Elektrobit

EB tresos® AutoCore Generic 8
IP Stack documentation

release notes update for the TcpIp module

product release 8.8.7

Elektrobit Automotive GmbH
Am Wolfsmantel 46
91058 Erlangen, Germany
Phone: +49 9131 7701 0
Fax: +49 9131 7701 6333
Email: info.automotive@elektrobit.com

Technical support

https://www.elektrobit.com/support

Legal disclaimer

Confidential information.

Table of Contents

1. Overview

This document provides you with the release notes to accompany an update to the `TcpIp` module. Refer to the changelog Section 2.1, “Change log” for details of changes made for this update.

Release notes details

- EB tresos AutoCore release version: 8.8.7
- EB tresos Studio release version: ..
- AUTOSAR R4.3 Rev 0
- Build number: B577598

2. TcpIp module release notes

- AUTOSAR R4.3 Rev 0
- AUTOSAR SWS document version: 4.3.0
- Module version: 3.5.18.B577598
- Supplier: Elektrobit Automotive GmbH

2.1. Change log

This chapter lists the changes between different versions.

Module version 3.5.18

2022-10-28

- Added support for external network IP address to TcpIp_IsConnectionReady
- Added support for Tls

Module version 3.5.17

2022-09-16

- ASCTCPIP-3050 Fixed known issue: Possible reuse of GCM initialization vectors when 2^32 INIT requests are forced by an adversary within IKE SA lifetime
- ASCTCPIP-3115 Fixed known issue: TcpIp continues processing the message discarded by the policy check
- Added support for detailed policy violations logging

Module version 3.5.16

2022-06-10

- ASCTCPIP-3035 Fixed known issue: TcpIp does not correctly access the security database

- ASCTCPIP-3044 Fixed known issue: Using trusted end-point certificate with sect571r1 public key allows bypassing authentication check
- ASCTCPIP-3047 Fixed known issue: IKE negotiation is not possible if the remote end-point certificate uses ECDSA with P-521 elliptic curve
- ASCTCPIP-3045 Fixed known issue: Possible out-of-bounds read when parsing the ID payload
- ASCTCPIP-3049 Fixed known issue: Possible read out-of-bounds on 16-bit architectures when parsing remote ID payload
- ASCTCPIP-3048 Fixed known issue: Possible read out-of-bounds when parsing the subject identifier in remote identification payload
- ASCTCPIP-3076 Fixed known issue: SA used before IKE AUTH Response is sent out
- Added support for introduction of new TcpIpCtrl instances at PostBuild time

Module version 3.5.15

2022-02-18

- ASCTCPIP-2934 Fixed known issue: TCP data transmission without window check
- Added the support for IKE
- ASCTCPIP-2935 Fixed known issue: Undefined behavior on reception of IKEv2 INIT message with ID payload
- Added IPv4 address format checks when calling TcpIp_RequestIpAddrAssignment to replace an existing static address with a new one
- ASCTCPIP-2944 Fixed known issue: TcpIp_IsConnectionReady() obtains and uses an incorrect controller index
- ASCTCPIP-2966 Fixed known issue: Certificate trust anchor of a received chain is not properly recognized in IKEv2
- ASCTCPIP-2992 Fixed known issue: Tcp handles the FIN transmission, data transmission, and FIN acknowledgement during controller SHUTDOWN procedure incorrectly
- Improve measurement data counter handling
- Added the support for reporting security events to IdsM

Module version 3.5.14

2021-10-08

- Added support to handle different return values from TcpIp_CopyTxDataAPI

- Added Measurement data support update
- Added IPv6 address format checks when calling TcpIp_RequestIpAddrAssignment to replace an existing static address with a new one

Module version 3.5.13

2021-06-25

- Added the support of Duplicate Address Detection for DHCPv4
- Added support to enable UDP sockets to always listen to limited broadcasts
- Added support for IpV4 Router extension
- ASCTCPIP-2777 Fixed known issue: AutoIp NvM - First generation yields the same address

