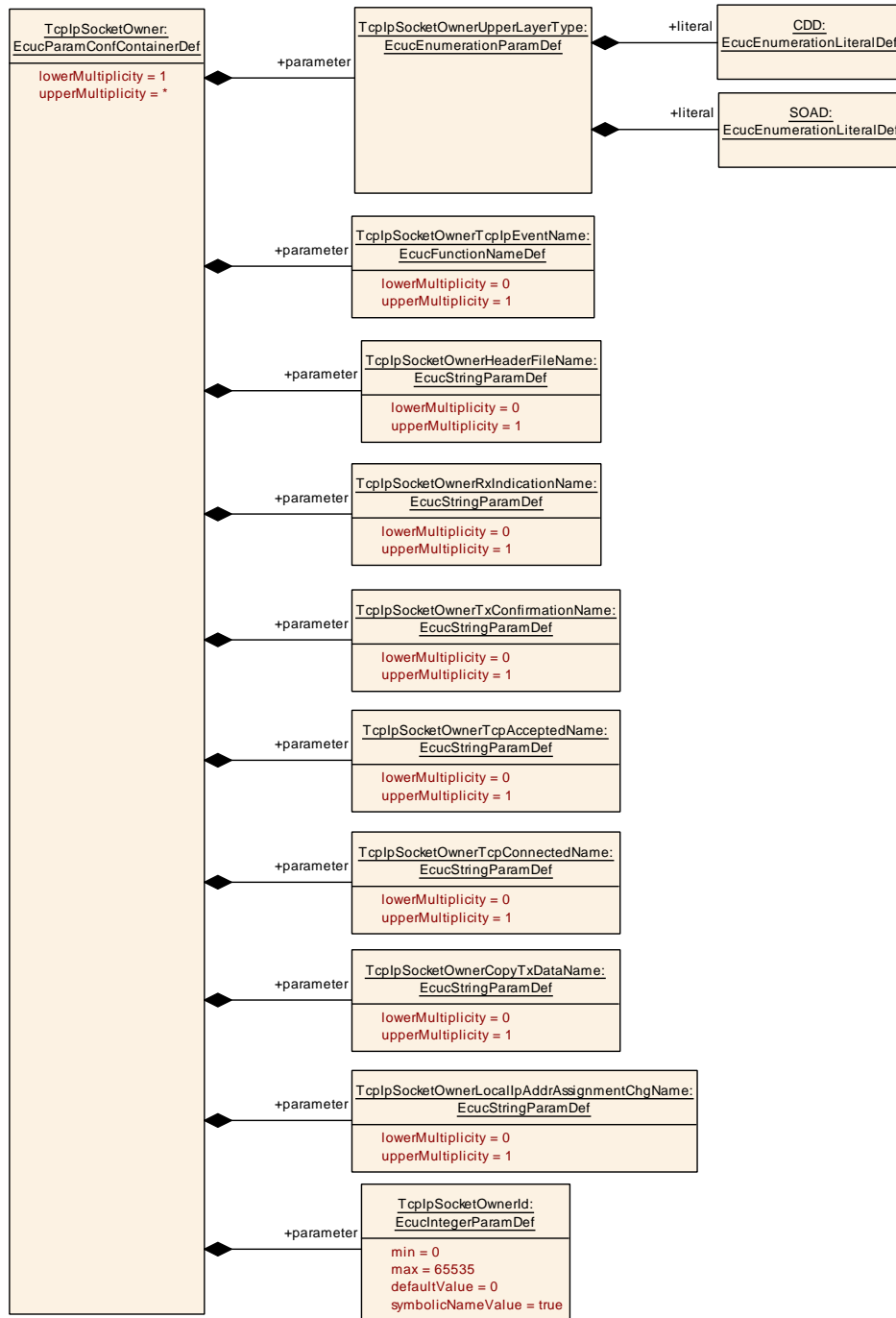


Figure 10.28: TcpIpSocketOwnerConfig

#### 10.2.44 TcpIpSocketOwnerConfig

[ECUC\_TcpIp\_00172] Definition of EcucParamConfContainerDef TcpIpSocket OwnerConfig

<b>Container Name</b>	TcplpSocketOwnerConfig
<b>Parent Container</b>	<a href="#">TcplpConfig</a>
<b>Description</b>	Specifies the upper layer modules of Tcplp using the socket API.
<b>Configuration Parameters</b>	

No Included Parameters

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpSocketOwner</a>	1..*	This container is a subcontainer of TcplpSocketOwnerConfig and specifies an upper layer of Tcplp that uses the socket API.

]

## 10.2.45 TcplpSocketOwner

### [ECUC\_Tcplp\_00173] Definition of EcucParamConfContainerDef TcplpSocket Owner [

<b>Container Name</b>	TcplpSocketOwner
<b>Parent Container</b>	<a href="#">TcplpSocketOwnerConfig</a>
<b>Description</b>	This container is a subcontainer of TcplpSocketOwnerConfig and specifies an upper layer of Tcplp that uses the socket API.
<b>Configuration Parameters</b>	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpSocketOwnerCopyTxDataName</a>	0..1	[ECUC_Tcplp_00180]
<a href="#">TcplpSocketOwnerHeaderFileName</a>	0..1	[ECUC_Tcplp_00175]
<a href="#">TcplpSocketOwnerId</a>	1	[ECUC_Tcplp_00316]
<a href="#">TcplpSocketOwnerLocalIpAddrAssignmentChgName</a>	0..1	[ECUC_Tcplp_00181]
<a href="#">TcplpSocketOwnerRxIndicationName</a>	0..1	[ECUC_Tcplp_00176]
<a href="#">TcplpSocketOwnerTcpAcceptedName</a>	0..1	[ECUC_Tcplp_00178]
<a href="#">TcplpSocketOwnerTcpConnectedName</a>	0..1	[ECUC_Tcplp_00179]
<a href="#">TcplpSocketOwnerTcplpEventName</a>	0..1	[ECUC_Tcplp_00197]
<a href="#">TcplpSocketOwnerTxConfirmationName</a>	0..1	[ECUC_Tcplp_00177]
<a href="#">TcplpSocketOwnerUpperLayerType</a>	1	[ECUC_Tcplp_00174]

No Included Containers

]

## [ECUC\_Tcplp\_00180] Definition of EcucStringParamDef TcplpSocketOwnerCopyTxDataName

Parameter Name	TcplpSocketOwnerCopyTxDataName		
Parent Container	<a href="#">TcplpSocketOwner</a>		
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	–		
Regular Expression	–		
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		

## [ECUC\_Tcplp\_00175] Definition of EcucStringParamDef TcplpSocketOwnerHeaderFileName

Parameter Name	TcplpSocketOwnerHeaderFileName		
Parent Container	<a href="#">TcplpSocketOwner</a>		
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	–		
Regular Expression	–		
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		

## [ECUC\_Tcplp\_00316] Definition of EcucIntegerParamDef TcplpSocketOwnerId

Parameter Name	TcplpSocketOwnerId		
Parent Container	<a href="#">TcplpSocketOwner</a>		
Description	This value specifies the ID of the socket user.		





<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b>	0 .. 65535		
<b>Default value</b>	0		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

### [ECUC\_Tcplp\_00181] Definition of EcucStringParamDef TcplpSocketOwnerLocalIpAddrAssignmentChgName

<b>Parameter Name</b>	TcplpSocketOwnerLocalIpAddrAssignmentChgName		
<b>Parent Container</b>	<a href="#">TcplpSocketOwner</a>		
<b>Description</b>	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.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucStringParamDef		
<b>Default value</b>	–		
<b>Regular Expression</b>	–		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local dependency: TcplpSocketOwnerUpperLayerType		

### [ECUC\_Tcplp\_00176] Definition of EcucStringParamDef TcplpSocketOwnerRxIndicationName

<b>Parameter Name</b>	TcplpSocketOwnerRxIndicationName		
<b>Parent Container</b>	<a href="#">TcplpSocketOwner</a>		
<b>Description</b>	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.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucStringParamDef		
<b>Default value</b>	–		
<b>Regular Expression</b>	–		
<b>Post-Build Variant Value</b>	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		

### [ECUC\_TcIp\_00178] Definition of EcucStringParamDef TcpIpSocketOwnerTcpAcceptedName

Parameter Name	TcpIpSocketOwnerTcpAcceptedName		
Parent Container	<a href="#">TcpIpSocketOwner</a>		
Description	This parameter defines the name of the <Up_TcpAccepted> function of the TcpIpSocketOwner module. The function name shall only be configurable if TcpIpSocketOwnerUpperLayerType is set to CDD.		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		
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		

### [ECUC\_TcIp\_00179] Definition of EcucStringParamDef TcpIpSocketOwnerTcpConnectedName

Parameter Name	TcpIpSocketOwnerTcpConnectedName		
Parent Container	<a href="#">TcpIpSocketOwner</a>		
Description	This parameter defines the name of the <Up_TcpConnected> function of the TcpIpSocketOwner module. The function name shall only be configurable if TcpIpSocketOwnerUpperLayerType is set to CDD.		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD





	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local dependency: TcplpSocketOwnerUpperLayerType		

### [ECUC\_Tcplp\_00197] Definition of EcucFunctionNameDef TcplpSocketOwner TcplpEventName

<b>Parameter Name</b>	TcplpSocketOwnerTcplpEventName		
<b>Parent Container</b>	<a href="#">TcplpSocketOwner</a>		
<b>Description</b>	This parameter defines the name of the <Up_TcplpEvent> function of the TcplpSocket Owner module. The function name shall only be configurable if TcplpSocketOwner UpperLayerType is set to CDD.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default value</b>	–		
<b>Regular Expression</b>	–		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local dependency: TcplpSocketOwnerUpperLayerType		

### [ECUC\_Tcplp\_00177] Definition of EcucStringParamDef TcplpSocketOwnerTx ConfirmationName

<b>Parameter Name</b>	TcplpSocketOwnerTxConfirmationName		
<b>Parent Container</b>	<a href="#">TcplpSocketOwner</a>		
<b>Description</b>	This parameter defines the name of the <Up_TxConfirmation> function of the Tcplp SocketOwner module. The function name shall only be configurable if TcplpSocket OwnerUpperLayerType is set to CDD.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucStringParamDef		
<b>Default value</b>	–		
<b>Regular Expression</b>	–		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	–	





<b>Scope / Dependency</b>	scope: local dependency: TcpIpSocketOwnerUpperLayerType
---------------------------	--

]

## [ECUC\_Tcplp\_00174] Definition of EcucEnumerationParamDef TcpIpSocketOwnerUpperLayerType [

Parameter Name	TcpIpSocketOwnerUpperLayerType		
Parent Container	<a href="#">TcpIpSocketOwner</a>		
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, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local		

]



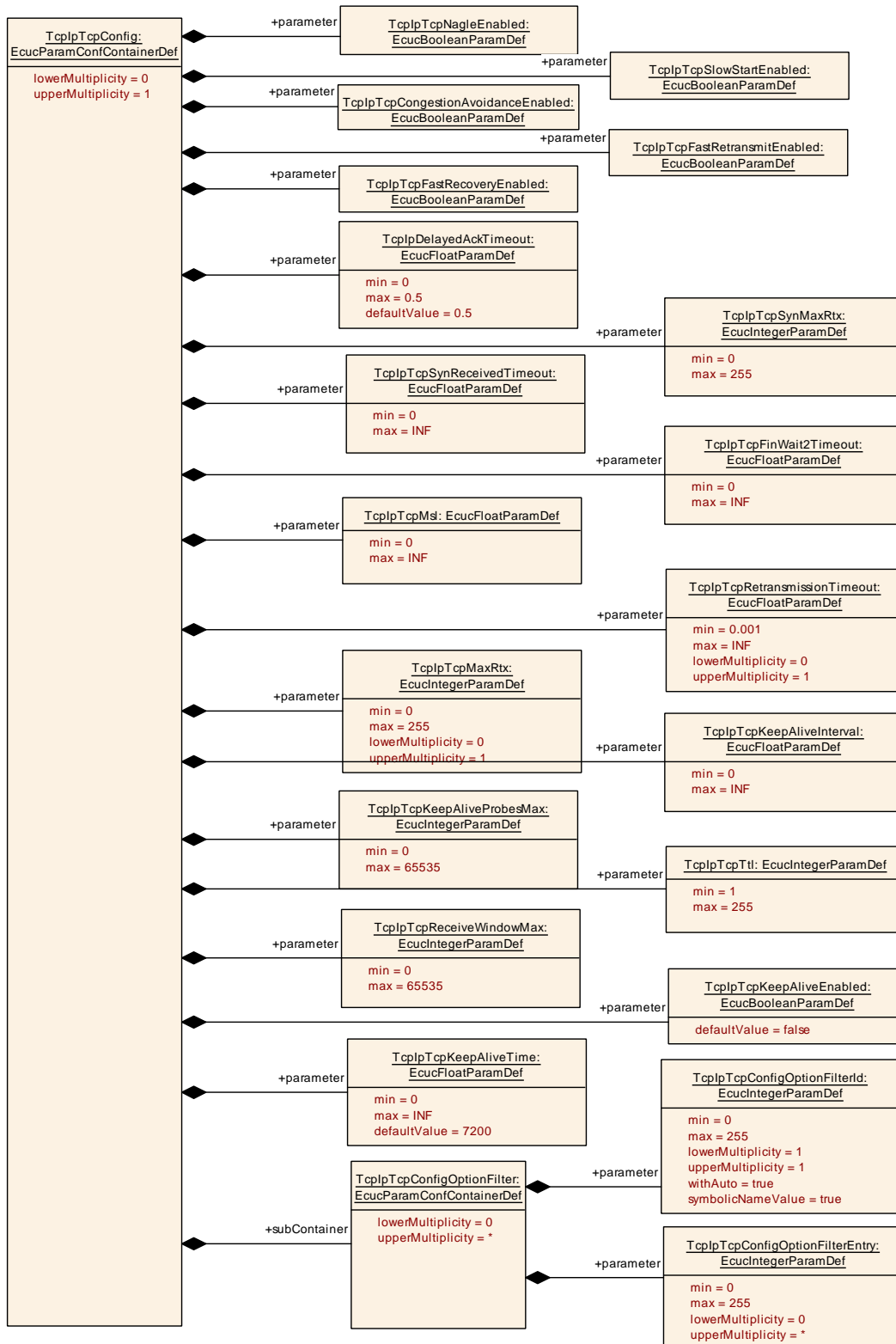


Figure 10.29: TcplpTcpConfig

## 10.2.46 TcplpTcpConfig

### [ECUC\_Tcplp\_00025] Definition of EcucParamConfContainerDef TcplpTcpConfig [

<b>Container Name</b>	TcplpTcpConfig
<b>Parent Container</b>	<a href="#">TcplpConfig</a>
<b>Description</b>	Specifies the configuration parameters of the TCP (Transmission Control Protocol) sub-module.
<b>Configuration Parameters</b>	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpDelayedAckTimeout</a>	1	[ECUC_Tcplp_00318]
<a href="#">TcplpTcpCongestionAvoidanceEnabled</a>	1	[ECUC_Tcplp_00061]
<a href="#">TcplpTcpFastRecoveryEnabled</a>	1	[ECUC_Tcplp_00063]
<a href="#">TcplpTcpFastRetransmitEnabled</a>	1	[ECUC_Tcplp_00062]
<a href="#">TcplpTcpFinWait2Timeout</a>	1	[ECUC_Tcplp_00066]
<a href="#">TcplpTcpKeepAliveEnabled</a>	1	[ECUC_Tcplp_00082]
<a href="#">TcplpTcpKeepAliveInterval</a>	1	[ECUC_Tcplp_00070]
<a href="#">TcplpTcpKeepAliveProbesMax</a>	1	[ECUC_Tcplp_00071]
<a href="#">TcplpTcpKeepAliveTime</a>	1	[ECUC_Tcplp_00087]
<a href="#">TcplpTcpMaxRetransmissionTimeout</a>	0..1	[ECUC_Tcplp_00340]
<a href="#">TcplpTcpMaxRtx</a>	0..1	[ECUC_Tcplp_00069]
<a href="#">TcplpTcpMsl</a>	1	[ECUC_Tcplp_00067]
<a href="#">TcplpTcpNagleEnabled</a>	1	[ECUC_Tcplp_00059]
<a href="#">TcplpTcpReceiveWindowMax</a>	1	[ECUC_Tcplp_00073]
<a href="#">TcplpTcpRetransmissionTimeout</a>	0..1	[ECUC_Tcplp_00068]
<a href="#">TcplpTcpSackEnabled</a>	1	[ECUC_Tcplp_00327]
<a href="#">TcplpTcpSlowStartEnabled</a>	1	[ECUC_Tcplp_00060]
<a href="#">TcplpTcpSynMaxRtx</a>	1	[ECUC_Tcplp_00064]
<a href="#">TcplpTcpSynReceivedTimeout</a>	1	[ECUC_Tcplp_00065]
<a href="#">TcplpTcpTtl</a>	1	[ECUC_Tcplp_00072]
<a href="#">TcplpTcpWindowScale</a>	1	[ECUC_Tcplp_00329]
<a href="#">TcplpTcpWindowScaleOptionEnabled</a>	1	[ECUC_Tcplp_00328]

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpTcpConfigOptionFilter</a>	0..*	This container describes the white list for the filtering of TCP options, i.e. segments containing TCP options not listed here shall be silently dropped.

]

## [ECUC\_Tcplp\_00318] Definition of EcucFloatParamDef TcplpDelayedAckTimeout

Parameter Name	TcplpDelayedAckTimeout		
Parent Container	<a href="#">TcplpTcpConfig</a>		
Description	The maximal time an acknowledgment is delayed for transmission in seconds. For further details, see also IETF RFC 1122 section 4.2.3.2.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	]0 .. 0.5]		
Default value	0.5		
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		

## [ECUC\_Tcplp\_00061] Definition of EcucBooleanParamDef TcplpTcpCongestionAvoidanceEnabled

Parameter Name	TcplpTcpCongestionAvoidanceEnabled		
Parent Container	<a href="#">TcplpTcpConfig</a>		
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		

## [ECUC\_Tcplp\_00063] Definition of EcucBooleanParamDef TcplpTcpFastRecoveryEnabled

Parameter Name	TcplpTcpFastRecoveryEnabled		
Parent Container	<a href="#">TcplpTcpConfig</a>		
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		

### [ECUC\_Tcplp\_00062] Definition of EcucBooleanParamDef TcplpTcpFastRetransmitEnabled

Parameter Name	TcplpTcpFastRetransmitEnabled		
Parent Container	<a href="#">TcplpTcpConfig</a>		
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		

### [ECUC\_Tcplp\_00066] Definition of EcucFloatParamDef TcplpTcpFinWait2Timeout

Parameter Name	TcplpTcpFinWait2Timeout		
Parent Container	<a href="#">TcplpTcpConfig</a>		
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		

## [ECUC\_Tcplp\_00082] Definition of EcucBooleanParamDef TcplpTcpKeepAlive Enabled

Parameter Name	TcplpTcpKeepAliveEnabled		
Parent Container	<a href="#">TcplpTcpConfig</a>		
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		

## [ECUC\_Tcplp\_00070] Definition of EcucFloatParamDef TcplpTcpKeepAliveInterval

Parameter Name	TcplpTcpKeepAliveInterval		
Parent Container	<a href="#">TcplpTcpConfig</a>		
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		

## [ECUC\_Tcplp\_00071] Definition of EcucIntegerParamDef TcplpTcpKeepAlive ProbesMax

Parameter Name	TcplpTcpKeepAliveProbesMax		
Parent Container	<a href="#">TcplpTcpConfig</a>		
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: TcpIpTcpKeepAliveEnabled		

### [ECUC\_Tcplp\_00087] Definition of EcucFloatParamDef TcpIpTcpKeepAliveTime

Parameter Name	TcpIpTcpKeepAliveTime		
Parent Container	<a href="#">TcpIpTcpConfig</a>		
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 TcpIpMainFunctionPeriod 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: TcpIpTcpKeepAliveEnabled		

### [ECUC\_Tcplp\_00340] Definition of EcucFloatParamDef TcpIpTcpMaxRetransmissionTimeout

Parameter Name	TcpIpTcpMaxRetransmissionTimeout		
Parent Container	<a href="#">TcpIpTcpConfig</a>		
Description	Maximum value (clamp) of clamped exponential backoff timeout in [s] before an unacknowledged TCP segment is sent again. If the timeout is disabled or set to INF, clamped exponential backoff shall not be used.		
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





	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local dependency: TcpIpTcpRetransmissionTimeout		

]

### [ECUC\_TcpIp\_00069] Definition of EcucIntegerParamDef TcpIpTcpMaxRtx [

<b>Parameter Name</b>	TcpIpTcpMaxRtx		
<b>Parent Container</b>	<a href="#">TcpIpTcpConfig</a>		
<b>Description</b>	Maximum number of times that a TCP segment is retransmitted before the TCP connection is closed. This parameter is only valid if TcpIpTcpRetransmissionTimeout/TcpIpTcpMaxRetransmissionTimeout is configured. Note: This parameter also applies for FIN retransmissions.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default value</b>	–		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	dependency: TcpIpTcpRetransmissionTimeout		

]

### [ECUC\_TcpIp\_00067] Definition of EcucFloatParamDef TcpIpTcpMsl [

<b>Parameter Name</b>	TcpIpTcpMsl		
<b>Parent Container</b>	<a href="#">TcpIpTcpConfig</a>		
<b>Description</b>	Maximum segment lifetime in [s]. (Note: TIME-WAIT = 2 x TcpIpTcpMsl - to ensure that the remote node received the acknowledgment to its connection termination request.)		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	[0 .. INF]		
<b>Default value</b>	–		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

]

## [ECUC\_Tcplp\_00059] Definition of EcucBooleanParamDef TcplpTcpNagleEnabled

Parameter Name	TcplpTcpNagleEnabled		
Parent Container	<a href="#">TcplpTcpConfig</a>		
Description	Enables (TRUE) or disables (FALSE) support of Nagle's algorithm according to IETF RFC 1122 (chapter 4.2.3.4 When to Send Data). 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		

## [ECUC\_Tcplp\_00073] Definition of EcucIntegerParamDef TcplpTcpReceiveWindowMax

Parameter Name	TcplpTcpReceiveWindowMax		
Parent Container	<a href="#">TcplpTcpConfig</a>		
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		



## [ECUC\_Tcplp\_00068] Definition of EcucFloatParamDef TcplpTcpRetransmissionTimeout

Parameter Name	TcplpTcpRetransmissionTimeout		
Parent Container	<a href="#">TcplpTcpConfig</a>		
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. Value can be overwritten by Tcplp_ChangeParameter() API for a particular connection. If TcplpTcpMaxRetransmissionTimeout is enabled then TcplpTcpRetransmissionTimeout or value overwritten by Tcplp_ChangeParameter() API is considered as initial value for first retransmission before the next valid acknowledgment arrives.		
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 dependency: TcplpTcpMaxRetransmissionTimeout		

## [ECUC\_Tcplp\_00327] Definition of EcucBooleanParamDef TcplpTcpSackEnabled

*Status:* DRAFT

Parameter Name	TcplpTcpSackEnabled		
Parent Container	<a href="#">TcplpTcpConfig</a>		
Description	Defines if the SACK (selective acknowledgement) mechanism shall be supported according to IETF RFC 2018. <b>Tags:</b> atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

## [ECUC\_Tcplp\_00060] Definition of EcucBooleanParamDef TcplpTcpSlowStart Enabled

Parameter Name	TcplpTcpSlowStartEnabled		
Parent Container	<a href="#">TcplpTcpConfig</a>		
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		

## [ECUC\_Tcplp\_00064] Definition of EcucIntegerParamDef TcplpTcpSynMaxRtx

Parameter Name	TcplpTcpSynMaxRtx		
Parent Container	<a href="#">TcplpTcpConfig</a>		
Description	Maximum number of times that a TCP SYN is retransmitted. Note: SYN will be retried after TcplpTcpRetransmissionTimeout/TcplpTcpMaxRetransmissionTimeout. The connection will be dropped if no matching connection request has been received after the last TCP SYN has been sent and TcplpTcpRetransmissionTimeout/TcplpTcpMaxRetransmissionTimeout 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		

## [ECUC\_Tcplp\_00065] Definition of EcucFloatParamDef TcplpTcpSynReceived Timeout

Parameter Name	TcplpTcpSynReceivedTimeout		
Parent Container	<a href="#">TcplpTcpConfig</a>		
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		

]

### [ECUC\_Tcplp\_00072] Definition of EcucIntegerParamDef TcplpTcpTtl [

Parameter Name	TcplpTcpTtl		
Parent Container	<a href="#">TcplpTcpConfig</a>		
Description	Default Time-to-live value of outgoing TCP packets.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 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		

]

### [ECUC\_Tcplp\_00329] Definition of EcucIntegerParamDef TcplpTcpWindowScale

Status: DRAFT

[

Parameter Name	TcplpTcpWindowScale		
Parent Container	<a href="#">TcplpTcpConfig</a>		
Description	Defines the TCP window scale. <b>Tags:</b> atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 14		
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
--------------------	--------------

]

## [ECUC\_Tcplp\_00328] Definition of EcucBooleanParamDef TcplpTcpWindowScaleOptionEnabled

Status: DRAFT

[

Parameter Name	TcplpTcpWindowScaleOptionEnabled		
Parent Container	<a href="#">TcplpTcpConfig</a>		
Description	Defines if the TCP window scale option and mechanism shall be supported according to IETF RFC 7323, chapter 2. <b>Tags:</b> atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

## 10.2.47 TcplpTcpConfigOptionFilter

## [ECUC\_Tcplp\_00202] Definition of EcucParamConfContainerDef TcplpTcpConfigOptionFilter [

Container Name	TcplpTcpConfigOptionFilter		
Parent Container	<a href="#">TcplpTcpConfig</a>		
Description	This container describes the white list for the filtering of TCP options, i.e. segments containing TCP options not listed here shall be silently dropped.		
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			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpTcpConfigOptionFilterEntry</a>	0..*	[ECUC_Tcplp_00204]
<a href="#">TcplpTcpConfigOptionFilterId</a>	1	[ECUC_Tcplp_00203]

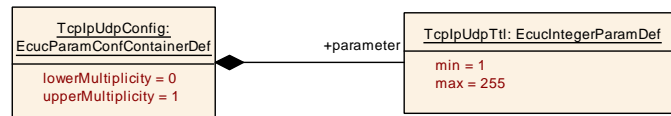
No Included Containers

## [ECUC\_Tcplp\_00204] Definition of EcucIntegerParamDef TcplpTcpConfigOptionFilterEntry

Parameter Name	TcplpTcpConfigOptionFilterEntry		
Parent Container	<a href="#">TcplpTcpConfigOptionFilter</a>		
Description	TCP option kind allowed by this filter.		
Multiplicity	0..*		
Type	EcucIntegerParamDef		
Range	0 .. 255		
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		