Module version 3.5.12

2021-03-05

- Added support to store IP address in NVM RAM
- Added the support of Duplicate Address Detection for DHCPv6
- ASCTCPIP-2632 Fixed known issue: TcpIp uses the wrong data type for the Eth buffer index

Module version 3.5.11

2020-10-23

- ASCTCPIP-2436 Fixed known issue: Out-of-bounds read access by packet reception in dual-stack and dual-protocol configuration
- ASCTCPIP-2438 Fixed known issue: Out-of-bounds read access by reception of corrupted DHCPv4 message
- ASCTCPIP-2452 Fixed known issue: TcpIp inverts byte order on big endian platforms and reports false generator error
- ASCTCPIP-2352 Fixed known issue: TcpIp does not release an allocated TCP socket
- ASCTCPIP-2489 Fixed known issue: TcpIp does not create a new initial sequence number for each passive connection
- ASCTCPIP-2500 Fixed known issue: TcpIp_TcpConnect returns E_NOT_OK instead of E_OK if SYN transmission fails

Module version 3.5.10

2020-08-07

- Added support to make TCP upper layer copying algorithm configurable
- Added the support of unpredictable sequence numbers according to IETF RFC 6528
- Added the support for ignoring TCP RST frames when in state TIME_WAIT according to IETF RFC 1337
- ASCTCPIP-2375 Fixed known issue: TcpIp sends packages which are addressed to a link local address to the router
- Added the TCP timeout for closing sockets in state SYN_RECEIVED
- ASCTCPIP-2399 Fixed known issue: TcpIp does not release DHCPv4 address if address conflict is detected

Module version 3.5.9

2020-06-19

- ASCTCPIP-2081 Fixed known issue: TcpIp does not send configurable options in DHCP Request
- Added support for filtering received packets by source MAC address
- Added counters for IP frames dropped or passed due to firewall rule
- Added support for filtering received packets by traffic class and flow label
- Added security Architecture for the Internet Protocol according to IETF RFC 4301
- Added IP Authentication Header according to IETF RFC 4302
- Added the AES-CMAC-96 Algorithm for Authentication Header according to IETF RFC 4494
- Added the Galois Message Authentication Code (GMAC) Algorithm (AUTH_AES_128_GMAC, AUTH_AES_256_GMAC) for Authentication Header according to IETF RFC 4543
- Added the HMAC Algorithm (HMAC-SHA-256-128) for Authentication Header according to IETF RFC 4868

  Note: If HMAC is used to secure IPv4 traffic with a Linux host, the respective transform state i.e. security association needs to be configured with the `align4` flag. For more information please refer to `man ip-xfrm`.
- Added new API, TcpIp_IsConnectionReady, to check if physical address is known and IpSec SA is established
- Added the support of SYN cookies according to IETF RFC 4987
- ASCTCPIP-2249 Fixed known issue: ARP creates entries for multicast remote Ip address
- Changed value range of parameter TcpIpArpTableEntryTimeout (1 second to 65535 seconds or Infinity)
- Fixed processing of TCP SYN segments which contain a RST

Module version 3.5.8

2020-02-21

- Improved Window Update transmission in ACK
- Improved precision of ARP timeout counter
- Rework DHCPv4 to use UdpTransmit API
- Rework DHCPv6 to use UdpTransmit API
- Updated `TcpIp_RxIndication` to always drop the packet if the total length is greater than MTU, regardless if DET reporting is enabled or not.
- Improved message transmission (TCP, UDP and ICMP use the same transmit API)
- Fixed unreachable code assert failing on reception of TCP FIN segment with retransmitted data.
- Fixed unreachable code assert failing on TCP reception when listen socket closed or upper layer not accepting.
- ASCTCPIP-1979 Fixed known issue: Possible buffer overflow on TCP segment reception if out-of-order buffering is activated
- Improved checking Server Identifier option in every DhcpV6 message
- Improved handling of the valid and preferred lifetime for DhcpV6 messages
- Improved handling of restarting the 2 MSL timeout when retransmitted FIN is received in the state Time-Wait
- ASCTCPIP-2012 Fixed known issue: Compilation fails when SoAd is not configured
- Improved handling of discarding DHCP messages in state init