## [ECUC\_Tcplp\_00203] Definition of EcucIntegerParamDef TcplpTcpConfigOptionFilterId

Parameter Name	TcplpTcpConfigOptionFilterId		
Parent Container	<a href="#">TcplpTcpConfigOptionFilter</a>		
Description	Identification of the TCP option filter.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
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 withAuto = true		



**Figure 10.30: TcpIpUdpConfig**

## 10.2.48 TcpIpUdpConfig

**[ECUC\_Tcplp\_00026] Definition of EcucParamConfContainerDef TcpIpUdpConfig** [

Container Name	TcpIpUdpConfig
Parent Container	<a href="#">TcpIpConfig</a>
Description	Specifies the configuration parameters of the UDP (User Datagram Protocol) sub-module
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcpIpUdpTtl</a>	1	[ <a href="#">ECUC_Tcplp_00075</a> ]

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

]

**[ECUC\_Tcplp\_00075] Definition of EcucIntegerParamDef TcpIpUdpTtl** [

Parameter Name	TcpIpUdpTtl		
Parent Container	<a href="#">TcpIpUdpConfig</a>		
Description	Default Time-to-live value of outgoing UDP packets.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 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		

]

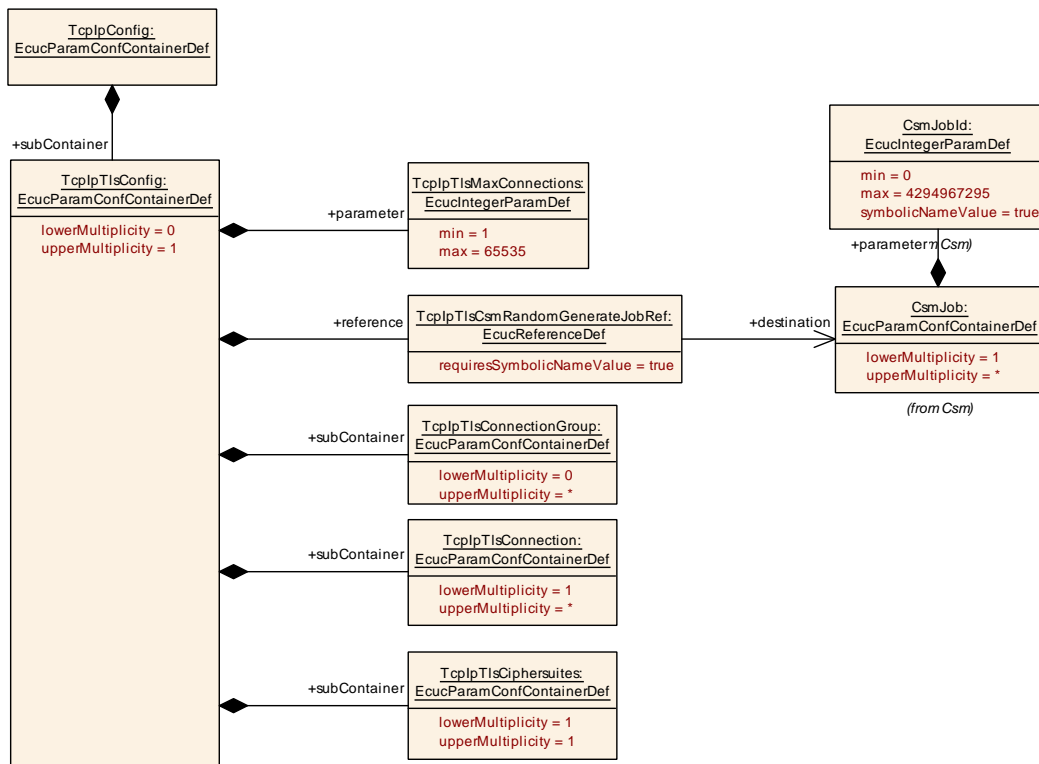


Figure 10.31: TcplpTlsConfig

## 10.2.49 TcplpTlsConfig

### [ECUC\_Tcplp\_00219] Definition of EcucParamConfContainerDef TcplpTlsConfig

Container Name	TcplpTlsConfig		
Parent Container	<a href="#">TcplpConfig</a>		
Description	Specifies the configuration parameters of the TLS (Transport Layer Security) sub module.		
Configuration Parameters			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpTlsMaxConnections</a>	1	[ECUC_Tcplp_00220]
<a href="#">TcplpTlsCsmRandomGenerateJobRef</a>	1	[ECUC_Tcplp_00221]

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpTlsCiphersuites</a>	1	This container provides the information about supported ciphersuites used by TLS.
<a href="#">TcplpTlsConnection</a>	1..*	This container defines the properties of a TLS connection





Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpTlsConnectionGroup</a>	0..*	This optional container is used to collect all TlsConnections that belong to a TlsConnectionGroup. The intention of a TLS connection group is to share resources among TLS connections collected in a group, because only one connection of a group can be used at a time.

## [ECUC\_Tcplp\_00220] Definition of EcucIntegerParamDef TcplpTlsMaxConnections

Parameter Name	TcplpTlsMaxConnections		
Parent Container	<a href="#">TcplpTlsConfig</a>		
Description	Defines the max. number of TLS connections that can be opened at the same time.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 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: local		

## [ECUC\_Tcplp\_00221] Definition of EcucReferenceDef TcplpTlsCsmRandomGenerateJobRef

Parameter Name	TcplpTlsCsmRandomGenerateJobRef		
Parent Container	<a href="#">TcplpTlsConfig</a>		
Description	Reference to a CSM job to generate a random value.		
Multiplicity	1		
Type	Symbolic name reference to CsmJob		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		



## 10.2.50 TcplpTlsConnectionGroup

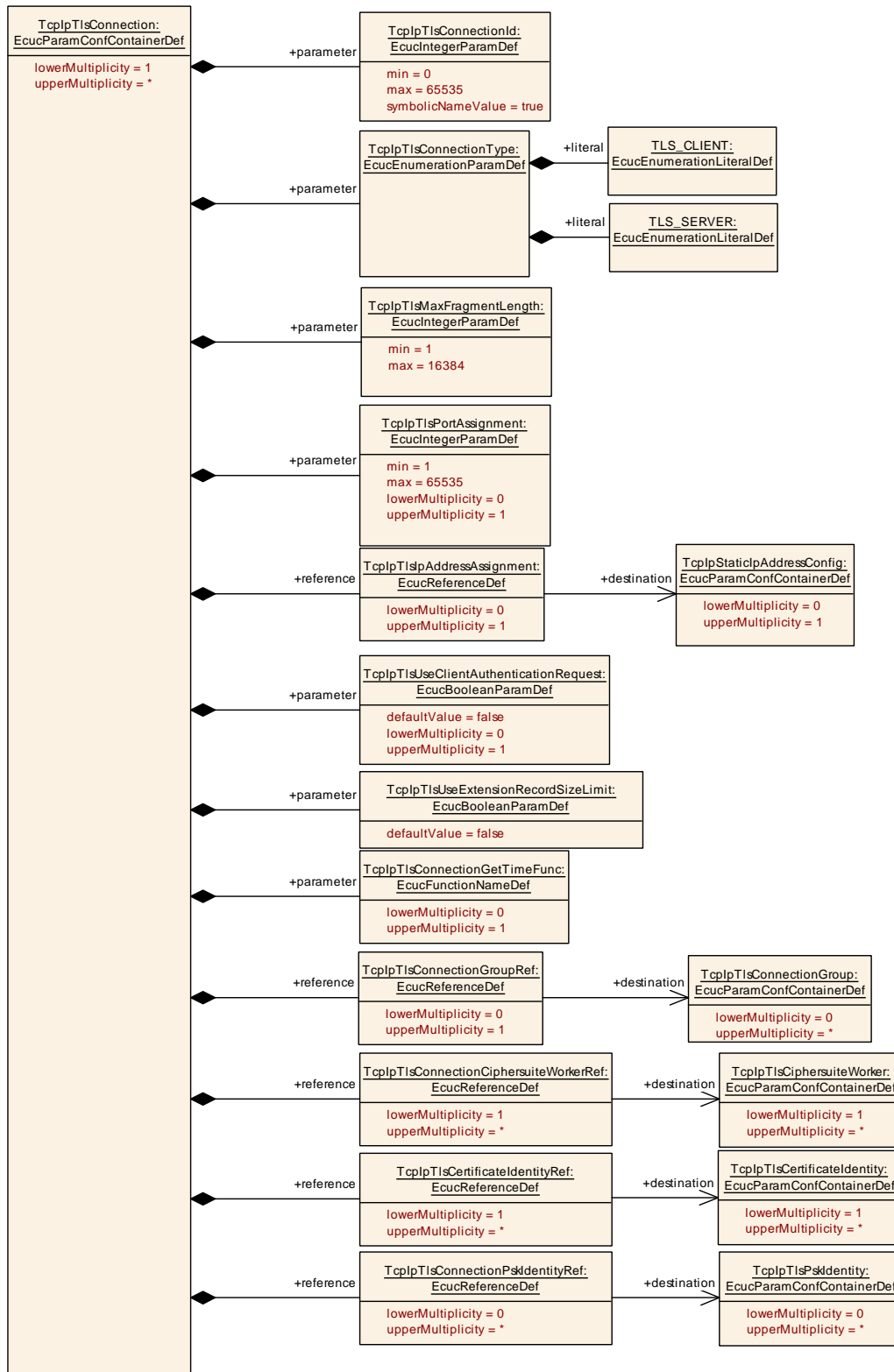
**[ECUC\_Tcplp\_00224] Definition of EcucParamConfContainerDef TcplpTlsConnectionGroup** [

<b>Container Name</b>	TcplpTlsConnectionGroup
<b>Parent Container</b>	<a href="#">TcplpTlsConfig</a>
<b>Description</b>	This optional container is used to collect all TlsConnections that belong to a Tls ConnectionGroup. The intention of a TLS connection group is to share resources among TLS connections collected in a group, because only one connection of a group can be used at a time.
<b>Configuration Parameters</b>	

**No Included Parameters**

**No Included Containers**

]



**Figure 10.32: TcpIpTlsConnection**

## 10.2.51 TcplpTlsConnection

**[ECUC\_Tcplp\_00223] Definition of EcucParamConfContainerDef TcplpTlsConnection**

<b>Container Name</b>	TcplpTlsConnection
<b>Parent Container</b>	<a href="#">TcplpTlsConfig</a>
<b>Description</b>	This container defines the properties of a TLS connection
<b>Configuration Parameters</b>	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpTlsConnectionGetTimeFunc</a>	0..1	[ECUC_Tcplp_00232]
<a href="#">TcplpTlsConnectionId</a>	1	[ECUC_Tcplp_00225]
<a href="#">TcplpTlsConnectionType</a>	1	[ECUC_Tcplp_00226]
<a href="#">TcplpTlsMaxFragmentLength</a>	1	[ECUC_Tcplp_00227]
<a href="#">TcplpTlsPortAssignment</a>	0..1	[ECUC_Tcplp_00285]
<a href="#">TcplpTlsUseClientAuthenticationRequest</a>	0..1	[ECUC_Tcplp_00230]
<a href="#">TcplpTlsUseExtensionCertificateStatusRequest</a>	1	[ECUC_Tcplp_00334]
<a href="#">TcplpTlsUseExtensionMaxFragmentLength</a>	1	[ECUC_Tcplp_00332]
<a href="#">TcplpTlsUseExtensionRecordSizeLimit</a>	1	[ECUC_Tcplp_00231]
<a href="#">TcplpTlsUseExtensionTrustedCAKeys</a>	1	[ECUC_Tcplp_00333]
<a href="#">TcplpTlsCertificateIdentityRef</a>	1..*	[ECUC_Tcplp_00235]
<a href="#">TcplpTlsConnectionCiphersuiteWorkerRef</a>	1..*	[ECUC_Tcplp_00234]
<a href="#">TcplpTlsConnectionGroupRef</a>	0..1	[ECUC_Tcplp_00233]
<a href="#">TcplpTlsConnectionPskIdentityRef</a>	0..*	[ECUC_Tcplp_00236]
<a href="#">TcplpTlsIpAddressAssignment</a>	0..1	[ECUC_Tcplp_00229]

No Included Containers

**[ECUC\_Tcplp\_00232] Definition of EcucFunctionNameDef TcplpTlsConnectionGetTimeFunc**

<b>Parameter Name</b>	TcplpTlsConnectionGetTimeFunc		
<b>Parent Container</b>	<a href="#">TcplpTlsConnection</a>		
<b>Description</b>	Defines the function name for the Up_TlsGetCurrentTimeStamp() callback.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default value</b>	–		
<b>Regular Expression</b>	–		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants





Value Configuration Class	Link time	–	
	Post-build time	–	
	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: This definition is needed if a connection specific time shall be provided with the client hello message. If not present, the time will be set to 0.		

### [ECUC\_Tcplp\_00225] Definition of EcucIntegerParamDef TcplpTlsConnectionId

Parameter Name	TcplpTlsConnectionId		
Parent Container	<a href="#">TcplpTlsConnection</a>		
Description	Identifier of the connection. The set of configured identifiers shall be consecutive and gapless.		
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: local		

### [ECUC\_Tcplp\_00226] Definition of EcucEnumerationParamDef TcplpTlsConnectionType

Parameter Name	TcplpTlsConnectionType		
Parent Container	<a href="#">TcplpTlsConnection</a>		
Description	Specifies if the TLS connection is a server or a client.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	TLS_CLIENT	–	
	TLS_SERVER	–	
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

## [ECUC\_Tcplp\_00227] Definition of EcucIntegerParamDef TcplpTlsMaxFragmentLength

Parameter Name	TcplpTlsMaxFragmentLength		
Parent Container	<a href="#">TcplpTlsConnection</a>		
Description	Specifies the max length in bytes of a TLS fragment that is sent as a block. If ISO 15118-2 shall be supported, the range is 512 .. 16384.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 16384		
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		

## [ECUC\_Tcplp\_00285] Definition of EcucIntegerParamDef TcplpTlsPortAssignment

Parameter Name	TcplpTlsPortAssignment		
Parent Container	<a href="#">TcplpTlsConnection</a>		
Description	Specifies the port address that is used for TLS communication.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 65535		
Default value	–		
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		

## [ECUC\_Tcplp\_00230] Definition of EcucBooleanParamDef TcplpTlsUseClientAuthenticationRequest [

Parameter Name	TcplpTlsUseClientAuthenticationRequest		
Parent Container	<a href="#">TcplpTlsConnection</a>		
Description	Defines if client authentication shall be applied for this TLS connection.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
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	dependency: Informs the TLS_SERVER that a client authentication shall be requested. Can be omitted on TLS_CLIENT side.		

]

## [ECUC\_Tcplp\_00334] Definition of EcucBooleanParamDef TcplpTlsUseExtensionCertificateStatusRequest

Status: DRAFT

[

Parameter Name	TcplpTlsUseExtensionCertificateStatusRequest		
Parent Container	<a href="#">TcplpTlsConnection</a>		
Description	Defines if the optional extension status_request_v2 shall be supported. <b>Tags:</b> atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

## [ECUC\_Tcplp\_00332] Definition of EcucBooleanParamDef TcplpTlsUseExtensionMaxFragmentLength

Status: DRAFT

[

Parameter Name	TcplpTlsUseExtensionMaxFragmentLength		
Parent Container	<a href="#">TcplpTlsConnection</a>		
Description	Defines if the optional extension for max_fragment_length shall be supported. <b>Tags:</b> atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local dependency: TcplpTlsUseExtensionRecordSizeLimit is set to FALSE		

]

## [ECUC\_Tcplp\_00231] Definition of EcucBooleanParamDef TcplpTlsUseExtensionRecordSizeLimit

[

Parameter Name	TcplpTlsUseExtensionRecordSizeLimit		
Parent Container	<a href="#">TcplpTlsConnection</a>		
Description	Defines if the security extension for record_size_limit shall be supported as defined in IETF RFC 8449, chapter 4.1.		
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		

]

## [ECUC\_Tcplp\_00333] Definition of EcucBooleanParamDef TcplpTlsUseExtensionTrustedCAKeys

Status: DRAFT

[

Parameter Name	TcplpTlsUseExtensionTrustedCAKeys		
Parent Container	<a href="#">TcplpTlsConnection</a>		
Description	Defines if the optional extension for trusted_ca_keys shall be supported <b>Tags:</b> atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

## [ECUC\_Tcplp\_00235] Definition of EcucReferenceDef TcplpTlsCertificateIdentityRef

[

Parameter Name	TcplpTlsCertificateIdentityRef		
Parent Container	<a href="#">TcplpTlsConnection</a>		
Description	References the container that contains the certificate and identity information.		
Multiplicity	1..*		
Type	Reference to <a href="#">TcplpTlsCertificateIdentity</a>		
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		
	dependency: There shall be only one TlsCertificateIdentity reference if server name identification is not used.		

]



## [ECUC\_Tcplp\_00234] Definition of EcucReferenceDef TcplpTlsConnectionCiphersuiteWorkerRef

Parameter Name	TcplpTlsConnectionCiphersuiteWorkerRef		
Parent Container	<a href="#">TcplpTlsConnection</a>		
Description	References the container that contains the jobs and keys to process the application data.		
Multiplicity	1..*		
Type	Reference to <a href="#">TcplpTlsCiphersuiteWorker</a>		
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		

## [ECUC\_Tcplp\_00233] Definition of EcucReferenceDef TcplpTlsConnectionGroupRef

Parameter Name	TcplpTlsConnectionGroupRef		
Parent Container	<a href="#">TcplpTlsConnection</a>		
Description	Assigns the TLS connection to a connection group.		
Multiplicity	0..1		
Type	Reference to <a href="#">TcplpTlsConnectionGroup</a>		
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		

## [ECUC\_Tcplp\_00236] Definition of EcucReferenceDef TcplpTlsConnectionPskIdentityRef

Parameter Name	TcplpTlsConnectionPskIdentityRef		
Parent Container	<a href="#">TcplpTlsConnection</a>		
Description	References the container that contains information about pre-shared keys.		
Multiplicity	0..*		
Type	Reference to <a href="#">TcplpTlsPskIdentity</a>		
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  dependency: A reference to PskIdentity container is only useful if at least one CiphersuiteDefinition is referenced offering a PSK ciphersuite. Multiplicity might be reduced to 1 to provide a unique PSK identification depending on the TLS protocol version and/or if it is used for the TLS server or client.		

## [ECUC\_Tcplp\_00229] Definition of EcucReferenceDef TcplpTlsIpAddressAssignment

Parameter Name	TcplpTlsIpAddressAssignment		
Parent Container	<a href="#">TcplpTlsConnection</a>		
Description	Contains additional information about the endpoint IP address information. If this reference is present, the IP address of the connecting socket shall also be checked if a TLS connection shall be assigned automatically to a socket.		
Multiplicity	0..1		
Type	Reference to <a href="#">TcplpStaticIpAddressConfig</a>		
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  dependency: If this item is not present but TcplpTlsPortAssignment is defined, then IP address information is not relevant for the TLS connection assignment. If TcplpTlsPort Assignment is not defined this item has no affect and shall not be defined, too.		

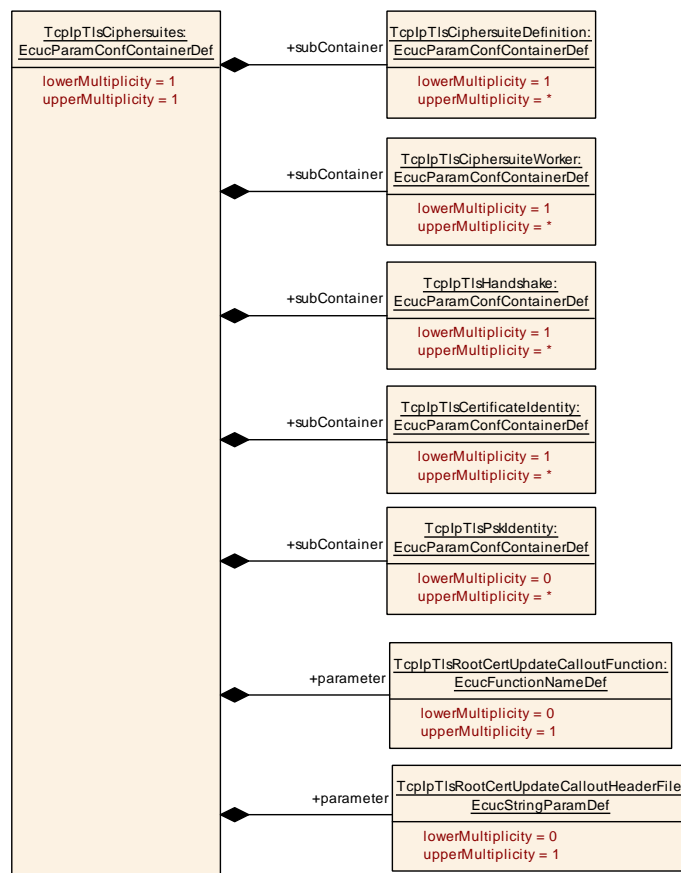


Figure 10.33: TcplpTlsCiphersuites

## 10.2.52 TcplpTlsCiphersuites

[ECUC\_Tcplp\_00222] Definition of EcucParamConfContainerDef TcplpTlsCiphersuites [

Container Name	TcplpTlsCiphersuites
Parent Container	<a href="#">TcplpTlsConfig</a>
Description	This container provides the information about supported ciphersuites used by TLS.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpTlsRootCertUpdateCalloutFunction</a>	0..1	[ <a href="#">ECUC_Tcplp_00330</a> ]
<a href="#">TcplpTlsRootCertUpdateCalloutHeaderFile</a>	0..1	[ <a href="#">ECUC_Tcplp_00331</a> ]

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpTlsCertificateIdentity</a>	1..*	This container provides information about the certificates used for ciphersuites.
<a href="#">TcplpTlsCiphersuiteDefinition</a>	1..*	This container provides the static information of a ciphersuite used by TLS.
<a href="#">TcplpTlsCiphersuiteWorker</a>	1..*	This container provides the jobs and keys necessary for TLS data transmission and reception.
<a href="#">TcplpTlsHandshake</a>	1..*	This container provides information that is needed to process a handshake. It contains the appropriate references to jobs and keys of the CSM to perform the key exchange cryptographic for the ciphersuite and involved certificates.
<a href="#">TcplpTlsPskIdentity</a>	0..*	This container provides information about static definition of pre-shared keys. It is used during the handshake to negotiate pre-shared keys between a client and a server. Note: The callbacks for pre-shared keys are an alternative to the static definition. The callbacks allow to define the associated keys at runtime if pre-shared keys are used but no static definition is available. The container definition is used for static configuration.

## [ECUC\_Tcplp\_00330] Definition of EcucFunctionNameDef TcplpTlsRootCertUpdateCalloutFunction

Status: DRAFT

Parameter Name	TcplpTlsRootCertUpdateCalloutFunction		
Parent Container	<a href="#">TcplpTlsCiphersuites</a>		
Description	<p>This optional parameter specifies the name of a callout function that is used when a new, valid, root certificate is received during a TLS handshake. Can be used to perform actions based on the received certificate.</p> <p><b>Tags:</b> atp.Status=draft</p>		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default value	–		
Regular Expression	–		
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		

## [ECUC\_Tcplp\_00331] Definition of EcucStringParamDef TcplpTlsRootCertUpdateCalloutHeaderFile

Status: DRAFT

[

Parameter Name	TcplpTlsRootCertUpdateCalloutHeaderFile		
Parent Container	<a href="#">TcplpTlsCiphersuites</a>		
Description	This optional parameter specifies the name of the header file containing the definition for the function specified in TcplpTlsRootCertUpdateCalloutFunction. <b>Tags:</b> atp.Status=draft		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		
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		

]

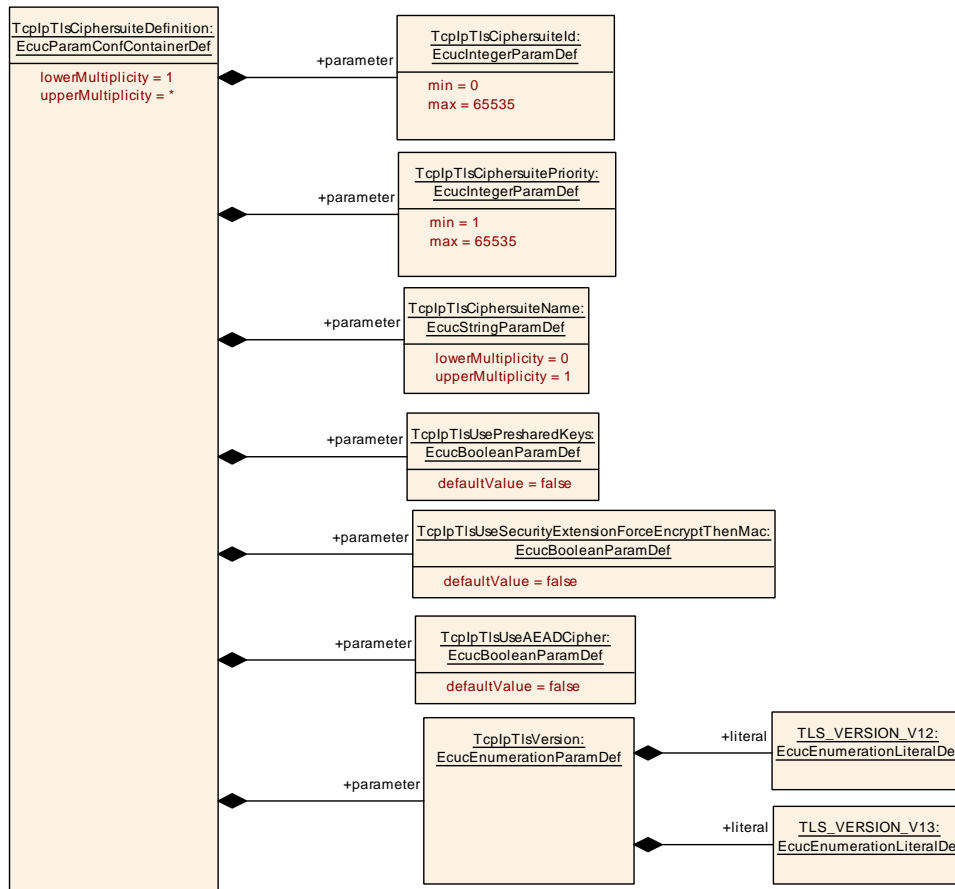


Figure 10.34: TcplpTlsCiphersuiteDefinition

### 10.2.53 TcplpTlsCiphersuiteDefinition

[ECUC\_Tcplp\_00237] Definition of EcucParamConfContainerDef TcplpTlsCiphersuiteDefinition [

Container Name	TcplpTlsCiphersuiteDefinition
Parent Container	<a href="#">TcplpTlsCiphersuites</a>
Description	This container provides the static information of a ciphersuite used by TLS.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpTlsCiphersuiteId</a>	1	[ECUC_Tcplp_00242]
<a href="#">TcplpTlsCiphersuiteName</a>	0..1	[ECUC_Tcplp_00244]
<a href="#">TcplpTlsCiphersuitePriority</a>	1	[ECUC_Tcplp_00243]
<a href="#">TcplpTlsUseAEADCipher</a>	1	[ECUC_Tcplp_00247]
<a href="#">TcplpTlsUsePresharedKeys</a>	1	[ECUC_Tcplp_00245]





Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpTlsUseSecurityExtensionForceEncryptThenMac</a>	1	<a href="#">[ECUC_Tcplp_00246]</a>
<a href="#">TcplpTlsVersion</a>	1	<a href="#">[ECUC_Tcplp_00248]</a>

No Included Containers

└

### [ECUC\_Tcplp\_00242] Definition of EcucIntegerParamDef TcplpTlsCiphersuiteld

└

Parameter Name	TcplpTlsCiphersuiteld		
Parent Container	<a href="#">TcplpTlsCiphersuiteDefinition</a>		
Description	ID that represents the ciphersuite according to IETF, e.g. RFC4492, Sect. 6, RFC8446, Appendix B.4 or RFC5246, Appendix A.5.		
Multiplicity	1		
Type	EcucIntegerParamDef		
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: local		

└

### [ECUC\_Tcplp\_00244] Definition of EcucStringParamDef TcplpTlsCiphersuite Name

└

Parameter Name	TcplpTlsCiphersuiteName		
Parent Container	<a href="#">TcplpTlsCiphersuiteDefinition</a>		
Description	Provides a verbal name for the ciphersuite. The name should be the one defined in the respective RFC, e.g. TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (TLS 1.2) or TLS_AES_128_GCM_SHA256 (TLS 1.3)		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

└

### [ECUC\_Tcplp\_00243] Definition of EcucIntegerParamDef TcplpTlsCiphersuite Priority

Parameter Name	TcplpTlsCiphersuitePriority		
Parent Container	<a href="#">TcplpTlsCiphersuiteDefinition</a>		
Description	Defines the priority of the cipher. The higher the number the lower the priority.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 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: local		

### [ECUC\_Tcplp\_00247] Definition of EcucBooleanParamDef TcplpTlsUseAEADCipher

Parameter Name	TcplpTlsUseAEADCipher		
Parent Container	<a href="#">TcplpTlsCiphersuiteDefinition</a>		
Description	Specifies if the ciphersuite supports AEAD for data en-/decryption.		
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		

### [ECUC\_Tcplp\_00245] Definition of EcucBooleanParamDef TcplpTlsUsePre-sharedKeys

Parameter Name	TcplpTlsUsePresharedKeys		
Parent Container	<a href="#">TcplpTlsCiphersuiteDefinition</a>		
Description	Defines if this ciphersuite uses pre-shared keys. If so, additional configuration or callbacks will be used for pre-shared key negotiation.		
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		

]

### [ECUC\_Tcplp\_00246] Definition of EcucBooleanParamDef TcplpTlsUseSecurityExtensionForceEncryptThenMac [

Parameter Name	TcplpTlsUseSecurityExtensionForceEncryptThenMac		
Parent Container	<a href="#">TcplpTlsCiphersuiteDefinition</a>		
Description	Defines if the security extension according to IETF RFC 7366 shall be supported. This is useful for ciphersuites using CBC mode.		
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		

]

### [ECUC\_Tcplp\_00248] Definition of EcucEnumerationParamDef TcplpTlsVersion [

Parameter Name	TcplpTlsVersion		
Parent Container	<a href="#">TcplpTlsCiphersuiteDefinition</a>		
Description	Declares the TLS version that this ciphersuite shall be used for.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	TLS_VERSION_V12	–	
	TLS_VERSION_V13	–	
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

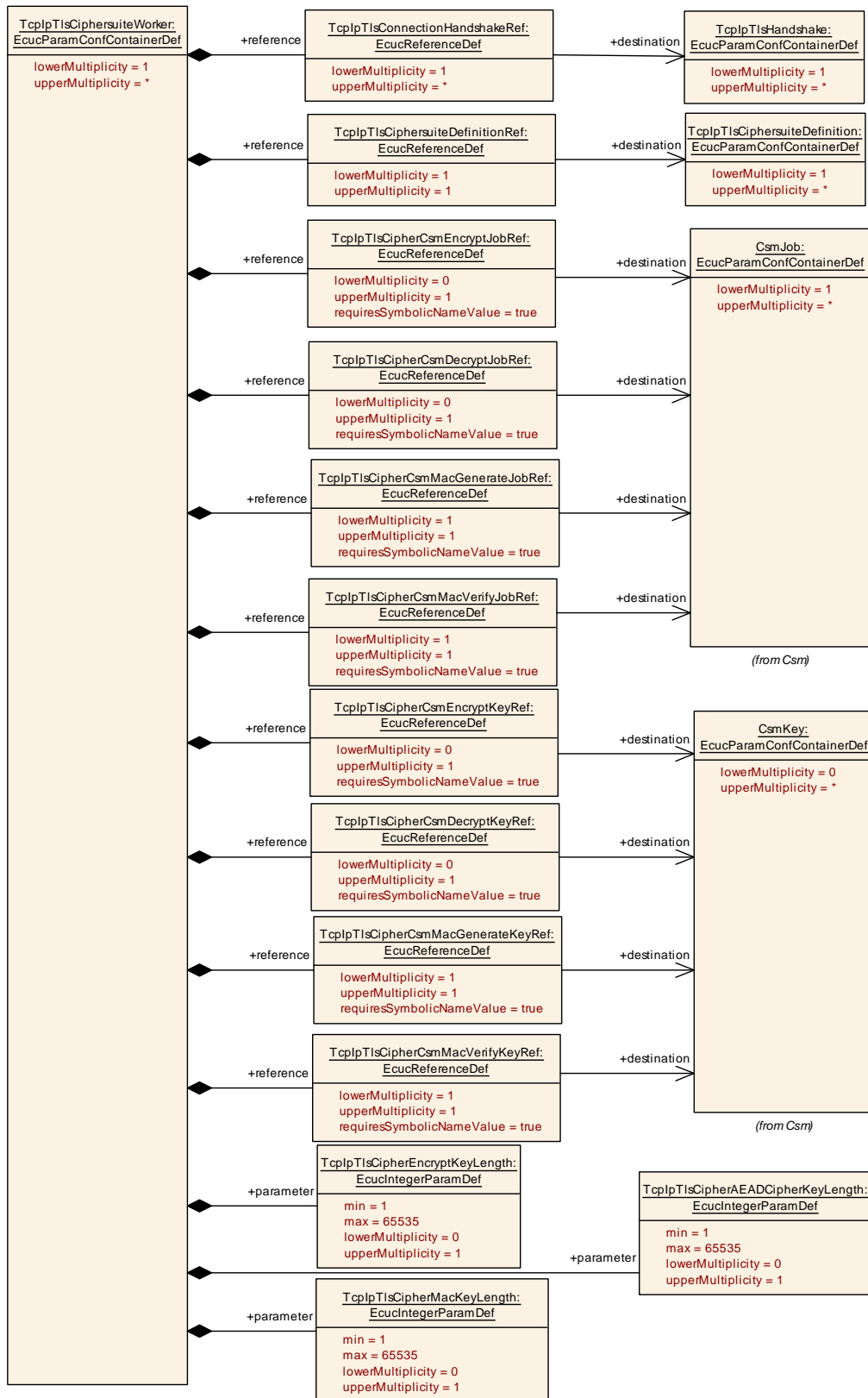


Figure 10.35: TcpIpTlsCiphersuiteWorker

## 10.2.54 TcplpTlsCiphersuiteWorker

### [ECUC\_Tcplp\_00238] Definition of EcucParamConfContainerDef TcplpTlsCiphersuiteWorker

Container Name	TcplpTlsCiphersuiteWorker
Parent Container	<a href="#">TcplpTlsCiphersuites</a>
Description	This container provides the jobs and keys necessary for TLS data transmission and reception.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpTlsCipherAEADCipherKeyLength</a>	0..1	[ECUC_Tcplp_00254]
<a href="#">TcplpTlsCipherEncryptKeyLength</a>	0..1	[ECUC_Tcplp_00253]
<a href="#">TcplpTlsCipherMacKeyLength</a>	0..1	[ECUC_Tcplp_00257]
<a href="#">TcplpTlsCipherCsmDecryptJobRef</a>	0..1	[ECUC_Tcplp_00255]
<a href="#">TcplpTlsCipherCsmDecryptKeyRef</a>	0..1	[ECUC_Tcplp_00256]
<a href="#">TcplpTlsCipherCsmEncryptJobRef</a>	0..1	[ECUC_Tcplp_00251]
<a href="#">TcplpTlsCipherCsmEncryptKeyRef</a>	0..1	[ECUC_Tcplp_00252]
<a href="#">TcplpTlsCipherCsmMacGenerateJobRef</a>	1	[ECUC_Tcplp_00258]
<a href="#">TcplpTlsCipherCsmMacGenerateKeyRef</a>	1	[ECUC_Tcplp_00259]
<a href="#">TcplpTlsCipherCsmMacVerifyJobRef</a>	1	[ECUC_Tcplp_00260]
<a href="#">TcplpTlsCipherCsmMacVerifyKeyRef</a>	1	[ECUC_Tcplp_00261]
<a href="#">TcplpTlsCiphersuiteDefinitionRef</a>	1	[ECUC_Tcplp_00250]
<a href="#">TcplpTlsConnectionHandshakeRef</a>	1..*	[ECUC_Tcplp_00249]

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

### [ECUC\_Tcplp\_00254] Definition of EcucIntegerParamDef TcplpTlsCipherAEAD-CipherKeyLength

Parameter Name	TcplpTlsCipherAEADCipherKeyLength		
Parent Container	<a href="#">TcplpTlsCiphersuiteWorker</a>		
Description	Defines the key length for en- / decryption with authentication data (AEAD).		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 65535		
Default value	–		
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  dependency: This value shall only be set if the cipher uses AEAD. If such a worker is selected, then Csm_AEADEncrypt() and Csm_AEADDecrypt() shall be used and AEAD shall be supported. Required to be set when TcpIpTlsCipherDefinition/TcpIpTlsAEADCipher is set to TRUE.		

### [ECUC\_Tcplp\_00253] Definition of EcucIntegerParamDef TcpIpTlsCipherEncryptKeyLength

Parameter Name	TcpIpTlsCipherEncryptKeyLength		
Parent Container	<a href="#">TcpIpTlsCiphersuiteWorker</a>		
Description	Defines the key length used for en- or decryption. The key length is valid for (symmetric) encryption and decryption.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 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: local		

### [ECUC\_Tcplp\_00257] Definition of EcucIntegerParamDef TcpIpTlsCipherMacKeyLength

Parameter Name	TcpIpTlsCipherMacKeyLength		
Parent Container	<a href="#">TcpIpTlsCiphersuiteWorker</a>		
Description	Specifies the length of the MAC key		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 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: local		

### [ECUC\_Tcplp\_00255] Definition of EcucReferenceDef TcplpTlsCipherCsmDecryptJobRef

Parameter Name	TcplpTlsCipherCsmDecryptJobRef		
Parent Container	<a href="#">TcplpTlsCiphersuiteWorker</a>		
Description	Reference to a CSM job to perform the data decryption operation		
Multiplicity	0..1		
Type	Symbolic name reference to CsmJob		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

### [ECUC\_Tcplp\_00256] Definition of EcucReferenceDef TcplpTlsCipherCsmDecryptKeyRef

Parameter Name	TcplpTlsCipherCsmDecryptKeyRef		
Parent Container	<a href="#">TcplpTlsCiphersuiteWorker</a>		
Description	Reference to a CSM key associated to the CSM job that performs the data decryption operation		
Multiplicity	0..1		
Type	Symbolic name reference to CsmKey		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

### [ECUC\_Tcplp\_00251] Definition of EcucReferenceDef TcplpTlsCipherCsmEncryptJobRef

Parameter Name	TcplpTlsCipherCsmEncryptJobRef		
Parent Container	<a href="#">TcplpTlsCiphersuiteWorker</a>		
Description	Reference to a CSM job to perform the data encryption operation		
Multiplicity	0..1		
Type	Symbolic name reference to CsmJob		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

### [ECUC\_Tcplp\_00252] Definition of EcucReferenceDef TcplpTlsCipherCsmEncryptKeyRef

Parameter Name	TcplpTlsCipherCsmEncryptKeyRef		
Parent Container	<a href="#">TcplpTlsCiphersuiteWorker</a>		
Description	Reference to a CSM key associated to the CSM job that performs the data encryption operation		
Multiplicity	0..1		
Type	Symbolic name reference to CsmKey		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

### [ECUC\_Tcplp\_00258] Definition of EcucReferenceDef TcplpTlsCipherCsmMacGenerateJobRef

Parameter Name	TcplpTlsCipherCsmMacGenerateJobRef		
Parent Container	<a href="#">TcplpTlsCiphersuiteWorker</a>		
Description	Reference to a CSM job to perform the MAC generate operation		
Multiplicity	1		
Type	Symbolic name reference to CsmJob		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

### [ECUC\_Tcplp\_00259] Definition of EcucReferenceDef TcplpTlsCipherCsmMacGenerateKeyRef

Parameter Name	TcplpTlsCipherCsmMacGenerateKeyRef		
Parent Container	<a href="#">TcplpTlsCiphersuiteWorker</a>		
Description	Reference to a CSM key associated to the CSM job that performs the MAC generate operation		
Multiplicity	1		
Type	Symbolic name reference to CsmKey		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	





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

└

### [ECUC\_Tcplp\_00260] Definition of EcucReferenceDef TcplpTlsCipherCsmMacVerifyJobRef

Parameter Name	TcplpTlsCipherCsmMacVerifyJobRef		
Parent Container	<a href="#">TcplpTlsCiphersuiteWorker</a>		
Description	Reference to a CSM job to perform the MAC verify operation		
Multiplicity	1		
Type	Symbolic name reference to CsmJob		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

└

### [ECUC\_Tcplp\_00261] Definition of EcucReferenceDef TcplpTlsCipherCsmMacVerifyKeyRef

Parameter Name	TcplpTlsCipherCsmMacVerifyKeyRef		
Parent Container	<a href="#">TcplpTlsCiphersuiteWorker</a>		
Description	Reference to a CSM key associated to the CSM job that performs the MAC verify operation		
Multiplicity	1		
Type	Symbolic name reference to CsmKey		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

└

### [ECUC\_Tcplp\_00250] Definition of EcucReferenceDef TcplpTlsCiphersuiteDefinitionRef

Parameter Name	TcplpTlsCiphersuiteDefinitionRef		
Parent Container	<a href="#">TcplpTlsCiphersuiteWorker</a>		
Description	Reference to a ciphersuite definition container		
Multiplicity	1		
Type	Reference to <a href="#">TcplpTlsCiphersuiteDefinition</a>		





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

┌

### [ECUC\_Tcplp\_00249] Definition of EcucReferenceDef TcplpTlsConnectionHandshakeRef ┌

<b>Parameter Name</b>	TcplpTlsConnectionHandshakeRef		
<b>Parent Container</b>	<a href="#">TcplpTlsCiphersuiteWorker</a>		
<b>Description</b>	References the container that contains the jobs and keys for handshake operation. Referencing multiple handshake containers allow to share them between workers and to choose the next unused during the handshake.		
<b>Multiplicity</b>	1..*		
<b>Type</b>	Reference to <a href="#">TcplpTlsHandshake</a>		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

└



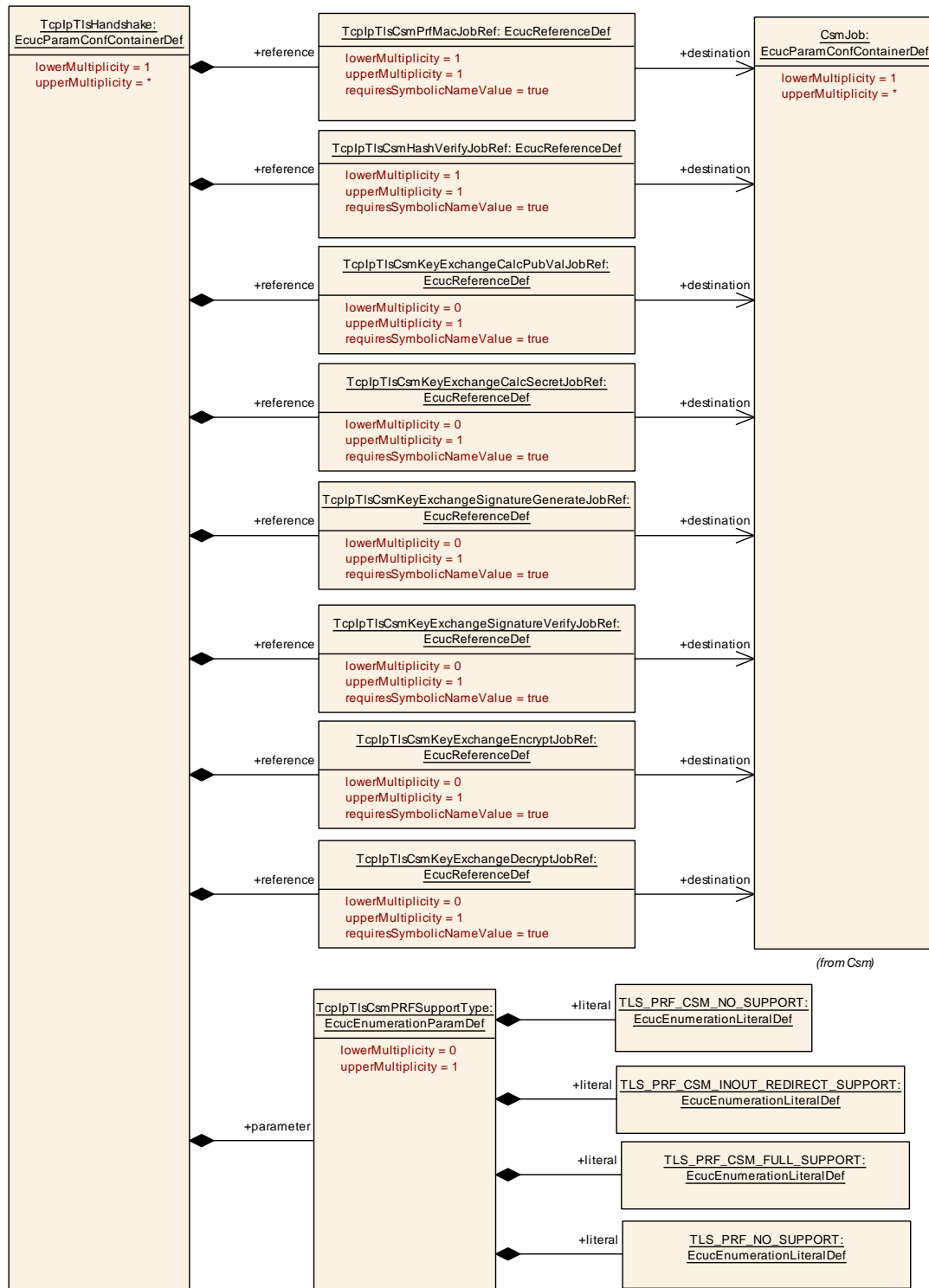


Figure 10.36: TcpIpTlsHandshake

## 10.2.55 TcplpTlsHandshake

### [ECUC\_Tcplp\_00239] Definition of EcucParamConfContainerDef TcplpTlsHandshake

Container Name	TcplpTlsHandshake
Parent Container	<a href="#">TcplpTlsCiphersuites</a>
Description	This container provides information that is needed to process a handshake. It contains the appropriate references to jobs and keys of the CSM to perform the key exchange cryptographic for the ciphersuite and involved certificates.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpTlsCsmPRFSupportType</a>	0..1	[ECUC_Tcplp_00264]
<a href="#">TcplpTlsCsmHashVerifyJobRef</a>	1	[ECUC_Tcplp_00265]
<a href="#">TcplpTlsCsmKeyExchangeCalcPubValJobRef</a>	0..1	[ECUC_Tcplp_00267]
<a href="#">TcplpTlsCsmKeyExchangeCalcSecretJobRef</a>	0..1	[ECUC_Tcplp_00269]
<a href="#">TcplpTlsCsmKeyExchangeDecryptJobRef</a>	0..1	[ECUC_Tcplp_00276]
<a href="#">TcplpTlsCsmKeyExchangeDecryptKeyRef</a>	0..1	[ECUC_Tcplp_00277]
<a href="#">TcplpTlsCsmKeyExchangeEncryptJobRef</a>	0..1	[ECUC_Tcplp_00274]
<a href="#">TcplpTlsCsmKeyExchangeEncryptKeyRef</a>	0..1	[ECUC_Tcplp_00275]
<a href="#">TcplpTlsCsmKeyExchangeKeyRef</a>	0..1	[ECUC_Tcplp_00268]
<a href="#">TcplpTlsCsmKeyExchangeSignatureGenerateJobRef</a>	0..1	[ECUC_Tcplp_00270]
<a href="#">TcplpTlsCsmKeyExchangeSignatureGenerateKeyRef</a>	0..1	[ECUC_Tcplp_00271]
<a href="#">TcplpTlsCsmKeyExchangeSignatureVerifyJobRef</a>	0..1	[ECUC_Tcplp_00272]
<a href="#">TcplpTlsCsmKeyExchangeSignatureVerifyKeyRef</a>	0..1	[ECUC_Tcplp_00273]
<a href="#">TcplpTlsCsmMasterSecretKeyRef</a>	0..1	[ECUC_Tcplp_00266]
<a href="#">TcplpTlsCsmPrfMacJobRef</a>	1	[ECUC_Tcplp_00262]
<a href="#">TcplpTlsCsmPrfMacKeyRef</a>	1	[ECUC_Tcplp_00263]

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

### [ECUC\_Tcplp\_00264] Definition of EcucEnumerationParamDef TcplpTlsCsmPRFSupportType

Parameter Name	TcplpTlsCsmPRFSupportType	
Parent Container	<a href="#">TcplpTlsHandshake</a>	
Description	Specifies how the CSM job supports the PRF operation.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	TLS_PRF_CSM_FULL_SUPPORT	–





	TLS_PRF_CSM_INOUT_REDIRECT_SUPPORT	—	
	TLS_PRF_CSM_NO_SUPPORT	—	
	TLS_PRF_NO_SUPPORT	—	
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		

## [ECUC\_Tcplp\_00265] Definition of EcucReferenceDef TcplpTlsCsmHashVerifyJobRef

Parameter Name	TcplpTlsCsmHashVerifyJobRef		
Parent Container	<a href="#">TcplpTlsHandshake</a>		
Description	Reference to a CSM job to perform the hash operation for the whole handshake.		
Multiplicity	1		
Type	Symbolic name reference to CsmJob		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

## [ECUC\_Tcplp\_00267] Definition of EcucReferenceDef TcplpTlsCsmKeyExchangeCalcPubValJobRef

Parameter Name	TcplpTlsCsmKeyExchangeCalcPubValJobRef		
Parent Container	<a href="#">TcplpTlsHandshake</a>		
Description	Reference to a CSM job to perform the DH Key Exchange algorithm operation		
Multiplicity	0..1		
Type	Symbolic name reference to CsmJob		
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		

]

### [ECUC\_Tcplp\_00269] Definition of EcucReferenceDef TcplpTlsCsmKeyExchangeCalcSecretJobRef [

Parameter Name	TcplpTlsCsmKeyExchangeCalcSecretJobRef		
Parent Container	<a href="#">TcplpTlsHandshake</a>		
Description	Reference to a CSM job to perform the Key Exchange algorithm operation		
Multiplicity	0..1		
Type	Symbolic name reference to CsmJob		
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 dependency: Only required if asynchronous job is used for key exchange calculation.		