Module version 3.5.7

2019-10-11

- ASCTCPIP-1823 Fixed known issue: Invalid controller transfer to OFFLINE state due to address with multiple assignment methods

Module version 3.5.6

2019-07-05

- ASCTCPIP-1792 Fixed known issue: Out-of-bounds read access caused by an invalid DHCPv4 router option
- ASCTCPIP-1794 Fixed known issue: Denial of service by reception of a Neighbor Solicitation/Advertisement with invalid option length

- ASCTCPIP-1795 Fixed known issue: Integer underflow causes wrong length information for the upper layer

Module version 3.5.5

2019-06-14

- ASCTCPIP-1646 Fixed known issue: TcpIp sends ARP reply with incorrect target hardware address
- ASCTCPIP-1694 Fixed known issue: NDP cache entry is not unlocked if neighbor solicitation transmission fails
- ASCTCPIP-1654 Fixed known issue: TcpIp does not drop Neighbor Advertisements/Solicitation with broadcast/multicast target/source link layer address
- Added support for IPv6 Global address Duplicate Address Detection
- ASCTCPIP-1735 Fixed known issue: Denial of service by reception of a corrupted DHCPv6 response on 16-bit platforms
- Added support for extracting and transmitting arbitrary configured DHCP options
- Added support for IpV6 Router extension
- Added basic support for out-of-order reception and buffering of TCP segments.

  Note: With introduction of this feature, if out-of-order reception is disabled a duplicate ACK is not sent if an out-of-order segment is received. A duplicate ACK would indicate that out-of-order buffering is supported.
- ASCTCPIP-1759 Fixed known issue: Partial IPv4 and IPv6 checksum calculation might fail

Module version 3.5.4

2019-02-15

- Added support for Measurement data
- Added support for Ipv6 Next hop determination
- ASCTCPIP-1619 Fixed known issue: TcpIp not be able to assign a DhcpV6 address
- ASCTCPIP-1606 Fixed known issue: The DhcpV4 client transmits a message over a closed DhcpV4 Udp socket
- ASCTCPIP-1617 Fixed known issue: TcpIp does not compile if the configurable DHCP option feature is enabled and Det is disabled
- Added Duplicate Address Detection conflict callout support for IpV4
- Added support for Measurement data for discarded/replaced ARP entries
- Added Api for accessing DHCP status

- ASCTCPIP-1630 Fixed known issue: Out-of-bounds read access caused by malformed NDP packet reception
- ASCTCPIP-1638 Fixed known issue: TcpIp transmits a malformed IPv4 DHCP FQDN option
- ASCTCPIP-1631 Fixed known issue: Out-of-bounds read access and potential out-of-bounds write access caused by IPv4 ICMP echo request packet reception
- ASCTCPIP-1629 Fixed known issue: Out-of-bounds read access caused by corrupted IPv6 packet reception
- Added support for IPv6 Static address and IPv6 Link Local address Duplicate Address Detection
- Added support for IP Stack Hardening: ACK Loop DoS Attack

Module version 3.5.3

2018-11-23

- ASCTCPIP-1574 Fixed known issue: Potential corruption of internal data structure during reassembly of malicious fragments

Module version 3.5.2

2018-10-26

- ASCTCPIP-1528 Fixed known issue: Malformed ICMPv6 Echo Reply is sent as a response to a 4-byte ICMPv6 Echo Request
- ASCTCPIP-1552 Fixed known issue: Incorrect inclusion of ComStack_Types.h

Module version 3.5.1

2018-09-20

- ASCTCPIP-1520 Fixed known issue: TCP option filter does not compile if TCP keep alive is turned off
- Added support for Post Build Selectable

Module version 3.5.0

2018-09-06

- ASCTCPIP-1495 Fixed known issue: IPv4 Unicast address of controller is incorrectly generated

- ASCTCPIP-1466 Fixed known issue: TcpIp calls the external function <Up>_CopyTxData() in a critical section
- Added support for more flexible memory allocation per socket

  Note: With this feature the configuration of the memory for TCP sockets has changed. The conversion to the new memory configuration can be simply applied by adding one entry in TcpIpConfig/TcpIpMemoryConfig/TcpIpMemoryPool. TcpIpMemoryBlockSize needs to be set to TcpIpBufferMemory divided by TcpIpNumMemoryBlocks and TcpIpMemoryBlockCount to TcpIpNumMemoryBlocks.

Module version 3.4.0

2018-06-22

- Added support for Configurable DSCP and Flow Label
- Updated requirements and configuration to AUTOSAR SWS 4.3.0
- ASCTCPIP-1376 Fixed known issue: Memory section conflicts between definitions and declarations
- Added support for handling arbitrary configured DHCP options
- Added support for configurable UDP checksum calculation
- ASCTCPIP-1452 Fixed known issue: TCP stops transmitting data after retransmission
- Added TcpIp_MainFunctionTx() to allow immediate transmission of TCP segments
- ASCTCPIP-1443 Fixed known issue: Enabling DET in the dual stack version leads to incorrect function calls via the function pointers
- ASCTCPIP-1454 Fixed known issue: TcpIp might access data from wrong TcpIp controller
- ASCTCPIP-1457 Fixed known issue: Incorrect reassembly of fragmented IP message in case header size of IP fragments is not constant

Module version 3.3.0

2018-02-16

- Added Support of router and prefix discovery
- ASCTCPIP-1327 Fixed known issue: TcpIp might get stuck in an endless loop if more than 255 UDP/TCP sockets are configured
- Improved IP header writing to use 16 + 32 bit writes (configurable by integrator)
- Improved Checksum computation to use aligned 64 bit reads
- Added Support for TCP option filter
- Added support for IPv6 Extension Header Filter

- Added support for Defensive Neighbor Solicitation/Advertisement Processing

Module version 3.2.11

2017-12-15

- Added support for TCPIP_IPADDR_ASSIGNMENT_ALL for TcpIp_RequestIpAddrAssignment
- ASCTCPIP-1290 Fixed known issue: TcpIp_GetIpAddr() does not return the correct DHCP address
- Added Support of local address "ANY" for TcpIp_Request/ReleaseIpAddrAssignment

Module version 3.2.10

2017-10-19

- Added mechanism to prevent ARP floods (configurable)

Module version 3.2.9

2017-09-22

- ASCTCPIP-1158 Fixed known issue: TcpIp_GetIpAddr does not return correct IP address
- Updated to MISRA 2012
- Added mechanism to prevent ARP floods (non-configurable)
- Added support for checksum offloading according to AUTOSAR 4.2.1

Module version 3.2.8

2017-08-25

- ASCTCPIP-1136 Fixed known issue: TcpIp unexpectedly removes ARP entries

Module version 3.2.7

2017-07-28

- ASCTCPIP-1103 Fixed known issue: TcpIp receives frames on wrong controller

- ASCTCPIP-1120 Fixed known issue: Incorrect checksum calculation of fragmented IPv6 UDP message
- ASCTCPIP-1127 Fixed known issue: Incorrect length information for Out-Of-Order or disabled fragmentation

Module version 3.2.6

2017-06-30

- ASCTCPIP-1077 Fixed known issue: TcpIp_TcpTransmit unexpectedly reports TCPIP_E_NOBUFS
- ASCTCPIP-1086 Fixed known issue: Incorrect checksum calculation of fragmented UDP message