]

### [ECUC\_Tcplp\_00276] Definition of EcucReferenceDef TcplpTlsCsmKeyExchangeDecryptJobRef [

Parameter Name	TcplpTlsCsmKeyExchangeDecryptJobRef		
Parent Container	<a href="#">TcplpTlsHandshake</a>		
Description	Reference to a CSM job to perform data decryption, e.g. with RSA key exchange operation.		
Multiplicity	0..1		
Type	Symbolic name reference to CsmJob		
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		

]

## [ECUC\_Tcplp\_00277] Definition of EcucReferenceDef TcplpTlsCsmKeyExchangeDecryptKeyRef

Parameter Name	TcplpTlsCsmKeyExchangeDecryptKeyRef		
Parent Container	<a href="#">TcplpTlsHandshake</a>		
Description	Reference to a CSM key to perform data decryption, e.g. with RSA, used for exchange operation.		
Multiplicity	0..1		
Type	Symbolic name reference to CsmKey		
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		

## [ECUC\_Tcplp\_00274] Definition of EcucReferenceDef TcplpTlsCsmKeyExchangeEncryptJobRef

Parameter Name	TcplpTlsCsmKeyExchangeEncryptJobRef		
Parent Container	<a href="#">TcplpTlsHandshake</a>		
Description	Reference to a CSM job to perform data encryption, e.g. with RSA key exchange operation.		
Multiplicity	0..1		
Type	Symbolic name reference to CsmJob		
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		

## [ECUC\_Tcplp\_00275] Definition of EcucReferenceDef TcplpTlsCsmKeyExchangeEncryptKeyRef [

Parameter Name	TcplpTlsCsmKeyExchangeEncryptKeyRef		
Parent Container	<a href="#">TcplpTlsHandshake</a>		
Description	Reference to a CSM key to perform data encryption, e.g. with RSA, used for exchange operation.		
Multiplicity	0..1		
Type	Symbolic name reference to CsmKey		
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		

]

## [ECUC\_Tcplp\_00268] Definition of EcucReferenceDef TcplpTlsCsmKeyExchangeKeyRef [

Parameter Name	TcplpTlsCsmKeyExchangeKeyRef		
Parent Container	<a href="#">TcplpTlsHandshake</a>		
Description	Reference to a CSM key used for Diffie Hellman (DH) key exchange operation.		
Multiplicity	0..1		
Type	Symbolic name reference to CsmKey		
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		

]

### [ECUC\_Tcplp\_00270] Definition of EcucReferenceDef TcplpTlsCsmKeyExchangeSignatureGenerateJobRef

Parameter Name	TcplpTlsCsmKeyExchangeSignatureGenerateJobRef		
Parent Container	<a href="#">TcplpTlsHandshake</a>		
Description	Reference to a CSM job to perform signature generation for DH operation		
Multiplicity	0..1		
Type	Symbolic name reference to CsmJob		
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		

### [ECUC\_Tcplp\_00271] Definition of EcucReferenceDef TcplpTlsCsmKeyExchangeSignatureGenerateKeyRef

Parameter Name	TcplpTlsCsmKeyExchangeSignatureGenerateKeyRef		
Parent Container	<a href="#">TcplpTlsHandshake</a>		
Description	Reference to a CSM key to perform signature generation for DH operation		
Multiplicity	0..1		
Type	Symbolic name reference to CsmKey		
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		

### [ECUC\_Tcplp\_00272] Definition of EcucReferenceDef TcplpTlsCsmKeyExchangeSignatureVerifyJobRef

Parameter Name	TcplpTlsCsmKeyExchangeSignatureVerifyJobRef		
Parent Container	<a href="#">TcplpTlsHandshake</a>		
Description	Reference to a CSM job to perform signature verification for DH operation		





<b>Multiplicity</b>	0..1		
<b>Type</b>	Symbolic name reference to CsmJob		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

### [ECUC\_Tcplp\_00273] Definition of EcucReferenceDef TcplpTlsCsmKeyExchangeSignatureVerifyKeyRef

<b>Parameter Name</b>	TcplpTlsCsmKeyExchangeSignatureVerifyKeyRef		
<b>Parent Container</b>	<a href="#">TcplpTlsHandshake</a>		
<b>Description</b>	Reference to a CSM key to perform signature verification for DH operation		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Symbolic name reference to CsmKey		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

### [ECUC\_Tcplp\_00266] Definition of EcucReferenceDef TcplpTlsCsmMasterSecretKeyRef

<b>Parameter Name</b>	TcplpTlsCsmMasterSecretKeyRef		
<b>Parent Container</b>	<a href="#">TcplpTlsHandshake</a>		
<b>Description</b>	This is the reference to the master key that is calculated during the session.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Symbolic name reference to CsmKey		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants







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

### [ECUC\_Tcplp\_00262] Definition of EcucReferenceDef TcplpTlsCsmPrfMacJob Ref

Parameter Name	TcplpTlsCsmPrfMacJobRef		
Parent Container	<a href="#">TcplpTlsHandshake</a>		
Description	Reference to a CSM job to perform the PRF hash operation		
Multiplicity	1		
Type	Symbolic name reference to CsmJob		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

### [ECUC\_Tcplp\_00263] Definition of EcucReferenceDef TcplpTlsCsmPrfMacKey Ref

Parameter Name	TcplpTlsCsmPrfMacKeyRef		
Parent Container	<a href="#">TcplpTlsHandshake</a>		
Description	Reference to a CSM key associated to the CSM job that performs the PRF hash operation		
Multiplicity	1		
Type	Symbolic name reference to CsmKey		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

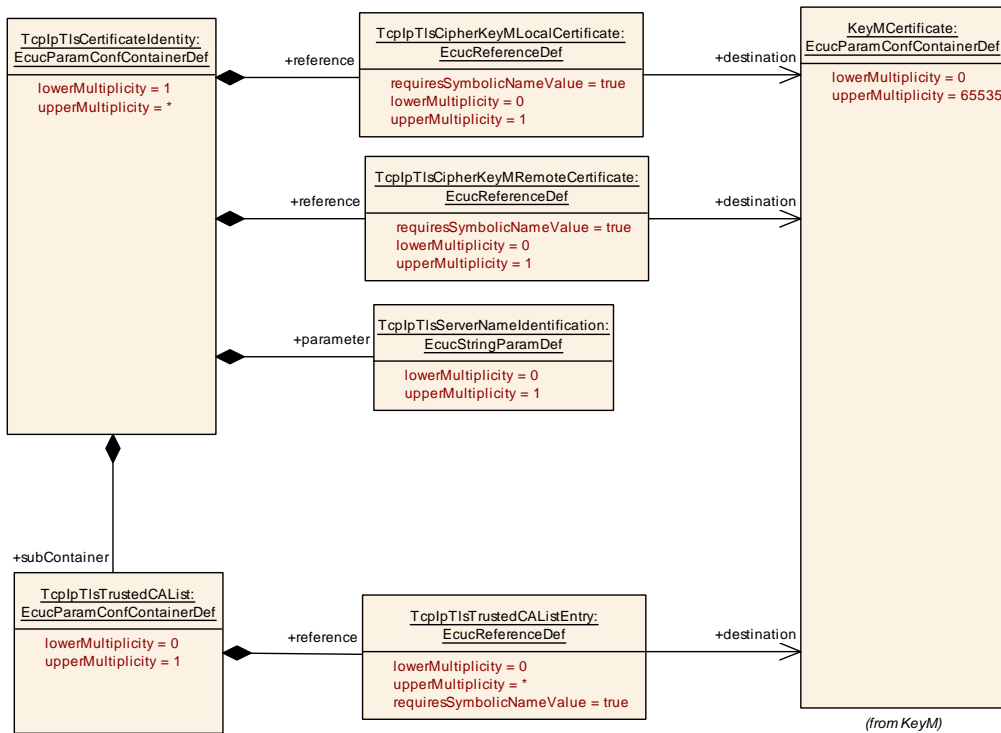


Figure 10.37: TcplpTlsCertificateIdentity

## 10.2.56 TcplpTlsCertificateIdentity

### [ECUC\_Tcplp\_00240] Definition of EcucParamConfContainerDef TcplpTlsCertificateIdentity

Container Name	TcplpTlsCertificateIdentity
Parent Container	<a href="#">TcplpTlsCiphersuites</a>
Description	This container provides information about the certificates used for ciphersuites.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpTlsServerNameIdentification</a>	0..1	[ECUC_Tcplp_00278]
<a href="#">TcplpTlsCipherKeyMLocalCertificate</a>	0..1	[ECUC_Tcplp_00286]
<a href="#">TcplpTlsCipherKeyMRemoteCertificate</a>	0..1	[ECUC_Tcplp_00287]

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpTlsTrustedCAList</a>	0..1	This container contains references to trusted CA certificates, whose names are sent in the client's ClientHello message, if *TcplpTlsUseExtensionTrustedCAKeys* is set to TRUE. <b>Tags:</b> atp.Status=draft

## [ECUC\_Tcplp\_00278] Definition of EcucStringParamDef TcplpTlsServerName Identification

Parameter Name	TcplpTlsServerNameIdentification		
Parent Container	<a href="#">TcplpTlsCertificateIdentity</a>		
Description	Defines a server identification name. If present, the name will be added as an extension with the "TLS client hello" handshake message. The TLS server will check for the name to identify the server certificate.		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		
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 dependency: Only needed if server name authentication is used.		

## [ECUC\_Tcplp\_00286] Definition of EcucReferenceDef TcplpTlsCipherKeyMLocal Certificate

Parameter Name	TcplpTlsCipherKeyMLocalCertificate		
Parent Container	<a href="#">TcplpTlsCertificateIdentity</a>		
Description	Reference to a KeyM certificate used to address the local certificate.		
Multiplicity	0..1		
Type	Symbolic name reference to KeyMCertificate		
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 dependency: Required if TcplpTlsConnectionType is TLS_SERVER. Also required if TcplpTlsConnectionType is TLS_CLIENT and the server requests a bidirectional authentication.		

## [ECUC\_Tcplp\_00287] Definition of EcucReferenceDef TcplpTlsCipherKeyMRemoteCertificate

Parameter Name	TcplpTlsCipherKeyMRemoteCertificate		
Parent Container	<a href="#">TcplpTlsCertificateIdentity</a>		
Description	Reference to KeyM certificate container to reference the remote certificate.		
Multiplicity	0..1		
Type	Symbolic name reference to KeyMCertificate		
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  dependency: This optional parameter is needed by the TLS_CLIENT and is used to verify the certificate provided by the TLS_SERVER. It is also required by the TLS_SERVER if bidirectional authentication will be requested. Otherwise, this parameter can be omitted.		

## [ECUC\_Tcplp\_00335] Definition of EcucParamConfContainerDef TcplpTlsTrustedCAList

*Status:* DRAFT

Container Name	TcplpTlsTrustedCAList		
Parent Container	<a href="#">TcplpTlsCertificateIdentity</a>		
Description	This container contains references to trusted CA certificates, whose names are sent in the client's ClientHello message, if *TcplpTlsUseExtensionTrustedCAKeys* is set to TRUE.  <b>Tags:</b> atp.Status=draft		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpTlsTrustedCAListEntry</a>	0..*	[ECUC_Tcplp_00336]

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

## [ECUC\_Tcplp\_00336] Definition of EcucReferenceDef TcplpTlsTrustedCAListEntry

Status: DRAFT

[

Parameter Name	TcplpTlsTrustedCAListEntry		
Parent Container	<a href="#">TcplpTlsTrustedCAList</a>		
Description	Reference to a KeyM certificate of a root CA. The list is sent by the TLS client in the Client Hello's trusted_ca_keys extension <b>Tags:</b> atp.Status=draft		
Multiplicity	0..*		
Type	Symbolic name reference to KeyMCertificate		
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		

]

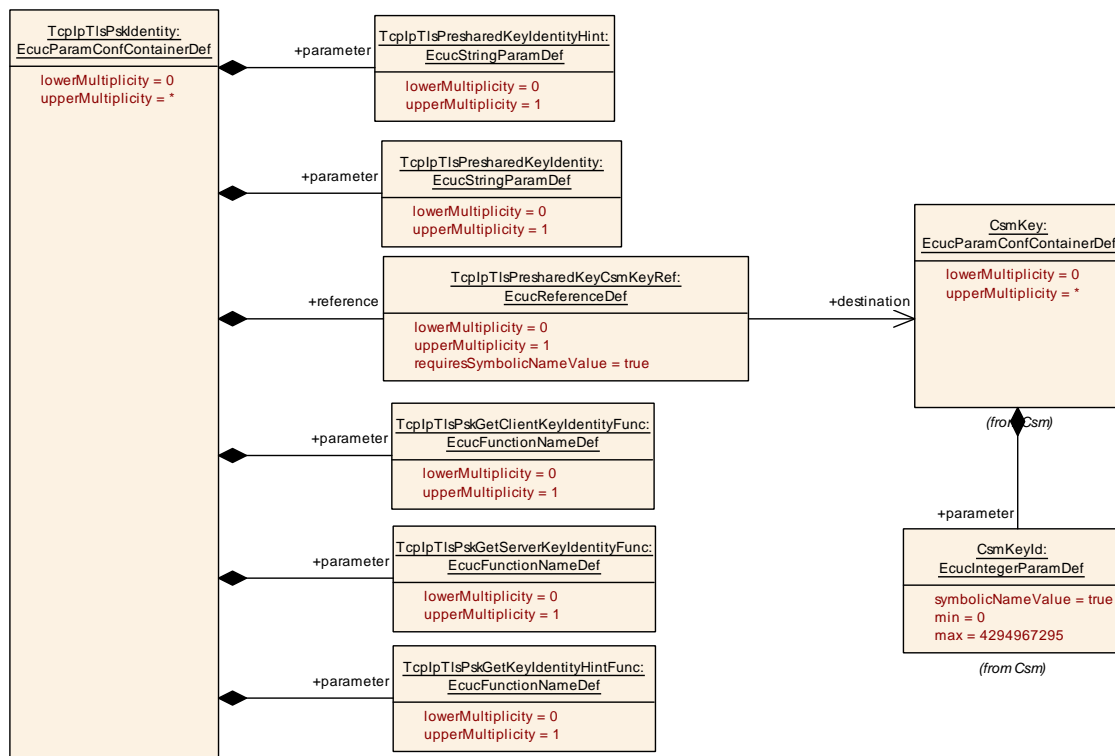


Figure 10.38: TcplpTlsPskIdentity

## 10.2.57 TcplpTlsPskIdentity

### [ECUC\_Tcplp\_00241] Definition of EcucParamConfContainerDef TcplpTlsPskIdentity [

<b>Container Name</b>	TcplpTlsPskIdentity
<b>Parent Container</b>	<a href="#">TcplpTlsCiphersuites</a>
<b>Description</b>	This container provides information about static definition of pre-shared keys. It is used during the handshake to negotiate pre-shared keys between a client and a server. Note: The callbacks for pre-shared keys are an alternative to the static definition. The callbacks allow to define the associated keys at runtime if pre-shared keys are used but no static definition is available. The container definition is used for static configuration.
<b>Configuration Parameters</b>	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpTlsPresharedKeyIdentity</a>	0..1	[ECUC_Tcplp_00284]
<a href="#">TcplpTlsPresharedKeyIdentityHint</a>	0..1	[ECUC_Tcplp_00279]
<a href="#">TcplpTlsPskGetClientKeyIdentityFunc</a>	0..1	[ECUC_Tcplp_00281]
<a href="#">TcplpTlsPskGetKeyIdentityHintFunc</a>	0..1	[ECUC_Tcplp_00283]
<a href="#">TcplpTlsPskGetServerKeyIdentityFunc</a>	0..1	[ECUC_Tcplp_00282]
<a href="#">TcplpTlsPresharedKeyCsmKeyRef</a>	0..1	[ECUC_Tcplp_00280]

<b>No Included Containers</b>
-------------------------------

]

### [ECUC\_Tcplp\_00284] Definition of EcucStringParamDef TcplpTlsPresharedKeyIdentity [

<b>Parameter Name</b>	TcplpTlsPresharedKeyIdentity		
<b>Parent Container</b>	<a href="#">TcplpTlsPskIdentity</a>		
<b>Description</b>	This item provides the key identification. The TLS client selects the pre-shared key based on the identification hint provided by the server and returns the key identification name back to the server.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucStringParamDef		
<b>Default value</b>	–		
<b>Regular Expression</b>	–		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	





<b>Scope / Dependency</b>	scope: local dependency: The callback function < Up_TlsClientGetPskIdentity> is used if the ciphersuite defines pre-shared key but this parameter is not present.
---------------------------	--

┌

### [ECUC\_Tcplp\_00279] Definition of EcucStringParamDef TcplpTlsPresharedKeyIdentityHint

<b>Parameter Name</b>	TcplpTlsPresharedKeyIdentityHint		
<b>Parent Container</b>	<a href="#">TcplpTlsPskIdentity</a>		
<b>Description</b>	Provides the identity hint for a pre-shared key. This information is transmitted by the TLS Server to provide its identification to the TLS client. The TLS client uses the same information to select the pre-shared key.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucStringParamDef		
<b>Default value</b>	–		
<b>Regular Expression</b>	–		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local dependency: The callback function <Up_TlsServerGetPskIdentityHint> is used if the ciphersuite defines pre-shared key but this parameter is not present.		

└

### [ECUC\_Tcplp\_00281] Definition of EcucFunctionNameDef TcplpTlsPskGetClientKeyIdentityFunc

<b>Parameter Name</b>	TcplpTlsPskGetClientKeyIdentityFunc		
<b>Parent Container</b>	<a href="#">TcplpTlsPskIdentity</a>		
<b>Description</b>	Defines the function name for the Up_TlsClientGetPskIdentity() callback.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default value</b>	–		
<b>Regular Expression</b>	–		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	





	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local  dependency: This definition is needed if a pre-shared key ciphersuite is used and TcpIpTlsPresharedKeyIdentity configuration parameter is not present. In this case, the callback function will be used to query the key identification.		

### [ECUC\_Tcplp\_00283] Definition of EcucFunctionNameDef TcplpTlsPskGetKeyIdentityHintFunc

<b>Parameter Name</b>	TcplpTlsPskGetKeyIdentityHintFunc		
<b>Parent Container</b>	<a href="#">TcplpTlsPskIdentity</a>		
<b>Description</b>	Defines the function name for the Up_TlsServerGetPskIdentityHint() callback.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default value</b>	–		
<b>Regular Expression</b>	–		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local  dependency: This definition is needed if a pre-shared key ciphersuite is used and TcpIpTlsPresharedKeyGetKeyIdentityHint configuration parameter is not present. In this case, the callback function will be used to query the key identity hint.		

### [ECUC\_Tcplp\_00282] Definition of EcucFunctionNameDef TcplpTlsPskGetServerKeyIdentityFunc

<b>Parameter Name</b>	TcplpTlsPskGetServerKeyIdentityFunc		
<b>Parent Container</b>	<a href="#">TcplpTlsPskIdentity</a>		
<b>Description</b>	Defines the function name for the Up_TlsServerGetPskIdentity () callback.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default value</b>	–		
<b>Regular Expression</b>	–		







<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local  dependency: This definition is needed if a pre-shared key ciphersuite is used and TcpIpTlsPresharedKeyIdentity configuration parameter is not present. In this case, the callback function will be used to query the key identification.		

]

### [ECUC\_Tcplp\_00280] Definition of EcucReferenceDef TcplpTlsPresharedKey CsmKeyRef [

<b>Parameter Name</b>	TcplpTlsPresharedKeyCsmKeyRef		
<b>Parent Container</b>	<a href="#">TcplpTlsPskIdentity</a>		
<b>Description</b>	Reference to a CSM key associated to the CSM job that performs the PRF hash operation		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Symbolic name reference to CsmKey		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local  dependency: Callback <Up_Tls[Server Client]GetPskIdentity> is used instead if this parameter is not present.		

]

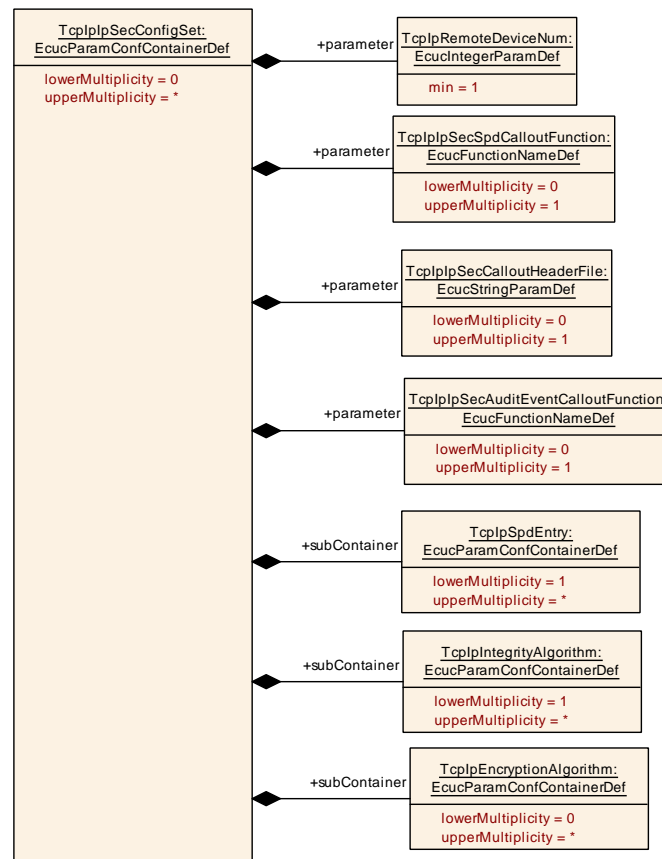


Figure 10.39: TcpIpSecConfig

## 10.2.58 TcpIpSecConfigSet

[ECUC\_Tcplp\_00288] Definition of EcucParamConfContainerDef TcpIpSec ConfigSet [

Container Name	TcpIpSecConfigSet
Parent Container	<a href="#">TcpIpConfig</a>
Description	Specifies the IPsec configuration.
Post-Build Variant Multiplicity	false
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcpIpSecAuditEventCalloutFunction</a>	0..1	[ECUC_Tcplp_00292]
<a href="#">TcpIpSecCalloutHeaderFile</a>	0..1	[ECUC_Tcplp_00291]
<a href="#">TcpIpSecSpdCalloutFunction</a>	0..1	[ECUC_Tcplp_00290]
<a href="#">TcpIpRemoteDeviceNum</a>	1	[ECUC_Tcplp_00289]

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpEncryptionAlgorithm</a>	0..*	Container for configuration of supported encryption algorithm transforms. This container is used to configure supported algorithms for ESP. The transform algorithm must be configured in the Crypto module.
<a href="#">TcplpIntegrityAlgorithm</a>	1..*	Container for configuration of supported integrity algorithm transforms. This container is used to configure supported algorithms for AH. The transform algorithm must be configured in the Crypto module.
<a href="#">TcplpSpdEntry</a>	1..*	Entry of the Security Policy Database (SPD).

### [ECUC\_Tcplp\_00292] Definition of EcucFunctionNameDef TcplpSecAuditEvent CalloutFunction

Parameter Name	TcplpSecAuditEventCalloutFunction		
Parent Container	<a href="#">TcplpSecConfigSet</a>		
Description	This parameter specifies the name of a callout function that will be called for each auditable event.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default value	–		
Regular Expression	–		
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		

### [ECUC\_Tcplp\_00291] Definition of EcucStringParamDef TcplpSecCallout HeaderFile

Parameter Name	TcplpSecCalloutHeaderFile		
Parent Container	<a href="#">TcplpSecConfigSet</a>		
Description	This parameter specifies the name of the header file containing the definition for the functions specified in TcplpSecSpdCalloutFunction and TcplpSecAuditEvent		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		





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		

### [ECUC\_Tcplp\_00290] Definition of EcucFunctionNameDef TcplpIpSecSpdCallout Function

Parameter Name	TcplpIpSecSpdCalloutFunction		
Parent Container	<a href="#">TcplpIpSecConfigSet</a>		
Description	This parameter specifies the name of a callout function that shall be called for each Rx/Tx message, after the IPsec has processed all corresponding SPD entries and has determined the policy. The callout function allows it to override the applied policy.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default value	–		
Regular Expression	–		
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		

### [ECUC\_Tcplp\_00289] Definition of EcucIntegerParamDef TcplpRemoteDevice Num

Parameter Name	TcplpRemoteDeviceNum		
Parent Container	<a href="#">TcplpIpSecConfigSet</a>		
Description	Amount of remote clients which will negotiate a Security Association (SA).		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 18446744073709551615		
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		

]

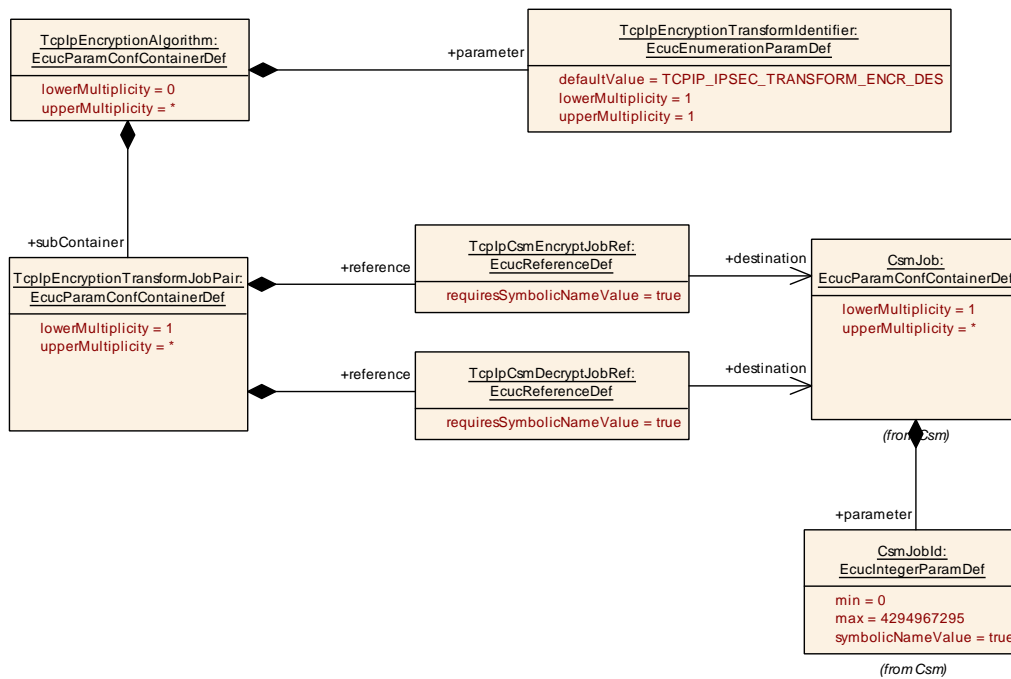


Figure 10.40: TcpIpEncryptionAlgorithm

## 10.2.59 TcpIpEncryptionAlgorithm

[ECUC\_TcpIp\_00317] Definition of EcucParamConfContainerDef TcpIpEncryptionAlgorithm [

Container Name	TcpIpEncryptionAlgorithm		
Parent Container	<a href="#">TcpIpSecConfigSet</a>		
Description	Container for configuration of supported encryption algorithm transforms. This container is used to configure supported algorithms for ESP. The transform algorithm must be configured in the Crypto module.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	





Configuration Parameters		
Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpEncryptionTransformIdentifier</a>	1	[ECUC_Tcplp_00311]
Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpEncryptionTransformJobPair</a>	1..*	Container for storing the CSM integrity transform job references for performing authentication. Valid for ESP and AH. At least one Integrity transform job pair needs to be configured for each Integrity Algorithm.

## [ECUC\_Tcplp\_00311] Definition of EcucEnumerationParamDef TcplpEncryptionTransformIdentifier

Parameter Name	TcplpEncryptionTransformIdentifier	
Parent Container	<a href="#">TcplpEncryptionAlgorithm</a>	
Description	Encryption algorithm transform identifier. Parameter values are defined as per IETF RFC 7296 3.3.2	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	TCPIP_IPSEC_TRANSFORM_ENCR_3DES	–
	TCPIP_IPSEC_TRANSFORM_ENCR_3IDEA	–
	TCPIP_IPSEC_TRANSFORM_ENCR_AES_CBC	–
	TCPIP_IPSEC_TRANSFORM_ENCR_AES_CCM_12	–
	TCPIP_IPSEC_TRANSFORM_ENCR_AES_CCM_16	–
	TCPIP_IPSEC_TRANSFORM_ENCR_AES_CCM_8	–
	TCPIP_IPSEC_TRANSFORM_ENCR_AES_CCM_8_IIV	–
	TCPIP_IPSEC_TRANSFORM_ENCR_AES_CTR	–
	TCPIP_IPSEC_TRANSFORM_ENCR_AES_GCM_12	–
	TCPIP_IPSEC_TRANSFORM_ENCR_AES_GCM_16	–
	TCPIP_IPSEC_TRANSFORM_ENCR_AES_GCM_16_IIV	–
	TCPIP_IPSEC_TRANSFORM_ENCR_AES_GCM_8	–
	TCPIP_IPSEC_TRANSFORM_ENCR_BLOWFISH	–





	TCPIP_IPSEC_TRANSFORM_ENCR_CAMELLIA_CBC	–	
	TCPIP_IPSEC_TRANSFORM_ENCR_CAMELLIA_CCM_12	–	
	TCPIP_IPSEC_TRANSFORM_ENCR_CAMELLIA_CCM_16	–	
	TCPIP_IPSEC_TRANSFORM_ENCR_CAMELLIA_CCM_8	–	
	TCPIP_IPSEC_TRANSFORM_ENCR_CAMELLIA_CTR	–	
	TCPIP_IPSEC_TRANSFORM_ENCR_CAST	–	
	TCPIP_IPSEC_TRANSFORM_ENCR_CHACHA20_POLY1305	–	
	TCPIP_IPSEC_TRANSFORM_ENCR_CHACHA20_POLY1305_II	–	
	TCPIP_IPSEC_TRANSFORM_ENCR_DES	–	
	TCPIP_IPSEC_TRANSFORM_ENCR_DES_IV32	–	
	TCPIP_IPSEC_TRANSFORM_ENCR_DES_IV64	–	
	TCPIP_IPSEC_TRANSFORM_ENCR_IDEA	–	
	TCPIP_IPSEC_TRANSFORM_ENCR_NULL	–	
	TCPIP_IPSEC_TRANSFORM_ENCR_RC5	–	
Default value	TCPIP_IPSEC_TRANSFORM_ENCR_DES		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

## 10.2.60 TcplpEncryptionTransformJobPair

[ECUC\_Tcplp\_00312] Definition of EcucParamConfContainerDef TcplpEncryptionTransformJobPair [

Container Name	TcplpEncryptionTransformJobPair
Parent Container	<a href="#">TcplpEncryptionAlgorithm</a>
Description	Container for storing the CSM integrity transform job references for performing authentication. Valid for ESP and AH. At least one Integrity transform job pair needs to be configured for each Integrity Algorithm.