Module version 3.2.5

2017-06-02

- Added support to read and write the NDP cache table

Module version 3.2.4

2017-05-05

- ASCTCPIP-1025 Fixed known issue: ICMP Echo Replies might exceed MTU
- Added support to read and write the ARP cache table
- ASCTCPIP-1029 Fixed known issue: Unreachable code assertion in LocalAddrSM causing execution to stop

Module version 3.2.3

2017-03-31

- Added support for Simple DHCPv4
- ASCTCPIP-932 Fixed known issue: Incorrect DHCPv6 timeout calculation if more than one DHCPv6 assignment is configured
- Added support of IPv6 source address selection algorithm according to IETF RFC 6724
- ASCTCPIP-989 Fixed known issue: Incorrect IPv4 Link Local timeout calculation
- ASCTCPIP-985 Fixed known issue: Link Local IPv6 address cannot be released
- ASCTCPIP-1001 Fixed known issue: Incorrect DHCPv6 retransmission timeout calculation

Module version 3.2.2

2017-03-03

- ASCTCPIP-931 Fixed known issue: DHCPv6 assignment fails if more than one controller is configured
- Added support for indicating change in the physical address table (Up_PhysAddrTableChg) for IPv6
- Improved initialization of module (name of configuration can be used as symbol for TcpIp_Init())

Module version 3.2.1

2017-02-03

- Support of Fully Qualified Domain Name (FQDN) Option for Dynamic Host Configuration Protocol for IPv6 Clients
- Added support for IPv4 Address Conflict Detection and Defense according to IETF RFC 5227
- ASCTCPIP-873 Fixed known issue: TcpIp might send datagrams to wrong destination MAC address
- ASCTCPIP-890 Fixed known issue: The FQDN option is incorrectly terminated
- ASCTCPIP-834 Fixed known issue: Dhcp does not release the IPv6 address after the lease time expires
- ASCTCPIP-905 Fixed known issue: The variable TcpIp_TCP_fragmentIDCounter is created without using the MemMap concept

Module version 3.2.0

2016-11-04

- ASCTCPIP-564 Fixed known issue: TCPIP_E_NOTCONN is unexpectedly reported to Det because the socket is already closed
- ASCTCPIP-590 Fixed known issue: TcpIp might use a wrong netmask
- ASCTCPIP-664 Fixed known issue: ARP Callout is not called for received multicast datagrams
- ASCTCPIP-679 Fixed known issue: TCP does not transmit RST in LISTEN state
- ASCTCPIP-677 Fixed known issue: IP address is released if DHCP server does not acknowledge the request
- Updated Up_IcmpMsgHandler to AUTOSAR SWS 4.3
- Updated TcpIp Configuration to AUTOSAR SWS 4.2.2
- Fixed Compiler abstractions and memory sections
- ASCTCPIP-398 Fixed known issue: Source address in DHCP messages is not set to the unspecified address

- Added support of Internet Protocol version 6 (IPv6) including Hop-By-Hop Option Header and Destination Option Header
- Added support for fragmentation of over sized IPv6 and IPv4 frames
- Added support for reception and reassembly of fragmented IPv6 and IPv4 frames
- Added support of Internet Control Message Protocol Version 6 (ICMPv6) including Destination Unreachable, Time Exceeded, Parameter Problem, Echo Request/Reply Messages
- Added support of Address Resolution and Neighbor Unreachability Detection (NDP)
- Added client support of Dynamic Host Configuration Protocol for IPv6
- Added support for Stateless Address Autoconfiguration of IPv6 Link-Local Addresses
- Updated TcpIp_IpAddrAssignmentType and TcpIp_ReturnType to AUTOSAR SWS 4.2.2
- Updated TcpIp to use Eth_BufIdxType