<b>Post-Build Variant Multiplicity</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Configuration Parameters</b>			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpCsmDecryptJobRef</a>	1	[ECUC_Tcplp_00314]
<a href="#">TcplpCsmEncryptJobRef</a>	1	[ECUC_Tcplp_00313]

<b>No Included Containers</b>
-------------------------------

]

### [ECUC\_Tcplp\_00314] Definition of EcucReferenceDef TcplpCsmDecryptJobRef

[

<b>Parameter Name</b>	TcplpCsmDecryptJobRef		
<b>Parent Container</b>	<a href="#">TcplpEncryptionTransformJobPair</a>		
<b>Description</b>	The referenced Csm job is used for the execution of the CsmMacVerify primitive needed for this transform. Must be a valid decryption job of the parent type.		
<b>Multiplicity</b>	1		
<b>Type</b>	Symbolic name reference to CsmJob		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

]

### [ECUC\_Tcplp\_00313] Definition of EcucReferenceDef TcplpCsmEncryptJobRef

[

<b>Parameter Name</b>	TcplpCsmEncryptJobRef		
<b>Parent Container</b>	<a href="#">TcplpEncryptionTransformJobPair</a>		
<b>Description</b>	The referenced Csm job is used for the execution of the CsmMacGenerate primitive needed for this transform. Must be a valid encryption job of the parent type.		
<b>Multiplicity</b>	1		
<b>Type</b>	Symbolic name reference to CsmJob		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	







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

┌

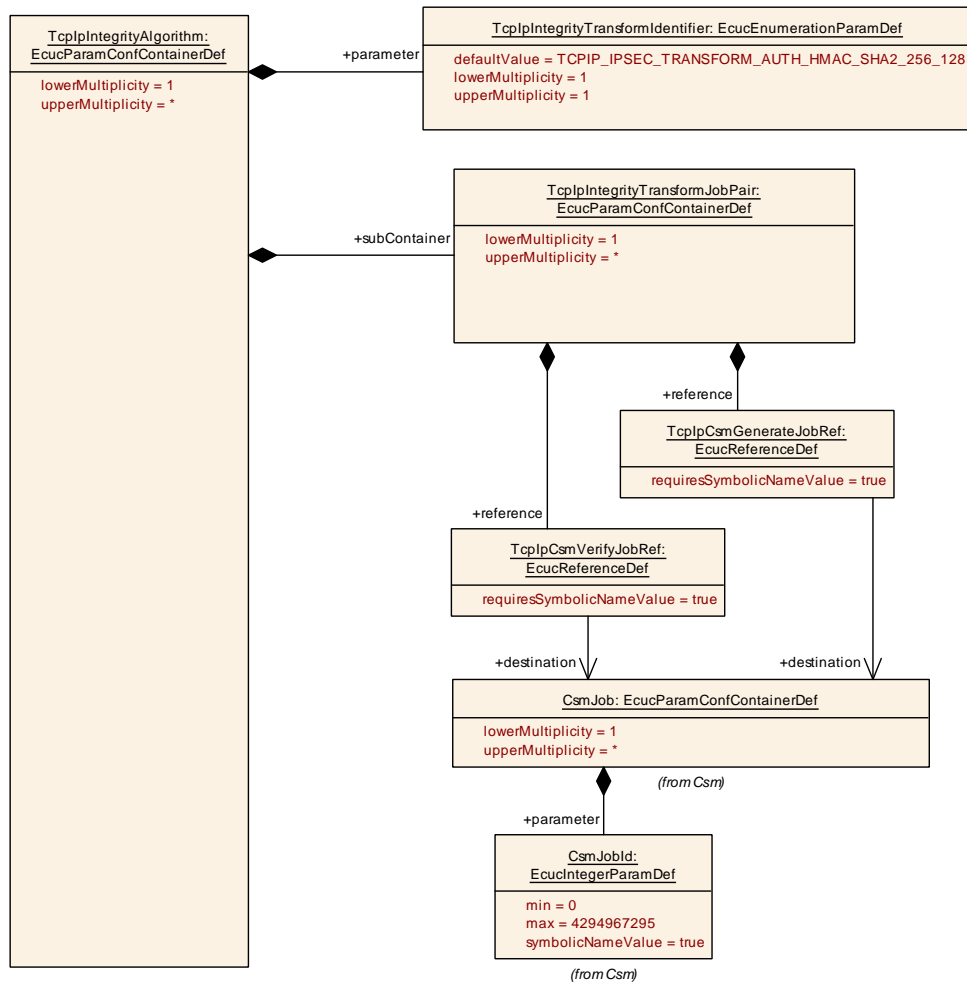


Figure 10.41: TcpIpIntegrityAlgorithm

## 10.2.61 TcpIpIntegrityAlgorithm

[ECUC\_TcpIp\_00294] Definition of EcucParamConfContainerDef TcpIpIntegrity Algorithm

<b>Container Name</b>	TcplpIntegrityAlgorithm		
<b>Parent Container</b>	<a href="#">TcplpSecConfigSet</a>		
<b>Description</b>	Container for configuration of supported integrity algorithm transforms. This container is used to configure supported algorithms for AH. The transform algorithm must be configured in the Crypto module.		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Configuration Parameters</b>			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">TcplpIntegrityTransformIdentifier</a>	1	[ <a href="#">ECUC_Tcplp_00307</a> ]

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">TcplpIntegrityTransformJobPair</a>	1..*	Container for storing the CSM integrity transform job references for performing authentication. Valid for ESP and AH. At least one Integrity transform job pair needs to be configured for each Integrity Algorithm.

]

## [[ECUC\\_Tcplp\\_00307](#)] Definition of EcucEnumerationParamDef TcplpIntegrityTransformIdentifier [

<b>Parameter Name</b>	TcplpIntegrityTransformIdentifier	
<b>Parent Container</b>	<a href="#">TcplpIntegrityAlgorithm</a>	
<b>Description</b>	Integrity algorithm transform identifier. Parameter values are defined as per IETF RFC 7296 3.3.2	
<b>Multiplicity</b>	1	
<b>Type</b>	EcucEnumerationParamDef	
<b>Range</b>	TCPIP_IPSEC_TRANSFORM_AUTH_AES_128_GMAC	–
	TCPIP_IPSEC_TRANSFORM_AUTH_AES_192_GMAC	–
	TCPIP_IPSEC_TRANSFORM_AUTH_AES_256_GMAC	–
	TCPIP_IPSEC_TRANSFORM_AUTH_AES_CMAC_96	–
	TCPIP_IPSEC_TRANSFORM_AUTH_AES_XCBC_96	–
	TCPIP_IPSEC_TRANSFORM_AUTH_DES_MAC	–
	TCPIP_IPSEC_TRANSFORM_AUTH_HMAC_MD5_128	–
	TCPIP_IPSEC_TRANSFORM_AUTH_HMAC_MD5_96	–





	TCPIP_IPSEC_TRANSFORM_AUTH_HMAC_SHA1_160	–	
	TCPIP_IPSEC_TRANSFORM_AUTH_HMAC_SHA1_96	–	
	TCPIP_IPSEC_TRANSFORM_AUTH_HMAC_SHA2_256_128	–	
	TCPIP_IPSEC_TRANSFORM_AUTH_HMAC_SHA2_384_192	–	
	TCPIP_IPSEC_TRANSFORM_AUTH_HMAC_SHA2_512_256	–	
	TCPIP_IPSEC_TRANSFORM_AUTH_KPDK_MD5	–	
Default value	TCPIP_IPSEC_TRANSFORM_AUTH_HMAC_SHA2_256_128		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

## 10.2.62 TcplpIntegrityTransformJobPair

### [ECUC\_Tcplp\_00308] Definition of EcucParamConfContainerDef TcplpIntegrity TransformJobPair [

Container Name	TcplpIntegrityTransformJobPair		
Parent Container	TcplpIntegrityAlgorithm		
Description	Container for storing the CSM integrity transform job references for performing authentication. Valid for ESP and AH. At least one Integrity transform job pair needs to be configured for each Integrity Algorithm.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
TcplpCsmGenerateJobRef	1	[ECUC_Tcplp_00309]
TcplpCsmVerifyJobRef	1	[ECUC_Tcplp_00310]

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

]

## [ECUC\_Tcplp\_00309] Definition of EcucReferenceDef TcplpCsmGenerateJobRef

[

Parameter Name	TcplpCsmGenerateJobRef		
Parent Container	<a href="#">TcplpIntegrityTransformJobPair</a>		
Description	The referenced Csm job is used for the execution of the CsmMacGenerate primitive needed for this transform. Must be a valid MAC generate job of the parent type.		
Multiplicity	1		
Type	Symbolic name reference to CsmJob		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

## [ECUC\_Tcplp\_00310] Definition of EcucReferenceDef TcplpCsmVerifyJobRef

[

Parameter Name	TcplpCsmVerifyJobRef		
Parent Container	<a href="#">TcplpIntegrityTransformJobPair</a>		
Description	The referenced Csm job is used for the execution of the CsmMacVerify primitive needed for this transform. Must be a valid MAC verify job of the parent type.		
Multiplicity	1		
Type	Symbolic name reference to CsmJob		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

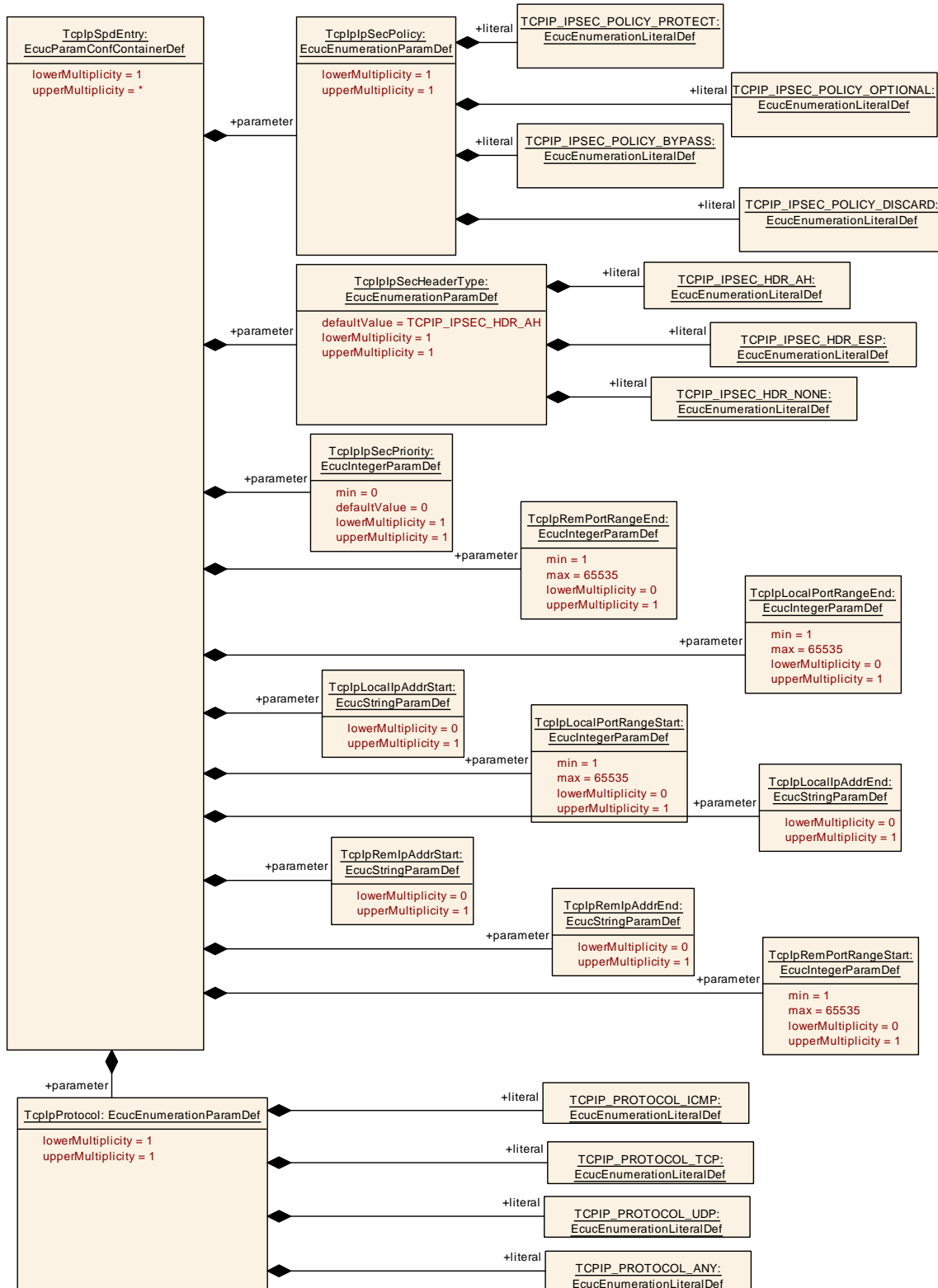


Figure 10.42: TcpIpSpdEntry

### 10.2.63 TcplpSpdEntry

#### [ECUC\_Tcplp\_00293] Definition of EcucParamConfContainerDef TcplpSpdEntry

Container Name	TcplpSpdEntry		
Parent Container	<a href="#">TcplpSecConfigSet</a>		
Description	Entry of the Security Policy Database (SPD).		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	—	
	Post-build time	—	
Configuration Parameters			
Included Parameters			
Parameter Name	Multiplicity	ECUC ID	
<a href="#">TcplpSecHeaderType</a>	1	<a href="#">[ECUC_Tcplp_00297]</a>	
<a href="#">TcplpSecPolicy</a>	1	<a href="#">[ECUC_Tcplp_00295]</a>	
<a href="#">TcplpSecPriority</a>	1	<a href="#">[ECUC_Tcplp_00296]</a>	
<a href="#">TcplpLocalIpAddrEnd</a>	0..1	<a href="#">[ECUC_Tcplp_00301]</a>	
<a href="#">TcplpLocalIpAddrStart</a>	0..1	<a href="#">[ECUC_Tcplp_00300]</a>	
<a href="#">TcplpLocalPortRangeEnd</a>	0..1	<a href="#">[ECUC_Tcplp_00299]</a>	
<a href="#">TcplpLocalPortRangeStart</a>	0..1	<a href="#">[ECUC_Tcplp_00298]</a>	
<a href="#">TcplpProtocol</a>	1	<a href="#">[ECUC_Tcplp_00306]</a>	
<a href="#">TcplpRemIpAddrEnd</a>	0..1	<a href="#">[ECUC_Tcplp_00303]</a>	
<a href="#">TcplpRemIpAddrStart</a>	0..1	<a href="#">[ECUC_Tcplp_00302]</a>	
<a href="#">TcplpRemPortRangeEnd</a>	0..1	<a href="#">[ECUC_Tcplp_00305]</a>	
<a href="#">TcplpRemPortRangeStart</a>	0..1	<a href="#">[ECUC_Tcplp_00304]</a>	
No Included Containers			

#### [ECUC\_Tcplp\_00297] Definition of EcucEnumerationParamDef TcplpSecHeaderType

<b>Parameter Name</b>	TcplpSecHeaderType		
<b>Parent Container</b>	<a href="#">TcplpSpdEntry</a>		
<b>Description</b>	Header type specifying the IPsec security mechanism.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	TCPIP_IPSEC_HDR_AH	–	
	TCPIP_IPSEC_HDR_ESP	–	
	TCPIP_IPSEC_HDR_NONE	–	
<b>Default value</b>	<a href="#">TCPIP_IPSEC_HDR_AH</a>		





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

]

## [ECUC\_Tcplp\_00295] Definition of EcucEnumerationParamDef TcplpIpSecPolicy

[

Parameter Name	TcplpIpSecPolicy		
Parent Container	<a href="#">TcplpSpdEntry</a>		
Description	Policy for usage of IPsec.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	TCPIP_IPSEC_POLICY_BYPASS	–	
	TCPIP_IPSEC_POLICY_DISCARD	–	
	TCPIP_IPSEC_POLICY_OPTIONAL	–	
	TCPIP_IPSEC_POLICY_PROTECT	–	
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

## [ECUC\_Tcplp\_00296] Definition of EcucIntegerParamDef TcplpIpSecPriority

[

Parameter Name	TcplpIpSecPriority	
Parent Container	<a href="#">TcplpSpdEntry</a>	
Description	Priority of the SPD entry. The processing of entries is based on priority, starting with the highest priority "0". The first matching SPD entry defines the policy.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 18446744073709551615	
Default value	0	
Post-Build Variant Multiplicity	false	
Post-Build Variant Value	false	
Scope / Dependency	scope: local	

]

## [ECUC\_Tcplp\_00301] Definition of EcucStringParamDef TcplpLocallpAddrEnd [

Parameter Name	TcplpLocallpAddrEnd		
Parent Container	<a href="#">TcplpSpdEntry</a>		
Description	End value of the remote IP address range.		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		
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		

]

## [ECUC\_Tcplp\_00300] Definition of EcucStringParamDef TcplpLocallpAddrStart [

Parameter Name	TcplpLocallpAddrStart		
Parent Container	<a href="#">TcplpSpdEntry</a>		
Description	Start value of the local IP address range.		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		
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		

]



## [ECUC\_Tcplp\_00299] Definition of EcucIntegerParamDef TcplpLocalPortRangeEnd

Parameter Name	TcplpLocalPortRangeEnd		
Parent Container	<a href="#">TcplpSpdEntry</a>		
Description	End value of the local port range.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 65535		
Default value	–		
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		

## [ECUC\_Tcplp\_00298] Definition of EcucIntegerParamDef TcplpLocalPortRangeStart

Parameter Name	TcplpLocalPortRangeStart		
Parent Container	<a href="#">TcplpSpdEntry</a>		
Description	Start value of the local port range.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 65535		
Default value	–		
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		

## [ECUC\_Tcplp\_00306] Definition of EcucEnumerationParamDef TcplpProtocol [

Parameter Name	TcplpProtocol		
Parent Container	<a href="#">TcplpSpdEntry</a>		
Description	Relevant IP protocol. Note: As specified in IETF Rfc 4301 section 6, ICMP error messages will always be BYPASSED. The policy for TCPIP_PROTOCOL_ICMP only applies to ICMP non-error messages. (Echo reply/response).		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	TCPIP_PROTOCOL_ANY	–	
	TCPIP_PROTOCOL_ICMP	–	
	TCPIP_PROTOCOL_TCP	–	
	TCPIP_PROTOCOL_UDP	–	
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

## [ECUC\_Tcplp\_00303] Definition of EcucStringParamDef TcplpRemIpAddrEnd [

Parameter Name	TcplpRemIpAddrEnd		
Parent Container	<a href="#">TcplpSpdEntry</a>		
Description	End value of the remote IP address range.		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		
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		

## [ECUC\_Tcplp\_00302] Definition of EcucStringParamDef TcplpRemIpAddrStart [

Parameter Name	TcplpRemIpAddrStart		
Parent Container	<a href="#">TcplpSpdEntry</a>		
Description	Start value of the remote IP address range.		





Multiplicity	0..1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		
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		

### [ECUC\_Tcplp\_00305] Definition of EcucIntegerParamDef TcplpRemPortRange End

Parameter Name	TcplpRemPortRangeEnd		
Parent Container	<a href="#">TcplpSpdEntry</a>		
Description	End value of the remote port range.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 65535		
Default value	–		
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		

### [ECUC\_Tcplp\_00304] Definition of EcucIntegerParamDef TcplpRemPortRange Start

Parameter Name	TcplpRemPortRangeStart		
Parent Container	<a href="#">TcplpSpdEntry</a>		
Description	Start value of the remote port range.		
Multiplicity	0..1		





Type	EcucIntegerParamDef		
Range	1 .. 65535		
Default value	–		
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		

]

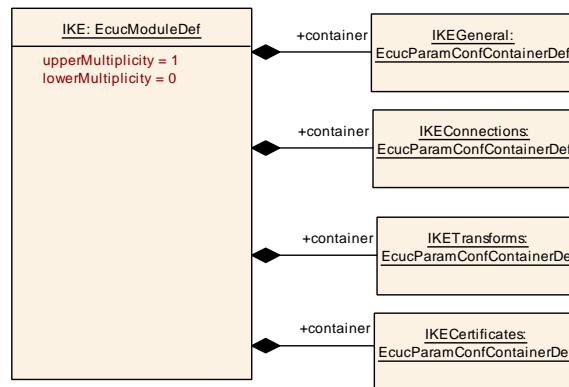


Figure 10.43: IKE

## 10.2.64 IKE

### [ECUC\_IKE\_00001] Definition of EcucModuleDef IKE [

Module Name	IKE
Description	Description for the Internet Key Exchange.
Post-Build Variant Support	false
Supported Config Variants	VARIANT-PRE-COMPILE

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">IKECertificates</a>	1	Container for configuration of IKE certificates.
<a href="#">IKEConnections</a>	1	Container for configuration of IKE connections.
<a href="#">IKEGeneral</a>	1	General module settings.
<a href="#">IKETransforms</a>	1	Container for configuration of IKE transforms.

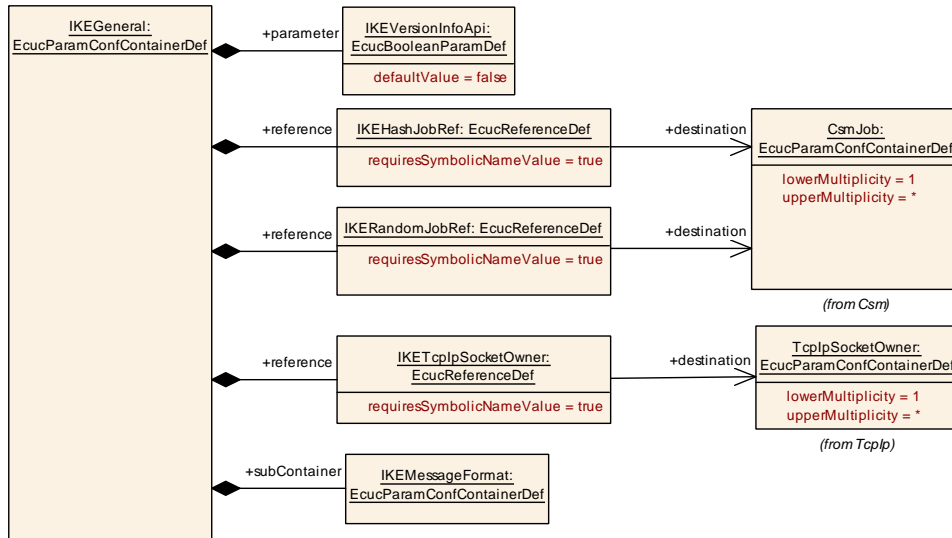


Figure 10.44: IKEGeneral

## 10.2.65 IKEGeneral

### [ECUC\_IKE\_00002] Definition of EcucParamConfContainerDef IKEGeneral

Container Name	IKEGeneral
Parent Container	IKE
Description	General module settings.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
IKEVersionInfoApi	1	[ECUC_IKE_00008]
IKEHashJobRef	1	[ECUC_IKE_00009]
IKERandomJobRef	1	[ECUC_IKE_00010]
IKETcpIpSocketOwner	1	[ECUC_IKE_00011]

Included Containers		
Container Name	Multiplicity	Scope / Dependency
IKEMessageFormat	1	In order to deserialize the byte stream of IKE messages to data structures memory is statically allocated. Use the parameters in this container to minimize the used memory. But, configuring too low maximum values might result in unsuccessful deserializations of received IKE messages.

### [ECUC\_IKE\_00008] Definition of EcucBooleanParamDef IKEVersionInfoApi [

Parameter Name	IKEVersionInfoApi		
Parent Container	<a href="#">IKEGeneral</a>		
Description	Pre-processor switch to enable and disable availability of the API IKE_GetVersionInfo(). <ul style="list-style-type: none"> <li>• True: API IKE_GetVersionInfo() is available.</li> <li>• False: API IKE_GetVersionInfo() is not available.</li> </ul>		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
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		

]

### [ECUC\_IKE\_00009] Definition of EcucReferenceDef IKEHashJobRef [

Parameter Name	IKEHashJobRef		
Parent Container	<a href="#">IKEGeneral</a>		
Description	The referenced crypto job is used to calculate the SHA-1 hash of the Subject Public Key Info element needed for the encoding of the certification authorities.		
Multiplicity	1		
Type	Symbolic name reference to CsmJob		
Scope / Dependency	scope: local		

]

### [ECUC\_IKE\_00010] Definition of EcucReferenceDef IKERandomJobRef [

Parameter Name	IKERandomJobRef		
Parent Container	<a href="#">IKEGeneral</a>		
Description	The referenced crypto job is used for random number generation.		
Multiplicity	1		
Type	Symbolic name reference to CsmJob		
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		

]

### [ECUC\_IKE\_00011] Definition of EcucReferenceDef IKETcpIpSocketOwner [

Parameter Name	IKETcpIpSocketOwner		
Parent Container	<a href="#">IKEGeneral</a>		
Description	The ID of the socket user.		
Multiplicity	1		
Type	Symbolic name reference to <a href="#">TcpIpSocketOwner</a>		
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		

]

## 10.2.66 IKEMessageFormat

### [ECUC\_IKE\_00012] Definition of EcucParamConfContainerDef IKEMessageFormat [

Container Name	IKEMessageFormat		
Parent Container	<a href="#">IKEGeneral</a>		
Description	In order to deserialize the byte stream of IKE messages to data structures memory is statically allocated. Use the parameters in this container to minimize the used memory. But, configuring too low maximum values might result in unsuccessful deserializations of received IKE messages.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">IKEMaxAttributesPerTransform</a>	1	<a href="#">[ECUC_IKE_00014]</a>
<a href="#">IKEMaxCertPayloadsPerMessage</a>	1	<a href="#">[ECUC_IKE_00020]</a>
<a href="#">IKEMaxCertreqPayloadsPerMessage</a>	1	<a href="#">[ECUC_IKE_00021]</a>
<a href="#">IKEMaxDeletePayloadsPerMessage</a>	1	<a href="#">[ECUC_IKE_00022]</a>
<a href="#">IKEMaxInitMessageSize</a>	1	<a href="#">[ECUC_IKE_00024]</a>
<a href="#">IKEMaxNonceSize</a>	1	<a href="#">[ECUC_IKE_00023]</a>
<a href="#">IKEMaxNotifyPayloadsPerMessage</a>	1	<a href="#">[ECUC_IKE_00019]</a>
<a href="#">IKEMaxPayloadsPerMessage</a>	1	<a href="#">[ECUC_IKE_00018]</a>
<a href="#">IKEMaxProposalsPerSaPayload</a>	1	<a href="#">[ECUC_IKE_00015]</a>
<a href="#">IKEMaxSpisPerDeletePayload</a>	1	<a href="#">[ECUC_IKE_00017]</a>
<a href="#">IKEMaxTrafficSelectorsPerTsPayload</a>	1	<a href="#">[ECUC_IKE_00016]</a>
<a href="#">IKEMaxTransformsPerProp</a>	1	<a href="#">[ECUC_IKE_00013]</a>

No Included Containers

## [ECUC\_IKE\_00014] Definition of EcucIntegerParamDef IKEMaxAttributesPer Transform

Parameter Name	IKEMaxAttributesPerTransform		
Parent Container	<a href="#">IKEMessageFormat</a>		
Description	The maximum number of attributes a transform may contain.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 2		
Default value	2		
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		



## [ECUC\_IKE\_00020] Definition of EcucIntegerParamDef IKEMaxCertPayloadsPerMessage

Parameter Name	IKEMaxCertPayloadsPerMessage		
Parent Container	<a href="#">IKEMessageFormat</a>		
Description	The maximum number of Certificate payloads an IKE message may contain.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 4		
Default value	4		
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		

## [ECUC\_IKE\_00021] Definition of EcucIntegerParamDef IKEMaxCertreqPayloadsPerMessage

Parameter Name	IKEMaxCertreqPayloadsPerMessage		
Parent Container	<a href="#">IKEMessageFormat</a>		
Description	The maximum number of Certificate Request payloads an IKE message may contain.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 4		
Default value	4		
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		

## [ECUC\_IKE\_00022] Definition of EcucIntegerParamDef IKEMaxDeletePayloadsPerMessage

Parameter Name	IKEMaxDeletePayloadsPerMessage		
Parent Container	<a href="#">IKEMessageFormat</a>		
Description	The maximum number of Delete payloads an IKE message may contain.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 2		
Default value	2		
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		

## [ECUC\_IKE\_00024] Definition of EcucIntegerParamDef IKEMaxInitMessageSize

Parameter Name	IKEMaxInitMessageSize		
Parent Container	<a href="#">IKEMessageFormat</a>		
Description	The maximum size of incoming IKE_INIT messages.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	256 .. 3000		
Default value	512		
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		

## [ECUC\_IKE\_00023] Definition of EcucIntegerParamDef IKEMaxNonceSize [

Parameter Name	IKEMaxNonceSize		
Parent Container	<a href="#">IKEMessageFormat</a>		
Description	The maximum size of incoming nonces. Must be at least 32 bytes and at least half the key size of the largest configured pseudorandom function (PRF).		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	32 .. 512		
Default value	64		
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		

]

## [ECUC\_IKE\_00019] Definition of EcucIntegerParamDef IKEMaxNotifyPayloadsPerMessage [

Parameter Name	IKEMaxNotifyPayloadsPerMessage		
Parent Container	<a href="#">IKEMessageFormat</a>		
Description	The maximum number of Notify payloads an IKE message may contain.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 10		
Default value	10		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

]

## [ECUC\_IKE\_00018] Definition of EcucIntegerParamDef IKEMaxPayloadsPerMessage

Parameter Name	IKEMaxPayloadsPerMessage		
Parent Container	<a href="#">IKEMessageFormat</a>		
Description	The maximum number of payloads an IKE message may contain.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 20		
Default value	20		
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		

## [ECUC\_IKE\_00015] Definition of EcucIntegerParamDef IKEMaxProposalsPerSaPayload

Parameter Name	IKEMaxProposalsPerSaPayload		
Parent Container	<a href="#">IKEMessageFormat</a>		
Description	The maximum number of proposals a SA payload may contain.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 5		
Default value	5		
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		

## [ECUC\_IKE\_00017] Definition of EcucIntegerParamDef IKEMaxSpisPerDelete Payload [

Parameter Name	IKEMaxSpisPerDeletePayload		
Parent Container	<a href="#">IKEMessageFormat</a>		
Description	The maximum number of SPIs a Delete payload may contain.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 2		
Default value	2		
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		

]

## [ECUC\_IKE\_00016] Definition of EcucIntegerParamDef IKEMaxTrafficSelectors PerTsPayload [

Parameter Name	IKEMaxTrafficSelectorsPerTsPayload		
Parent Container	<a href="#">IKEMessageFormat</a>		
Description	The maximum number of traffic selectors a Traffic Selector payload may contain.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 2		
Default value	2		
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		

]

# [ECUC\_IKE\_00013] Definition of EcucIntegerParamDef IKEMaxTransformsPerProp

Parameter Name	IKEMaxTransformsPerProp		
Parent Container	<a href="#">IKEMessageFormat</a>		
Description	The maximum number of transforms a proposal may contain.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 5		
Default value	5		
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		

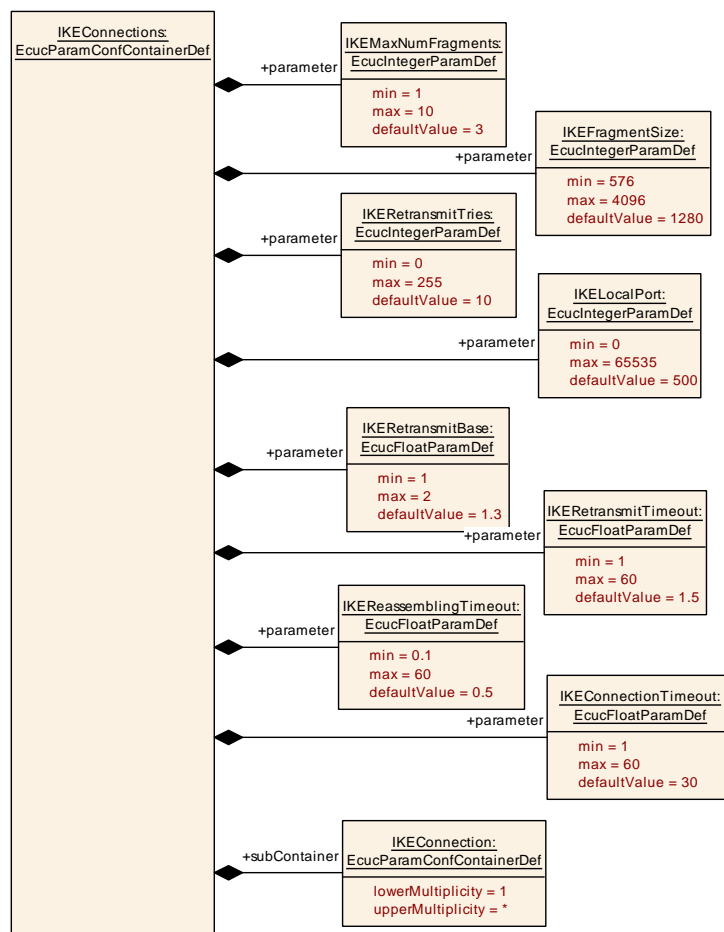


Figure 10.45: IKEConnections

## 10.2.67 IKEConnections

### [ECUC\_IKE\_00003] Definition of EcucParamConfContainerDef IKEConnections

<b>Container Name</b>	IKEConnections	
<b>Parent Container</b>	<a href="#">IKE</a>	
<b>Description</b>	Container for configuration of IKE connections.	
<b>Configuration Parameters</b>		

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">IKEConnectionTimeout</a>	1	<a href="#">[ECUC_IKE_00055]</a>
<a href="#">IKEFragmentSize</a>	1	<a href="#">[ECUC_IKE_00049]</a>
<a href="#">IKELocalPort</a>	1	<a href="#">[ECUC_IKE_00051]</a>
<a href="#">IKEMaxNumFragments</a>	1	<a href="#">[ECUC_IKE_00048]</a>
<a href="#">IKEReassemblingTimeout</a>	1	<a href="#">[ECUC_IKE_00054]</a>
<a href="#">IKERetransmitBase</a>	1	<a href="#">[ECUC_IKE_00052]</a>
<a href="#">IKERetransmitTimeout</a>	1	<a href="#">[ECUC_IKE_00053]</a>
<a href="#">IKERetransmitTries</a>	1	<a href="#">[ECUC_IKE_00050]</a>

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">IKEConnection</a>	1..*	Container for configuration of IKE connection.

### [ECUC\_IKE\_00055] Definition of EcucFloatParamDef IKEConnectionTimeout

Parameter Name	IKEConnectionTimeout		
Parent Container	<a href="#">IKEConnections</a>		
Description	Timeout for etsablishing a connection in order to handle a "half open" state.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[1 .. 60]		
Default value	30		
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		

## [ECUC\_IKE\_00049] Definition of EcucIntegerParamDef IKEFragmentSize [

Parameter Name	IKEFragmentSize		
Parent Container	<a href="#">IKEConnections</a>		
Description	The maximum size of IKE fragment messages when fragmentation is used. The resulting buffer size for subsequent fragment messages is (Number of Fragments * Fragment Size). This fragment size is the maximum IP datagram size, used for both RX and TX.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	576 .. 4096		
Default value	1280		
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		

]

## [ECUC\_IKE\_00051] Definition of EcucIntegerParamDef IKELocalPort [

Parameter Name	IKELocalPort		
Parent Container	<a href="#">IKEConnections</a>		
Description	The local port is the UDP port to listen to.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	500		
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		

]



## [ECUC\_IKE\_00048] Definition of EcucIntegerParamDef IKEMaxNumFragments [

Parameter Name	IKEMaxNumFragments		
Parent Container	<a href="#">IKEConnections</a>		
Description	The maximum number of fragment messages into which the an IKE message might be divided. If this value is set to 1, fragmentation is not supported. The resulting buffer size for subsequent fragment messages is (Number of Fragments * Fragment Size). Used for both RX and TX and affects size of TX buffer.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 10		
Default value	3		
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		

## [ECUC\_IKE\_00054] Definition of EcucFloatParamDef IKEReassemblingTimeout [

Parameter Name	IKEReassemblingTimeout		
Parent Container	<a href="#">IKEConnections</a>		
Description	The timeout for reassembling a fragmented message. All fragments of a message must be received within this interval, Otherwise all so far received fragments are discarded.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0.1 .. 60]		
Default value	0.5		
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		

## [ECUC\_IKE\_00052] Definition of EcucFloatParamDef IKERetransmitBase [

Parameter Name	IKERetransmitBase		
Parent Container	<a href="#">IKEConnections</a>		
Description	The base used for calculation of the exponantioal back-off of the retransmit timeouts.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[1 .. 2]		
Default value	1.3		
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		

## [ECUC\_IKE\_00053] Definition of EcucFloatParamDef IKERetransmitTimeout [

Parameter Name	IKERetransmitTimeout		
Parent Container	<a href="#">IKEConnections</a>		
Description	The initial retransmit timeout.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[1 .. 60]		
Default value	1.5		
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		

## [ECUC\_IKE\_00050] Definition of EcucIntegerParamDef IKERetransmitTries [

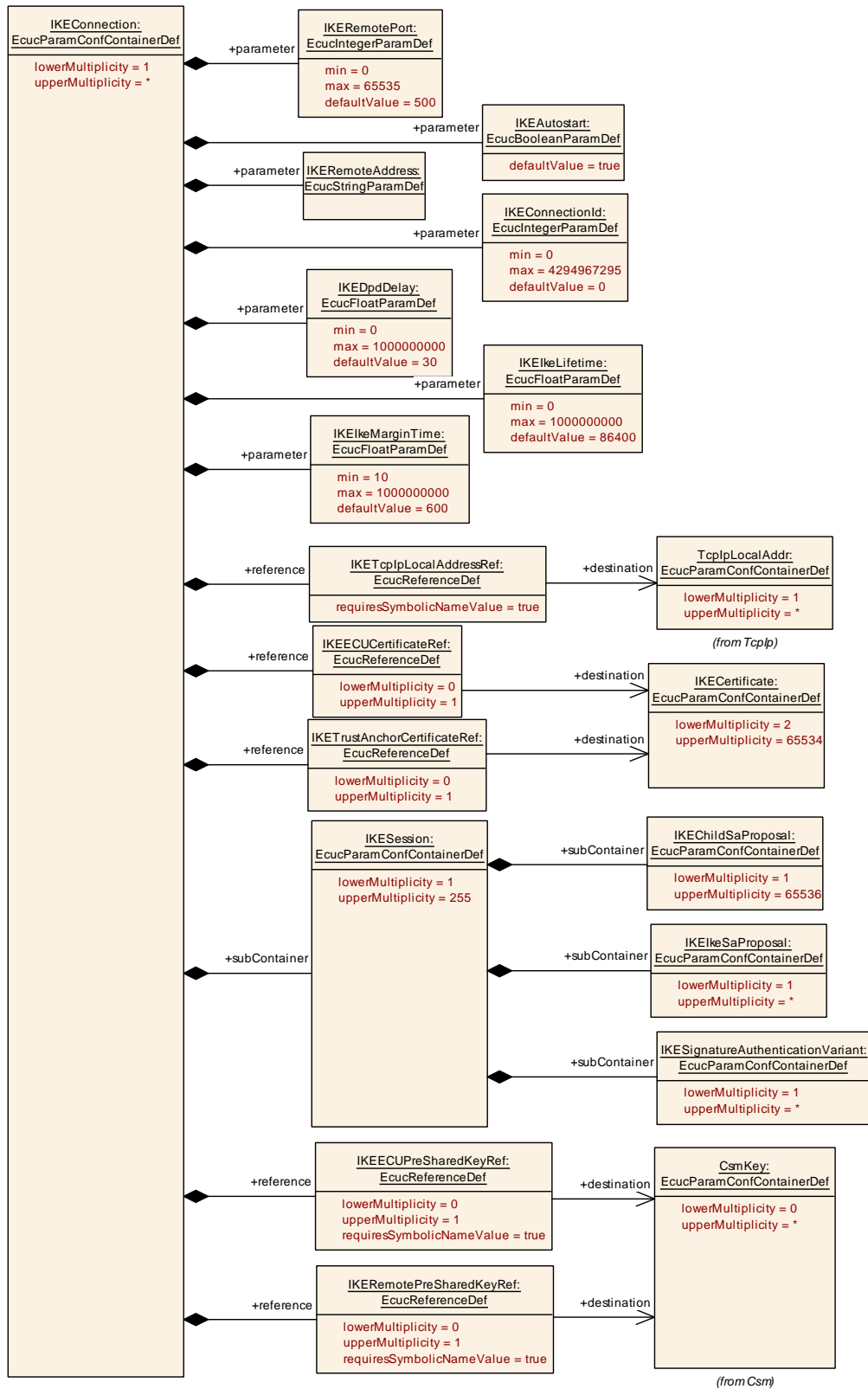
Parameter Name	IKERetransmitTries		
Parent Container	<a href="#">IKEConnections</a>		
Description	The maximum number of retransmits of a request before giving up.		



△

<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default value</b>	10		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

└



**Figure 10.46: IKEConnection**

## 10.2.68 IKEConnection

### [ECUC\_IKE\_00056] Definition of EcucParamConfContainerDef IKEConnection [

Container Name	IKEConnection		
Parent Container	<a href="#">IKEConnections</a>		
Description	Container for configuration of IKE connection.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">IKEAutostart</a>	1	<a href="#">[ECUC_IKE_00058]</a>
<a href="#">IKEConnectionId</a>	1	<a href="#">[ECUC_IKE_00060]</a>
<a href="#">IKEDpdDelay</a>	1	<a href="#">[ECUC_IKE_00063]</a>
<a href="#">IKEIkeLifetime</a>	1	<a href="#">[ECUC_IKE_00064]</a>
<a href="#">IKEIkeMarginTime</a>	1	<a href="#">[ECUC_IKE_00065]</a>
<a href="#">IKERemoteAddress</a>	1	<a href="#">[ECUC_IKE_00059]</a>
<a href="#">IKERemotePort</a>	1	<a href="#">[ECUC_IKE_00057]</a>
<a href="#">IKEECUCertificateRef</a>	0..1	<a href="#">[ECUC_IKE_00067]</a>
<a href="#">IKEECUPreSharedKeyRef</a>	0..1	<a href="#">[ECUC_IKE_00088]</a>
<a href="#">IKERemotePreSharedKeyRef</a>	0..1	<a href="#">[ECUC_IKE_00089]</a>
<a href="#">IKETcpIpLocalAddressRef</a>	1	<a href="#">[ECUC_IKE_00066]</a>
<a href="#">IKETrustAnchorCertificateRef</a>	0..1	<a href="#">[ECUC_IKE_00068]</a>

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">IKESession</a>	1..255	Container for configuration of IKE session.

]

### [ECUC\_IKE\_00058] Definition of EcucBooleanParamDef IKEAutostart [

Parameter Name	IKEAutostart		
Parent Container	<a href="#">IKEConnection</a>		
Description	If enabled, IKE wil automatically initiate an IKE SA on this connection after start-up of the module.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	true		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants





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

### [ECUC\_IKE\_00060] Definition of EcucIntegerParamDef IKEConnectionId [

Parameter Name	IKEConnectionId		
Parent Container	<a href="#">IKEConnection</a>		
Description	Identifier of the connection.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	0		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

### [ECUC\_IKE\_00063] Definition of EcucFloatParamDef IKEDpdDelay [

Parameter Name	IKEDpdDelay		
Parent Container	<a href="#">IKEConnection</a>		
Description	Specifies the interval in which Dead Peer Detection (DPD) packets shall be sent in the absence of other traffic Set to 0 to disable sending DPD packets.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. 1000000000]		
Default value	30		
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		

]

#### [ECUC\_IKE\_00064] Definition of EcucFloatParamDef IKEIkeLifetime [

Parameter Name	IKEIkeLifetime		
Parent Container	<a href="#">IKEConnection</a>		
Description	Specifies the time after which an IKE SA is terminated. Set to 0 if IKE SA never expires.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. 1000000000]		
Default value	86400		
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		

]

#### [ECUC\_IKE\_00065] Definition of EcucFloatParamDef IKEIkeMarginTime [

Parameter Name	IKEIkeMarginTime		
Parent Container	<a href="#">IKEConnection</a>		
Description	Specifies how many seconds before expiry an IKE SA should be renegotiated.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[10 .. 1000000000]		
Default value	600		
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		

]

## [ECUC\_IKE\_00059] Definition of EcucStringParamDef IKERemoteAddress [

Parameter Name	IKERemoteAddress		
Parent Container	<a href="#">IKEConnection</a>		
Description	The remote address is the IP address of the ECU which a IKE connection shall be established with, e.g. 192.168.50.101.		
Multiplicity	1		
Type	EcucStringParamDef		
Default value	–		
Regular Expression	–		
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		

## [ECUC\_IKE\_00057] Definition of EcucIntegerParamDef IKERemotePort [

Parameter Name	IKERemotePort		
Parent Container	<a href="#">IKEConnection</a>		
Description	The remote port is the UDP port of the ECU which a IKE connection shall be established with.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	500		
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		



### [ECUC\_IKE\_00067] Definition of EcucReferenceDef IKEECUCertificateRef [

Parameter Name	IKEECUCertificateRef		
Parent Container	<a href="#">IKEConnection</a>		
Description	The ECU certificate is the end-entity certificate. The referenced certificate is the ECU certificate which contains the public key used for authentication during the IKE connection setup.		
Multiplicity	0..1		
Type	Reference to <a href="#">IKECertificate</a>		
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		

]

### [ECUC\_IKE\_00088] Definition of EcucReferenceDef IKEECUPreSharedKeyRef [

Parameter Name	IKEECUPreSharedKeyRef		
Parent Container	<a href="#">IKEConnection</a>		
Description	The ECU's pre-shared key.		
Multiplicity	0..1		
Type	Symbolic name reference to CsmKey		
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		

]

### [ECUC\_IKE\_00089] Definition of EcucReferenceDef IKERemotePreSharedKeyRef [

Parameter Name	IKERemotePreSharedKeyRef		
Parent Container	<a href="#">IKEConnection</a>		
Description	The referenced key is the key which is used to identify the remote ECU during the IKE connection setup.		
Multiplicity	0..1		





Type	Symbolic name reference to CsmKey		
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		

### [ECUC\_IKE\_00066] Definition of EcucReferenceDef IKETcpIpLocalAddressRef

Parameter Name	IKETcpIpLocalAddressRef		
Parent Container	<a href="#">IKEConnection</a>		
Description	IP address table identifier assigned by TCP/IP stack.		
Multiplicity	1		
Type	Symbolic name reference to <a href="#">TcpIpLocalAddr</a>		
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		

### [ECUC\_IKE\_00068] Definition of EcucReferenceDef IKETrustAnchorCertificateRef

Parameter Name	IKETrustAnchorCertificateRef		
Parent Container	<a href="#">IKEConnection</a>		
Description	The referenced certificate is the Trust Anchor certificate which is used to identify the trusted Certification Authorities during the IKE connection setup.		
Multiplicity	0..1		
Type	Reference to <a href="#">IKECertificate</a>		
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		

]

## 10.2.69 IKESession

### [ECUC\_IKE\_00069] Definition of EcucParamConfContainerDef IKESession [

Container Name	IKESession		
Parent Container	<a href="#">IKEConnection</a>		
Description	Container for configuration of IKE session.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

#### No Included Parameters

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">IKEChildSaProposal</a>	1..65536	Container for configuration of IKE Authentication Header (AH) or Encapsulating Security Payload (ESP) Security Association Proposals.
<a href="#">IKEIkeSaProposal</a>	1..*	Container for configuration of IKE IKE Security Association Proposal.
<a href="#">IKESignatureAuthenticationVariant</a>	1..*	Defining variants for the IKEv2 Authentication Method "Digital Signature".

]

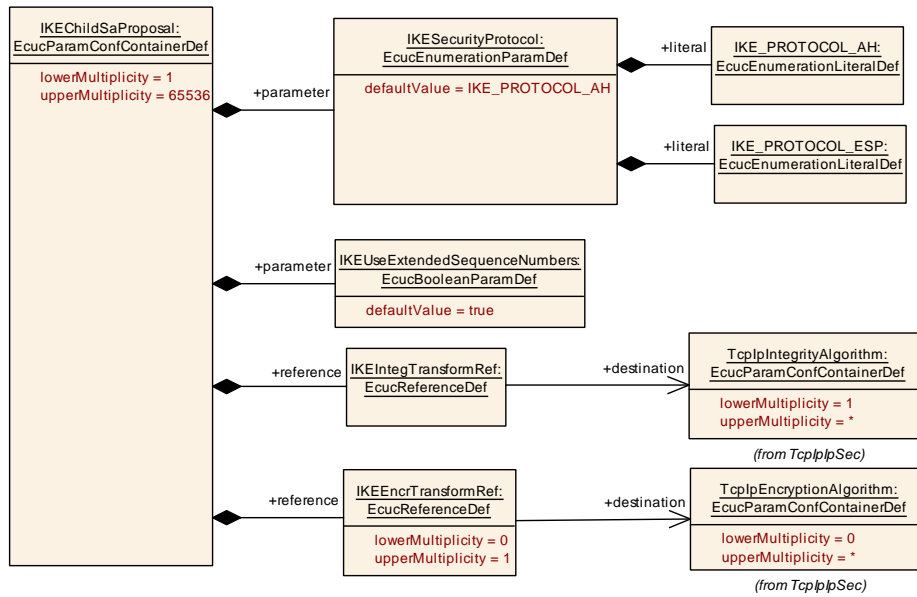


Figure 10.47: IKEChildSaProposal

## 10.2.70 IKEChildSaProposal

### [ECUC\_IKE\_00070] Definition of EcucParamConfContainerDef IKEChildSaProposal

Container Name	IKEChildSaProposal		
Parent Container	IKESession		
Description	Container for configuration of IKE Authentication Header (AH) or Encapsulating Security Payload (ESP) Security Association Proposals.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	—	
	Post-build time	—	
Configuration Parameters			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
IKESecurityProtocol	1	[ECUC_IKE_00074]
IKEUseExtendedSequenceNumbers	1	[ECUC_IKE_00075]
IKEEncrTransformRef	0..1	[ECUC_IKE_00077]
IKEIntegTransformRef	1	[ECUC_IKE_00076]

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

]

## [ECUC\_IKE\_00074] Definition of EcucEnumerationParamDef IKESecurityProtocol

Parameter Name	IKESecurityProtocol		
Parent Container	<a href="#">IKEChildSaProposal</a>		
Description	The security protocol (i.e., AH or ESP) to be used for this Child SA.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	IKE_PROTOCOL_AH	–	
	IKE_PROTOCOL_ESP	–	
Default value	<a href="#">IKE_PROTOCOL_AH</a>		
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		

## [ECUC\_IKE\_00075] Definition of EcucBooleanParamDef IKEUseExtendedSequenceNumbers

Parameter Name	IKEUseExtendedSequenceNumbers		
Parent Container	<a href="#">IKEChildSaProposal</a>		
Description	Whether this Child SA should use Extended Sequence Numbers (ESN), i.e., 64-Bit instead of 32-Bit sequence numbers.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	true		
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		

## [ECUC\_IKE\_00077] Definition of EcucReferenceDef IKEEncrTransformRef [

Parameter Name	IKEEncrTransformRef		
Parent Container	<a href="#">IKEChildSaProposal</a>		
Description	The referenced Encryption Algorithm is added to this proposal. Leave empty for AH and ESP in authentication-only mode.		
Multiplicity	0..1		
Type	Reference to <a href="#">TcplpEncryptionAlgorithm</a>		
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		

## [ECUC\_IKE\_00076] Definition of EcucReferenceDef IKEIntegTransformRef [

Parameter Name	IKEIntegTransformRef		
Parent Container	<a href="#">IKEChildSaProposal</a>		
Description	The referenced Integrity Algorithm is added to this proposal.		
Multiplicity	1		
Type	Reference to <a href="#">TcplpIntegrityAlgorithm</a>		
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		

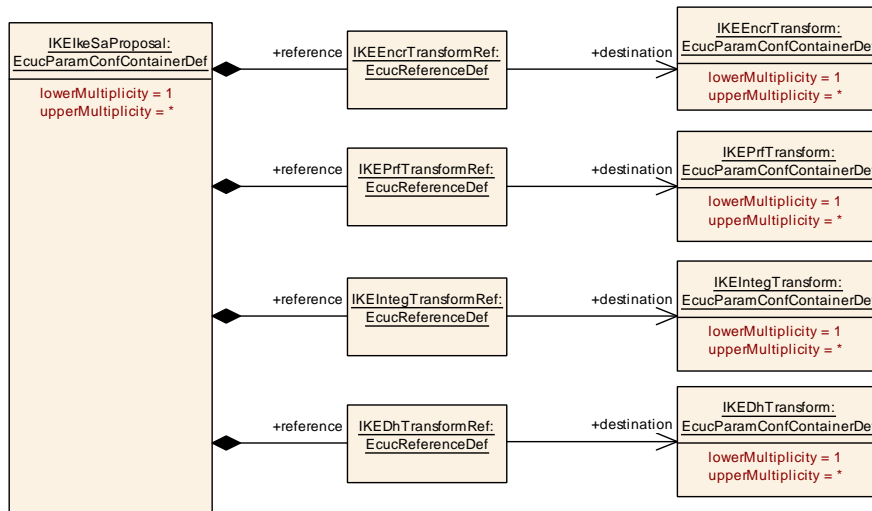


Figure 10.48: IKEIkeSaProposal

### 10.2.71 IKEIkeSaProposal

#### [ECUC\_IKE\_00071] Definition of EcucParamConfContainerDef IKEIkeSaProposal

Container Name	IKEIkeSaProposal		
Parent Container	IKESession		
Description	Container for configuration of IKE IKE Security Association Proposal.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
IKEDhTransformRef	1	[ECUC_IKE_00082]
IKEEncrTransformRef	1	[ECUC_IKE_00079]
IKEIntegTransformRef	1	[ECUC_IKE_00081]
IKEPrfTransformRef	1	[ECUC_IKE_00080]

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

### [ECUC\_IKE\_00082] Definition of EcucReferenceDef IKEDhTransformRef [

Parameter Name	IKEDhTransformRef		
Parent Container	<a href="#">IKEIkeSaProposal</a>		
Description	The referenced Diffie-Hellman Group Transform is added to this proposal.		
Multiplicity	1		
Type	Reference to <a href="#">IKEDhTransform</a>		
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		

]

### [ECUC\_IKE\_00079] Definition of EcucReferenceDef IKEEncrTransformRef [

Parameter Name	IKEEncrTransformRef		
Parent Container	<a href="#">IKEIkeSaProposal</a>		
Description	The referenced Encryption Algorithm Transform is added to this proposal.		
Multiplicity	1		
Type	Reference to <a href="#">IKEEncrTransform</a>		
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		

]

### [ECUC\_IKE\_00081] Definition of EcucReferenceDef IKEIntegTransformRef [

Parameter Name	IKEIntegTransformRef		
Parent Container	<a href="#">IKEIkeSaProposal</a>		
Description	The referenced Integrity Algorithm Transform is added to this proposal.		
Multiplicity	1		
Type	Reference to <a href="#">IKEIntegTransform</a>		
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		

]

**[ECUC\_IKE\_00080] Definition of EcucReferenceDef IKEPrfTransformRef [**

Parameter Name	IKEPrfTransformRef		
Parent Container	<a href="#">IKEIkeSaProposal</a>		
Description	The referenced Pseudorandom Function Transform is added to this proposal.		
Multiplicity	1		
Type	Reference to <a href="#">IKEPrfTransform</a>		
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		

]

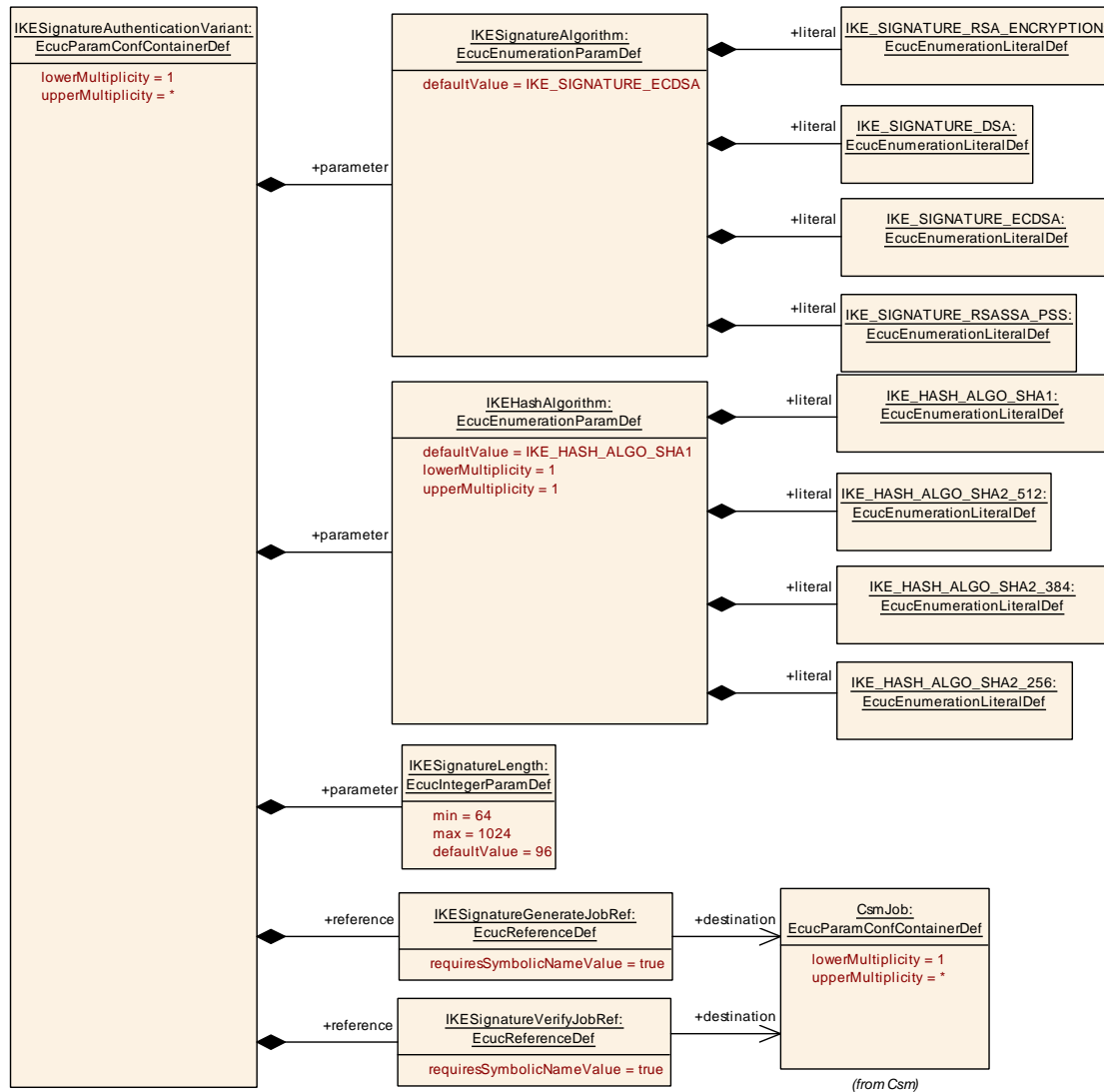


Figure 10.49: IKESignatureAuthenticationVariant

## 10.2.72 IKESignatureAuthenticationVariant

[ECUC\_IKE\_00072] Definition of EcucParamConfContainerDef IKESignatureAuthenticationVariant

Container Name	IKESignatureAuthenticationVariant		
Parent Container	IKESession		
Description	Defining variants for the IKEv2 Authentication Method "Digital Signature".		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	





### Configuration Parameters

#### Included Parameters

Parameter Name	Multiplicity	ECUC ID
<a href="#">IKEHashAlgorithm</a>	1	<a href="#">[ECUC_IKE_00084]</a>
<a href="#">IKESignatureAlgorithm</a>	1	<a href="#">[ECUC_IKE_00083]</a>
<a href="#">IKESignatureLength</a>	1	<a href="#">[ECUC_IKE_00085]</a>
<a href="#">IKESignatureGenerateJobRef</a>	1	<a href="#">[ECUC_IKE_00086]</a>
<a href="#">IKESignatureVerifyJobRef</a>	1	<a href="#">[ECUC_IKE_00087]</a>

#### No Included Containers

]

### [ECUC\_IKE\_00084] Definition of EcucEnumerationParamDef IKEHashAlgorithm

[

Parameter Name	IKEHashAlgorithm		
Parent Container	<a href="#">IKESignatureAuthenticationVariant</a>		
Description	Pre-hashing Algorithm. Please adapt to the referenced Csm jobs.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	IKE_HASH_ALGO_SHA1	–	
	IKE_HASH_ALGO_SHA2_256	–	
	IKE_HASH_ALGO_SHA2_384	–	
	IKE_HASH_ALGO_SHA2_512	–	
Default value	<a href="#">IKE_HASH_ALGO_SHA1</a>		
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		

]

## [ECUC\_IKE\_00083] Definition of EcucEnumerationParamDef IKESignatureAlgorithm

Parameter Name	IKESignatureAlgorithm		
Parent Container	IKESignatureAuthenticationVariant		
Description	Signature Algorithm. Please adapt to the referenced Csm jobs.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	IKE_SIGNATURE_DSA	–	
	IKE_SIGNATURE_ECDSA	–	
	IKE_SIGNATURE_RSASSA_PSS	–	
	IKE_SIGNATURE_RSA_ENCRYPTION	–	
Default value	IKESignatureECDSA		
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		

## [ECUC\_IKE\_00085] Definition of EcucIntegerParamDef IKESignatureLength

Parameter Name	IKESignatureLength		
Parent Container	IKESignatureAuthenticationVariant		
Description	The length of a signature generated by the configured generation job and verified by the configured verification job. E.g. 64 for ECDSA-256 or 96 for ECDSA-386.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	64 .. 1024		
Default value	96		
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		

## [ECUC\_IKE\_00086] Definition of EcucReferenceDef IKESignatureGenerateJobRef [

Parameter Name	IKESignatureGenerateJobRef		
Parent Container	<a href="#">IKESignatureAuthenticationVariant</a>		
Description	The referenced Csm job is used for the execution of the CsmSignatureGenerate primitive needed for this transform.		
Multiplicity	1		
Type	Symbolic name reference to CsmJob		
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		

]

## [ECUC\_IKE\_00087] Definition of EcucReferenceDef IKESignatureVerifyJobRef [

Parameter Name	IKESignatureVerifyJobRef		
Parent Container	<a href="#">IKESignatureAuthenticationVariant</a>		
Description	The referenced Csm job is used for the execution of the CsmSignatureVerify primitive needed for this transform.		
Multiplicity	1		
Type	Symbolic name reference to CsmJob		
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		

]

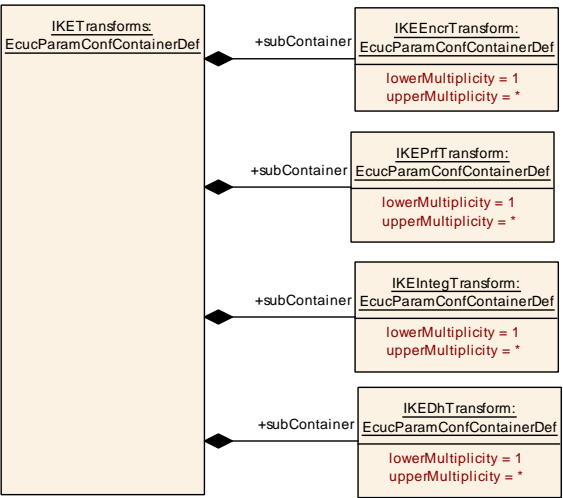


Figure 10.50: IKETransforms

10.2.73 IKETransforms

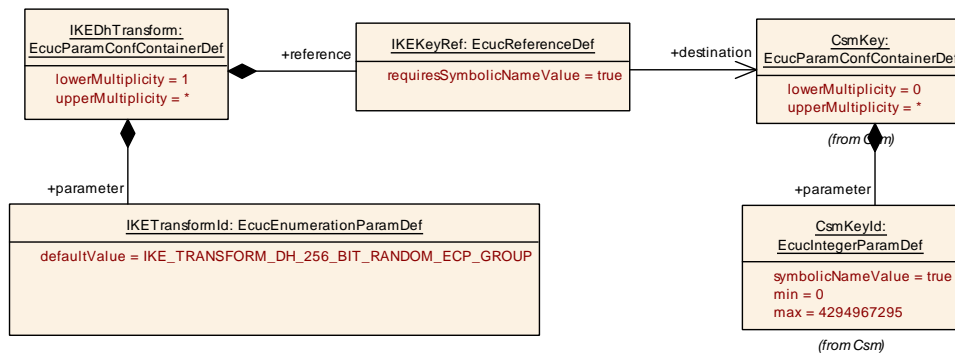
[ECUC\_IKE\_00004] Definition of EcucParamConfContainerDef IKETransforms [

Container Name	IKETransforms
Parent Container	IKE
Description	Container for configuration of IKE transforms.
Configuration Parameters	

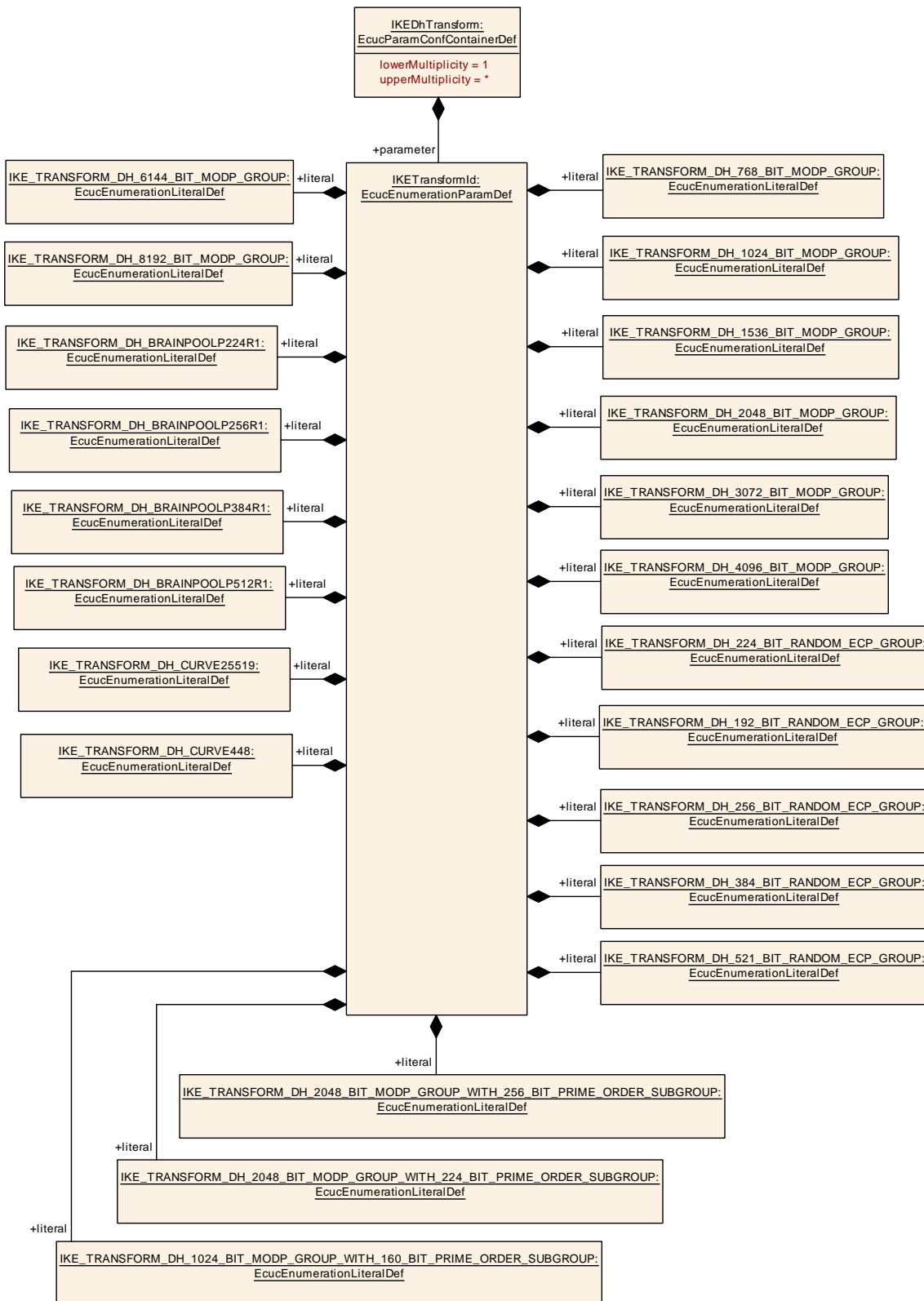
No Included Parameters
------------------------

Included Containers		
Container Name	Multiplicity	Scope / Dependency
IKEDhTransform	1..*	Container for configuration of Diffie-Hellman Group Transform.
IKEEncrTransform	1..*	Container for configuration of Encryption Algorithm Transform.
IKEIntegTransform	1..*	Container for configuration of Integrity Algorithm Transform.
IKEPrfTransform	1..*	Container for configuration of Pseudorandom Function Transform.

]



**Figure 10.51: IKEDhTransform**



**Figure 10.52: IKEEDhTransformIkeTransformId**



## 10.2.74 IKEDhTransform

### [ECUC\_IKE\_00028] Definition of EcucParamConfContainerDef IKEDhTransform

Container Name	IKEDhTransform		
Parent Container	<a href="#">IKETransforms</a>		
Description	Container for configuration of Diffie-Hellman Group Transform.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">IKETransformId</a>	1	<a href="#">[ECUC_IKE_00038]</a>
<a href="#">IKEKeyRef</a>	1	<a href="#">[ECUC_IKE_00039]</a>

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

### [ECUC\_IKE\_00038] Definition of EcucEnumerationParamDef IKETransformId

<b>Parameter Name</b>	IKETransformId	
<b>Parent Container</b>	<a href="#">IKEDhTransform</a>	
<b>Description</b>	Diffie-Hellman Group Transform ID.	
<b>Multiplicity</b>	1	
<b>Type</b>	EcucEnumerationParamDef	
<b>Range</b>	IKE_TRANSFORM_DH_1024_BIT_MODP_GROUP	–
	IKE_TRANSFORM_DH_1024_BIT_MODP_GROUP_WITH_160_BIT_PRIME_ORDER_SUBGROUP	–
	IKE_TRANSFORM_DH_1536_BIT_MODP_GROUP	–
	IKE_TRANSFORM_DH_192_BIT_RANDOM_ECP_GROUP	–
	IKE_TRANSFORM_DH_2048_BIT_MODP_GROUP	–
	IKE_TRANSFORM_DH_2048_BIT_MODP_GROUP_WITH_224_BIT_PRIME_ORDER_SUBGROUP	–



△

	IKE_TRANSFORM_DH_2048_BIT_MODP_GROUP_WITH_256_BIT_PRIME_ORDER_SUBGROUP	–	
	IKE_TRANSFORM_DH_224_BIT_RANDOM_ECP_GROUP	–	
	IKE_TRANSFORM_DH_256_BIT_RANDOM_ECP_GROUP	–	
	IKE_TRANSFORM_DH_3072_BIT_MODP_GROUP	–	
	IKE_TRANSFORM_DH_384_BIT_RANDOM_ECP_GROUP	–	
	IKE_TRANSFORM_DH_4096_BIT_MODP_GROUP	–	
	IKE_TRANSFORM_DH_521_BIT_RANDOM_ECP_GROUP	–	
	IKE_TRANSFORM_DH_6144_BIT_MODP_GROUP	–	
	IKE_TRANSFORM_DH_768_BIT_MODP_GROUP	–	
	IKE_TRANSFORM_DH_8192_BIT_MODP_GROUP	–	
	IKE_TRANSFORM_DH_BRAINPOOLP224R1	–	
	IKE_TRANSFORM_DH_BRAINPOOLP256R1	–	
	IKE_TRANSFORM_DH_BRAINPOOLP384R1	–	
	IKE_TRANSFORM_DH_BRAINPOOLP512R1	–	
	IKE_TRANSFORM_DH_CURVE25519	–	
	IKE_TRANSFORM_DH_CURVE448	–	
Default value	IKE_TRANSFORM_DH_256_BIT_RANDOM_ECP_GROUP		
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		

┌

[ECUC\_IKE\_00039] Definition of EcucReferenceDef IKEKeyRef [

Parameter Name	IKEKeyRef		
Parent Container	IKEDhTransform		
Description	The referenced Csm key is used for the execution of key management functions needed for this transform.		
Multiplicity	1		
Type	Symbolic name reference to CsmKey		
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		

]

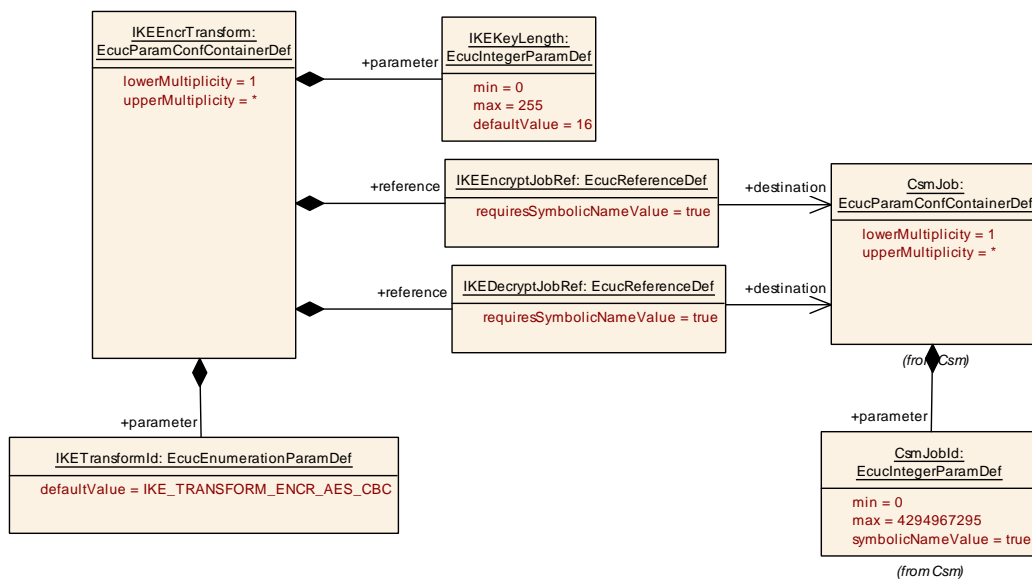


Figure 10.53: IKEEncrTransform

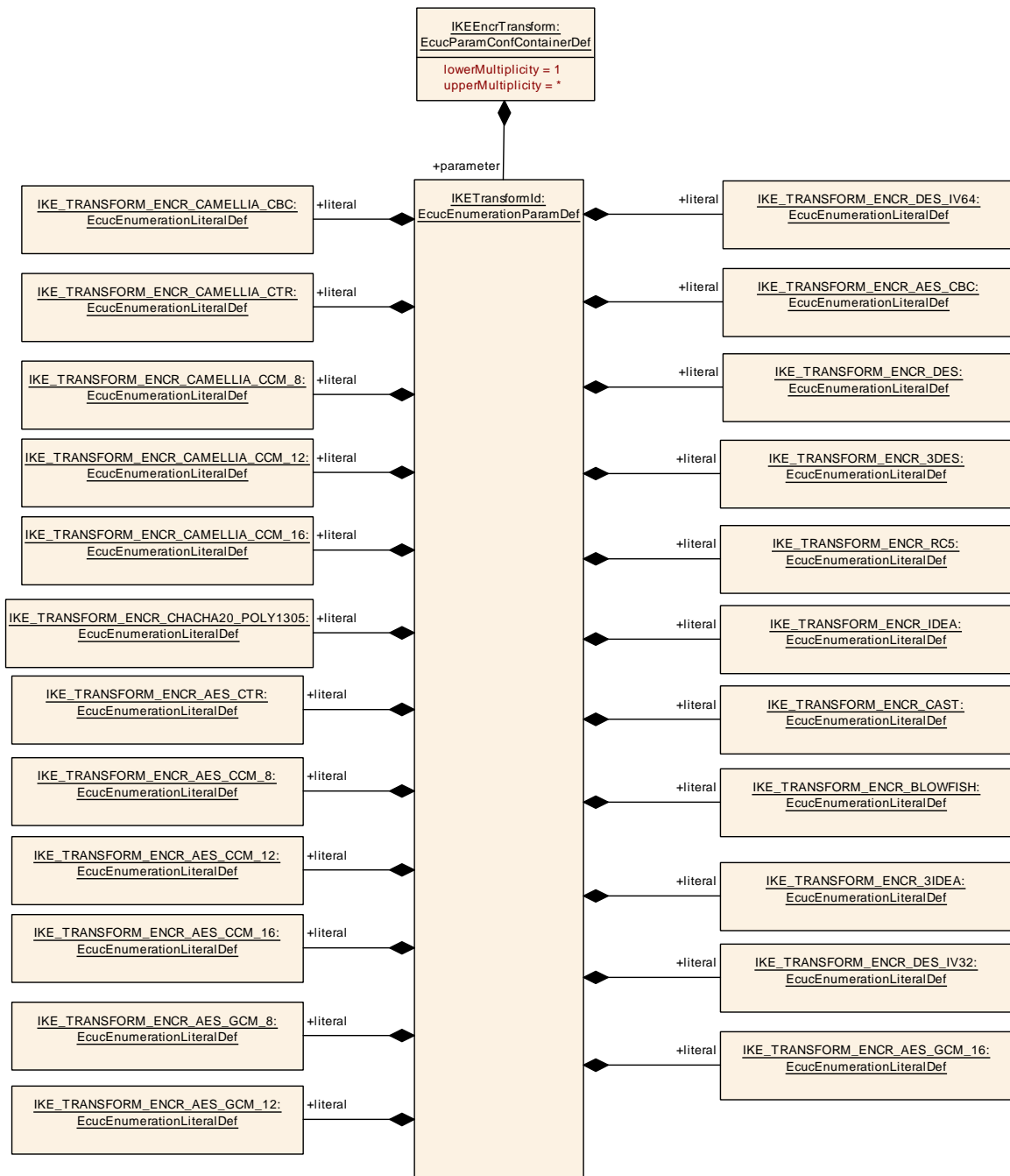


Figure 10.54: IKEEncrTransformIKETransformId

## 10.2.75 IKEEncrTransform

[ECUC\_IKE\_00025] Definition of EcucParamConfContainerDef IKEEncrTransform

Container Name	IKEEncrTransform		
Parent Container	<a href="#">IKETransforms</a>		
Description	Container for configuration of Encryption Algorithm Transform.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">IKEKeyLength</a>	1	<a href="#">[ECUC_IKE_00030]</a>
<a href="#">IKETransformId</a>	1	<a href="#">[ECUC_IKE_00029]</a>
<a href="#">IKEDecryptJobRef</a>	1	<a href="#">[ECUC_IKE_00032]</a>
<a href="#">IKEEncryptJobRef</a>	1	<a href="#">[ECUC_IKE_00031]</a>

No Included Containers

]

### [ECUC\_IKE\_00030] Definition of EcucIntegerParamDef IKEKeyLength [

Parameter Name	IKEKeyLength		
Parent Container	<a href="#">IKEEncrTransform</a>		
Description	The key length of the encryption algorithm in bytes.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	16		
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		

]

### [ECUC\_IKE\_00029] Definition of EcucEnumerationParamDef IKETransformId [

Parameter Name	IKETransformId		
Parent Container	<a href="#">IKEEncrTransform</a>		
Description	Encryption Algorithm Transform ID.		





<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	IKE_TRANSFORM_ENCR_3DES	–	
	IKE_TRANSFORM_ENCR_3IDEA	–	
	IKE_TRANSFORM_ENCR_AES_CBC	–	
	IKE_TRANSFORM_ENCR_AES_CCM_12	–	
	IKE_TRANSFORM_ENCR_AES_CCM_16	–	
	IKE_TRANSFORM_ENCR_AES_CCM_8	–	
	IKE_TRANSFORM_ENCR_AES_CTR	–	
	IKE_TRANSFORM_ENCR_AES_GCM_12	–	
	IKE_TRANSFORM_ENCR_AES_GCM_16	–	
	IKE_TRANSFORM_ENCR_AES_GCM_8	–	
	IKE_TRANSFORM_ENCR_BLOWFISH	–	
	IKE_TRANSFORM_ENCR_CAMELLIA_CBC	–	
	IKE_TRANSFORM_ENCR_CAMELLIA_CCM_12	–	
	IKE_TRANSFORM_ENCR_CAMELLIA_CCM_16	–	
	IKE_TRANSFORM_ENCR_CAMELLIA_CCM_8	–	
	IKE_TRANSFORM_ENCR_CAMELLIA_CTR	–	
	IKE_TRANSFORM_ENCR_CAST	–	
	IKE_TRANSFORM_ENCR_CHACHA20_POLY1305	–	
	IKE_TRANSFORM_ENCR_DES	–	
	IKE_TRANSFORM_ENCR_DES_IV32	–	
	IKE_TRANSFORM_ENCR_DES_IV64	–	
	IKE_TRANSFORM_ENCR_IDEA	–	
	IKE_TRANSFORM_ENCR_RC5	–	
<b>Default value</b>	IKE_TRANSFORM_ENCR_AES_CBC		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	





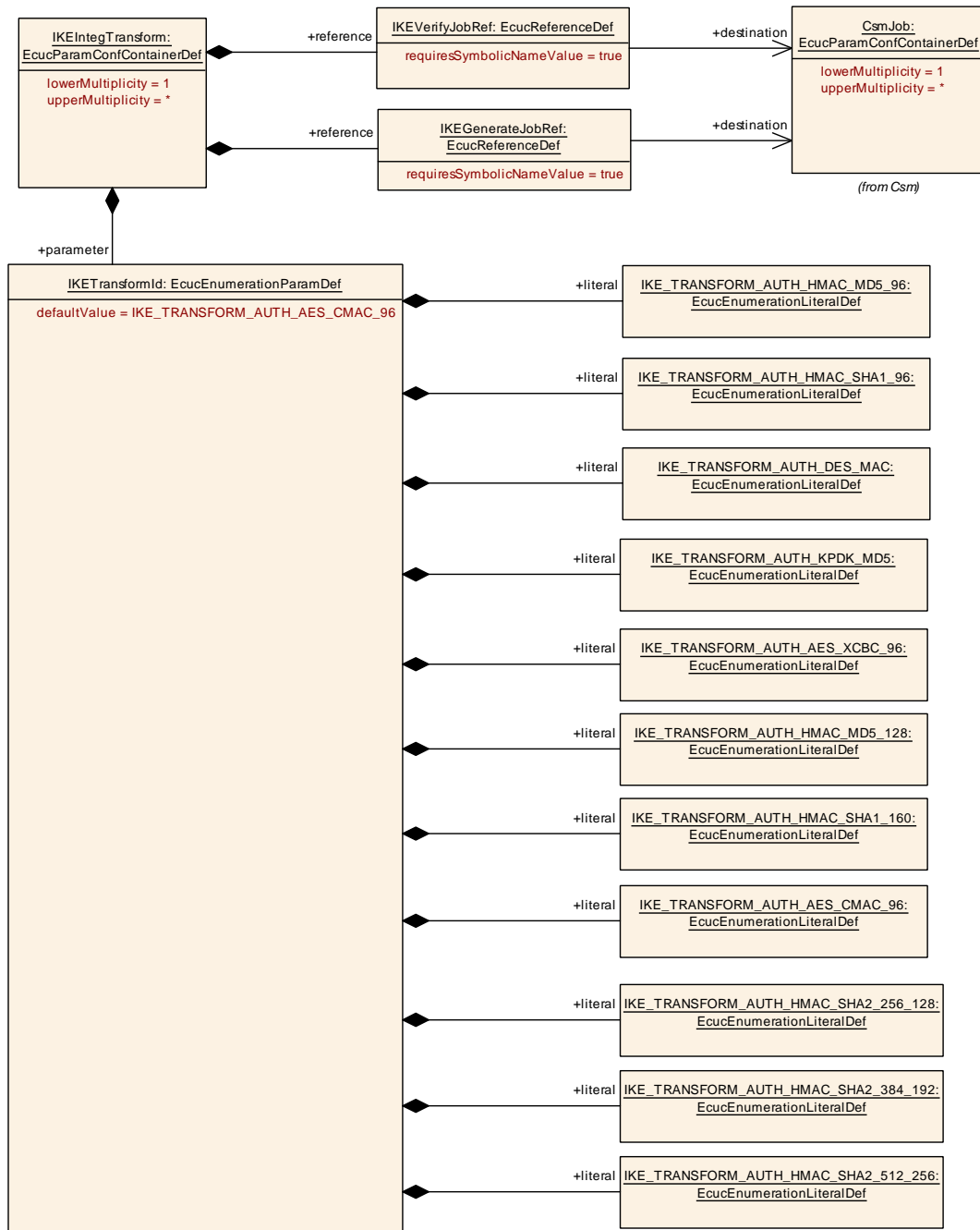
	Post-build time	–	
Scope / Dependency	scope: local		

### [ECUC\_IKE\_00032] Definition of EcucReferenceDef IKEDecryptJobRef [

Parameter Name	IKEDecryptJobRef		
Parent Container	<a href="#">IKEEncrTransform</a>		
Description	The referenced Csm job is used for the execution of the CsmDecrypt primitive needed for this transform.		
Multiplicity	1		
Type	Symbolic name reference to CsmJob		
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		

### [ECUC\_IKE\_00031] Definition of EcucReferenceDef IKEEncryptJobRef [

Parameter Name	IKEEncryptJobRef		
Parent Container	<a href="#">IKEEncrTransform</a>		
Description	The referenced Csm job is used for the execution of the CsmEncrypt primitive needed for this transform.		
Multiplicity	1		
Type	Symbolic name reference to CsmJob		
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		



**Figure 10.55: IKEIntegTransform**

### 10.2.76 IKEIntegTransform

**[ECUC\_IKE\_00027] Definition of EcucParamConfContainerDef IKEIntegTransform**



Container Name	IKEIntegTransform		
Parent Container	<a href="#">IKETransforms</a>		
Description	Container for configuration of Integrity Algorithm Transform.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">IKETransformId</a>	1	<a href="#">[ECUC_IKE_00037]</a>
<a href="#">IKEGenerateJobRef</a>	1	<a href="#">[ECUC_IKE_00036]</a>
<a href="#">IKEVerifyJobRef</a>	1	<a href="#">[ECUC_IKE_00035]</a>

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

]

## [ECUC\_IKE\_00037] Definition of EcucEnumerationParamDef IKETransformId [

Parameter Name	IKETransformId	
Parent Container	<a href="#">IKEIntegTransform</a>	
Description	Integrity Algorithm Transform ID.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	IKE_TRANSFORM_AUTH_AES_CMACH_96	–
	IKE_TRANSFORM_AUTH_AES_XCBC_96	–
	IKE_TRANSFORM_AUTH_DES_MAC	–
	IKE_TRANSFORM_AUTH_HMAC_MD5_128	–
	IKE_TRANSFORM_AUTH_HMAC_MD5_96	–
	IKE_TRANSFORM_AUTH_HMAC_SHA1_160	–
	IKE_TRANSFORM_AUTH_HMAC_SHA1_96	–
	IKE_TRANSFORM_AUTH_HMAC_SHA2_256_128	–
	IKE_TRANSFORM_AUTH_HMAC_SHA2_384_192	–
	IKE_TRANSFORM_AUTH_HMAC_SHA2_512_256	–
	IKE_TRANSFORM_AUTH_KPDK_MD5	–
Default value	<a href="#">IKE_TRANSFORM_AUTH_AES_CMACH_96</a>	





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

### [ECUC\_IKE\_00036] Definition of EcucReferenceDef IKEGenerateJobRef [

<b>Parameter Name</b>	IKEGenerateJobRef		
<b>Parent Container</b>	<a href="#">IKEIntegTransform</a>		
<b>Description</b>	The referenced Csm job is used for the execution of the CsmMacGenerate primitive needed for this transform.		
<b>Multiplicity</b>	1		
<b>Type</b>	Symbolic name reference to CsmJob		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

### [ECUC\_IKE\_00035] Definition of EcucReferenceDef IKEVerifyJobRef [

<b>Parameter Name</b>	IKEVerifyJobRef		
<b>Parent Container</b>	<a href="#">IKEIntegTransform</a>		
<b>Description</b>	The referenced Csm job is used for the execution of the CsmMacVerify primitive needed for this transform.		
<b>Multiplicity</b>	1		
<b>Type</b>	Symbolic name reference to CsmJob		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants





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

└

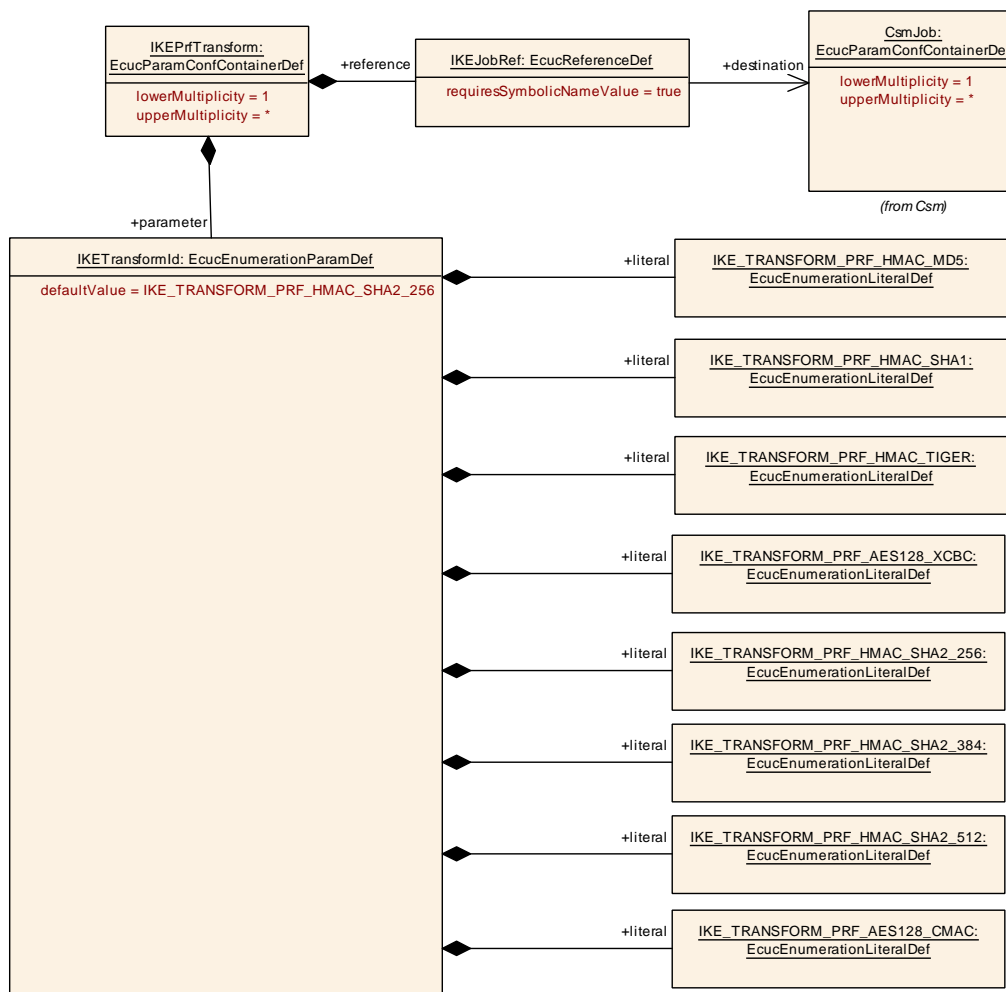


Figure 10.56: IKEPrfTransform

## 10.2.77 IKEPrfTransform

[ECUC\_IKE\_00026] Definition of EcucParamConfContainerDef IKEPrfTransform

└

Container Name	IKEPrfTransform		
Parent Container	<a href="#">IKETransforms</a>		
Description	Container for configuration of Pseudorandom Function Transform.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">IKETransformId</a>	1	<a href="#">[ECUC_IKE_00033]</a>
<a href="#">IKEJobRef</a>	1	<a href="#">[ECUC_IKE_00034]</a>

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

## [ECUC\_IKE\_00033] Definition of EcucEnumerationParamDef IKETransformId [

Parameter Name	IKETransformId		
Parent Container	<a href="#">IKEPrfTransform</a>		
Description	Pseudorandom Function Transform ID.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	IKE_TRANSFORM_PRF_AES128_CMAC	—	
	IKE_TRANSFORM_PRF_AES128_XCBC	—	
	IKE_TRANSFORM_PRF_HMAC_MD5	—	
	IKE_TRANSFORM_PRF_HMAC_SHA1	—	
	IKE_TRANSFORM_PRF_HMAC_SHA2_256	—	
	IKE_TRANSFORM_PRF_HMAC_SHA2_384	—	
	IKE_TRANSFORM_PRF_HMAC_SHA2_512	—	
	IKE_TRANSFORM_PRF_HMAC_TIGER	—	
Default value	<a href="#">IKE_TRANSFORM_PRF_HMAC_SHA2_256</a>		
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	—	





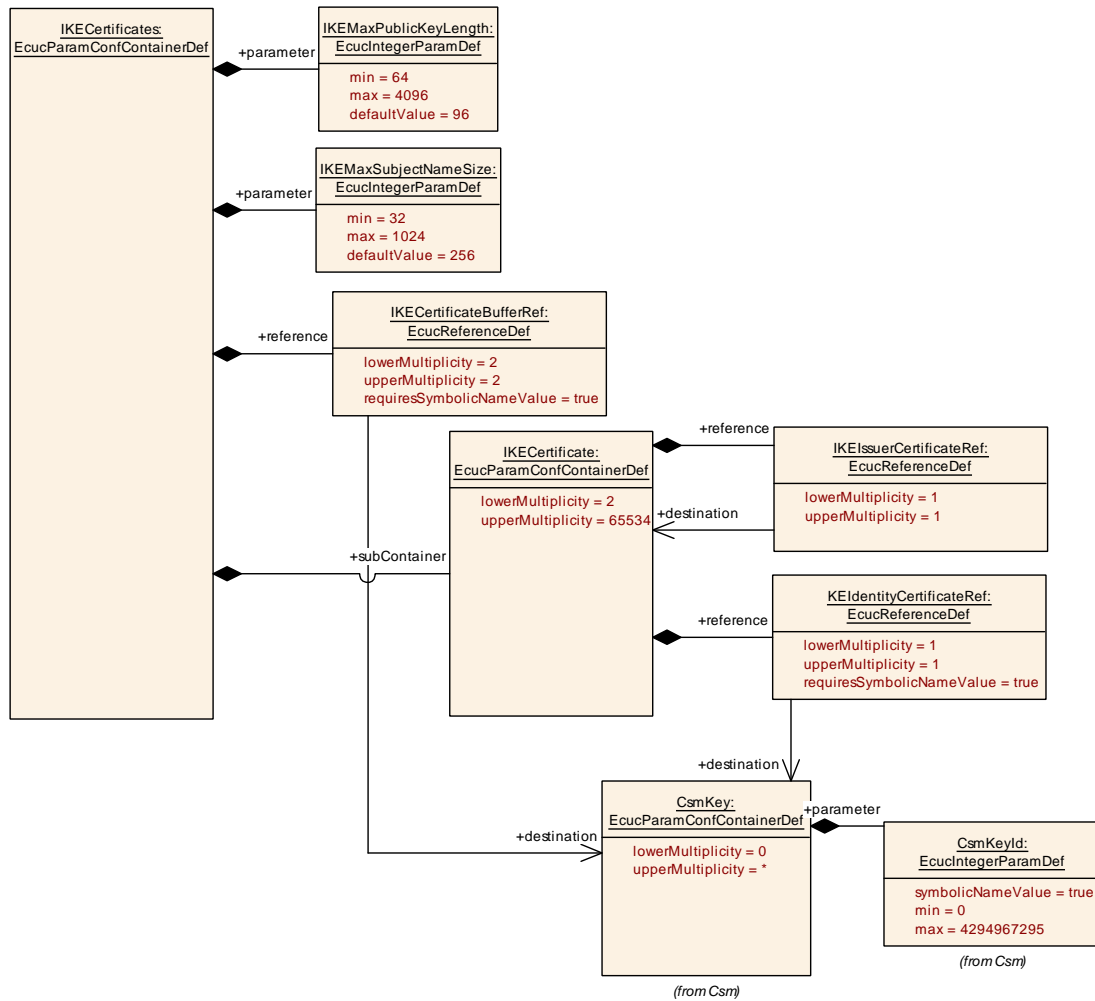
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

]

**[ECUC\_IKE\_00034] Definition of EcucReferenceDef IKEJobRef** [

<b>Parameter Name</b>	IKEJobRef		
<b>Parent Container</b>	<a href="#">IKEPrfTransform</a>		
<b>Description</b>	The referenced Csm job is used for the execution of the CsmMacGenerate primitive needed for this transform.		
<b>Multiplicity</b>	1		
<b>Type</b>	Symbolic name reference to CsmJob		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	–	
	<b>Post-build time</b>	–	
<b>Scope / Dependency</b>	scope: local		

]



**Figure 10.57: IKECertificates**

## 10.2.78 IKECertificates

### [ECUC\_IKE\_00005] Definition of EcucParamConfContainerDef IKECertificates [

Container Name	IKECertificates
Parent Container	<a href="#">IKE</a>
Description	Container for configuration of IKE certificates.
Configuration Parameters	

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">IKEMaxPublicKeyLength</a>	1	<a href="#">[ECUC_IKE_00042]</a>
<a href="#">IKEMaxSubjectNameSize</a>	1	<a href="#">[ECUC_IKE_00041]</a>
<a href="#">IKECertificateBufferRef</a>	2	<a href="#">[ECUC_IKE_00043]</a>

Included Containers		
Container Name	Multiplicity	Scope / Dependency
<a href="#">IKECertificate</a>	2..65534	Container for configuration of an identity certificate and its issuer certificate. Use this container to configure a valid chain of certificates. The top-level certificate must be a self-signed certificate.

## [ECUC\_IKE\_00042] Definition of EcucIntegerParamDef IKEMaxPublicKeyLength

Parameter Name	IKEMaxPublicKeyLength		
Parent Container	<a href="#">IKECertificates</a>		
Description	The maximum length of the public key in a certificate. Choose 64 for ECDSA-256, 96 for ECDSA-384, etc.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	64 .. 4096		
Default value	96		
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		

## [ECUC\_IKE\_00041] Definition of EcucIntegerParamDef IKEMaxSubjectNameSize

Parameter Name	IKEMaxSubjectNameSize		
Parent Container	<a href="#">IKECertificates</a>		
Description	The maximum size of the Subject Name field in certificates.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	32 .. 1024		
Default value	256		
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		

### [ECUC\_IKE\_00043] Definition of EcucReferenceDef IKECertificateBufferRef [

Parameter Name	IKECertificateBufferRef		
Parent Container	<a href="#">IKECertificates</a>		
Description	The referenced keys are used as buffers for temporarily storing the peer certificates.		
Multiplicity	2		
Type	Symbolic name reference to CsmKey		
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		

## 10.2.79 IKECertificate

### [ECUC\_IKE\_00044] Definition of EcucParamConfContainerDef IKECertificate [

Container Name	IKECertificate		
Parent Container	<a href="#">IKECertificates</a>		
Description	Container for configuration of an identity certificate and its issuer certificate. Use this container to configure a valid chain of certificates. The top-level certificate must be a self-signed certificate.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
<a href="#">IKEIssuerCertificateRef</a>	1	[ECUC_IKE_00045]
<a href="#">KEIdentityCertificateRef</a>	1	[ECUC_IKE_00046]



No Included Containers

## [ECUC\_IKE\_00045] Definition of EcucReferenceDef IKEIssuerCertificateRef [

Parameter Name	IKEIssuerCertificateRef		
Parent Container	<a href="#">IKECertificate</a>		
Description	The referenced certificate is the Issuer Certificate. The Issuer Certificate is used to identify the certificate authority (CA) which is the issuer of the Identity Certificate. The associated public key is used for verification of the certificate.		
Multiplicity	1		
Type	Reference to <a href="#">IKECertificate</a>		
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		

## [ECUC\_IKE\_00046] Definition of EcucReferenceDef KEIdentityCertificateRef [

Parameter Name	KEIdentityCertificateRef		
Parent Container	<a href="#">IKECertificate</a>		
Description	The referenced key is the Identity Certificate which is used to identify an entity and to associate that identity with a public key.		
Multiplicity	1		
Type	Symbolic name reference to CsmKey		
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		

### **10.3 Published Information**

For details refer to the chapter 10.3 “Published Information” in SWS\_BSWGeneral.

## A Change history of AUTOSAR traceable items

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

### A.1 Traceable item history of this document according to AUTOSAR Release R24-11

#### A.1.1 Added Specification Items in R24-11

[\[ECUC\\_Tcplp\\_00337\]](#) [\[ECUC\\_Tcplp\\_00338\]](#) [\[ECUC\\_Tcplp\\_00339\]](#) [\[ECUC\\_Tcplp\\_00340\]](#) [\[ECUC\\_Tcplp\\_00341\]](#) [\[ECUC\\_Tcplp\\_00342\]](#) [\[ECUC\\_Tcplp\\_00343\]](#) [\[ECUC\\_Tcplp\\_00344\]](#) [\[ECUC\\_Tcplp\\_00345\]](#) [\[ECUC\\_Tcplp\\_00346\]](#) [\[SWS\\_TCPIP\\_91017\]](#) [\[SWS\\_TCPIP\\_91018\]](#) [\[SWS\\_Tcplp\\_00382\]](#) [\[SWS\\_Tcplp\\_00383\]](#) [\[SWS\\_Tcplp\\_00384\]](#) [\[SWS\\_Tcplp\\_00385\]](#) [\[SWS\\_Tcplp\\_00386\]](#) [\[SWS\\_Tcplp\\_00387\]](#) [\[SWS\\_Tcplp\\_00388\]](#) [\[SWS\\_Tcplp\\_00389\]](#) [\[SWS\\_Tcplp\\_00390\]](#) [\[SWS\\_Tcplp\\_00391\]](#) [\[SWS\\_Tcplp\\_00392\]](#) [\[SWS\\_Tcplp\\_00393\]](#) [\[SWS\\_Tcplp\\_00394\]](#) [\[SWS\\_Tcplp\\_00395\]](#) [\[SWS\\_Tcplp\\_00396\]](#) [\[SWS\\_Tcplp\\_00397\]](#) [\[SWS\\_Tcplp\\_00398\]](#) [\[SWS\\_Tcplp\\_00399\]](#) [\[SWS\\_Tcplp\\_00400\]](#) [\[SWS\\_Tcplp\\_00401\]](#) [\[SWS\\_Tcplp\\_00402\]](#) [\[SWS\\_Tcplp\\_00403\]](#) [\[SWS\\_Tcplp\\_00404\]](#) [\[SWS\\_Tcplp\\_00405\]](#) [\[SWS\\_Tcplp\\_00406\]](#) [\[SWS\\_Tcplp\\_00407\]](#) [\[SWS\\_Tcplp\\_00408\]](#) [\[SWS\\_Tcplp\\_00409\]](#) [\[SWS\\_Tcplp\\_00410\]](#) [\[SWS\\_Tcplp\\_00411\]](#) [\[SWS\\_Tcplp\\_00412\]](#) [\[SWS\\_Tcplp\\_00413\]](#) [\[SWS\\_Tcplp\\_00414\]](#) [\[SWS\\_Tcplp\\_00415\]](#) [\[SWS\\_Tcplp\\_00416\]](#) [\[SWS\\_Tcplp\\_00417\]](#) [\[SWS\\_Tcplp\\_00418\]](#) [\[SWS\\_Tcplp\\_00419\]](#) [\[SWS\\_Tcplp\\_00420\]](#)

#### A.1.2 Changed Specification Items in R24-11

[\[ECUC\\_Tcplp\\_00014\]](#) [\[ECUC\\_Tcplp\\_00015\]](#) [\[ECUC\\_Tcplp\\_00017\]](#) [\[ECUC\\_Tcplp\\_00018\]](#) [\[ECUC\\_Tcplp\\_00021\]](#) [\[ECUC\\_Tcplp\\_00037\]](#) [\[ECUC\\_Tcplp\\_00052\]](#) [\[ECUC\\_Tcplp\\_00064\]](#) [\[ECUC\\_Tcplp\\_00068\]](#) [\[ECUC\\_Tcplp\\_00069\]](#) [\[ECUC\\_Tcplp\\_00174\]](#) [\[ECUC\\_Tcplp\\_00295\]](#) [\[ECUC\\_Tcplp\\_00297\]](#) [\[ECUC\\_Tcplp\\_00306\]](#) [\[ECUC\\_Tcplp\\_00307\]](#) [\[ECUC\\_Tcplp\\_00309\]](#) [\[ECUC\\_Tcplp\\_00310\]](#) [\[ECUC\\_Tcplp\\_00311\]](#) [\[ECUC\\_Tcplp\\_00313\]](#) [\[ECUC\\_Tcplp\\_00314\]](#) [\[ECUC\\_Tcplp\\_00316\]](#) [\[ECUC\\_Tcplp\\_00320\]](#) [\[SWS\\_TCPIP\\_00008\]](#) [\[SWS\\_TCPIP\\_00027\]](#) [\[SWS\\_TCPIP\\_00028\]](#) [\[SWS\\_TCPIP\\_00029\]](#) [\[SWS\\_TCPIP\\_00042\]](#) [\[SWS\\_TCPIP\\_00126\]](#) [\[SWS\\_TCPIP\\_00255\]](#) [\[SWS\\_Tcplp\\_00062\]](#) [\[SWS\\_Tcplp\\_00077\]](#) [\[SWS\\_Tcplp\\_00087\]](#) [\[SWS\\_Tcplp\\_00104\]](#) [\[SWS\\_Tcplp\\_00202\]](#) [\[SWS\\_Tcplp\\_00300\]](#) [\[SWS\\_Tcplp\\_00329\]](#) [\[SWS\\_Tcplp\\_00362\]](#)

#### A.1.3 Deleted Specification Items in R24-11

[\[SWS\\_Tcplp\\_00131\]](#) [\[SWS\\_Tcplp\\_00285\]](#)

#### **A.1.4 Added Constraints in R24-11**

[[SWS\\_Tcplp\\_CONSTR\\_00001](#)]

#### **A.1.5 Changed Constraints in R24-11**

none

#### **A.1.6 Deleted Constraints in R24-11**

none