Module version 3.1.6

2016-05-25

- ASCTCPIP-565 Fixed known issue: TcpIp calls the wrong upper layer functions

Module version 3.1.5

2016-04-29

- Added support for Debug & Trace with custom header file configurable via parameter `BaseDbgHeaderFile`
- ASCTCPIP-498 Fixed known issue: DHCP assignment fails if more than one DHCP assignment is configured
- ASCTCPIP-499 Fixed known issue: TCPIP_E_INV_ARG is unexpectedly reported to Det because the socket is already closed
- Added support for a user configurable packet filter, to enable ARP table updates from IpV4 datagrams based on payload content.
- ASCTCPIP-514 Fixed known issue: Retransmission timeout of a DHCP REQUEST might be less than 60 seconds
- ASCTCPIP-539 Fixed known issue: Link Local IP address is never assigned if TcpIpAutoIpInitTimeout is smaller than one
- ASCTCPIP-540 Fixed known issue: Address assignment change callouts for address ANY are not called
- ASCTCPIP-558 Fixed known issue: AutoIp/DoIp assignment fails if more than one AutoIp/DoIp assignment is configured

Module version 3.1.4

2016-02-05

- ASCTCPIP-470 Fixed known issue: DHCP IP address assignment doesn't work if TcpIpAssignmentTrigger = TCPIP_AUTO
- ASCTCPIP-475 Fixed known issue: TCP does not free allocated memory if socket is unexpectedly closed
- ASCTCPIP-476 Fixed known issue: Reset is not transmitted if socket in state CLOSE_WAIT is closed with TcpIp_Close and force = TRUE

Module version 3.1.3

2016-01-14

- ASCTCPIP-439 Fixed known issue: TcpIp_TcpTransmit might transmit wrong data
- ASCTCPIP-433 Fixed known issue: TCP might pass invalid data to upper layer
- ASCTCPIP-435 Fixed known issue: TCP might use an invalid MSS for segmentation
- ASCTCPIP-434 Fixed known issue: TCP might call <UL>_TxConfirmation with incorrect Length
- ASCTCPIP-432 Fixed known issue: <UL>_TxConfirmation might be called after <UL>_TcpEvent(TCPIP_-TCP_CLOSED)
- ASCTCPIP-430 Fixed known issue: If an out of order FIN is received TCP calls <UL>TcpIpEvent(TCPIP_-TCP_FIN_RECEIVED)
- ASCTCPIP-441 Fixed known issue: TCP might not accept a valid segment from remote host
- ASCTCPIP-431 Fixed known issue: If a valid FIN/ACK with data is received in state SYN-RECEIVED data is not passed to upper layer
- ASCTCPIP-436 Fixed known issue: If an upper layer calls TcpIp_Close a FIN might not be transmitted
- ASCTCPIP-440 Fixed known issue: TCP might close the connection although not all data has been transmitted
- ASCTCPIP-421 Fixed known issue: Initialization of TcpIp causes to call EthIf_GetPhysAddr() before EthIf_-ControllerInit()
- Added config check that EthIfctrlIdx does not exceed TcpIpEthIfCtrlIndexMax
- ASCTCPIP-444 Fixed known issue: TcpIp might send more data segments than a single full sized segment if nagle is used
- ASCTCPIP-449 Fixed known issue: TcpIp does not immediately shutdown if state OFFLINE is requested in state ONHOLD
- Add checks to TcpIp_Bind(), TcpIp_Listen() and TcpIp_Connect() to operate only if local address to use is assigned

- ASCTCPIP-457 Fixed known issue: Pending data might not be transmitted during transition from ONHOLD to ONLINE

Module version 3.1.2

2015-11-06

- ASCTCPIP-407 Fixed known issue: TcpIp might access an invalid memory address
- ASCTCPIP-409 Fixed known issue: TcpIp accepts SYN,ACK with incorrect ACK number in state SYN-SENT
- ASCTCPIP-415 Fixed known issue: TcpIp might call <UL>_TcpIpEvent() with an invalid SocketId for Tcp listen sockets
- ASCTCPIP-418 Fixed known issue: TcpIp uses wrong netmask on big endian CPU

Module version 3.1.1

2015-10-09

- ASCTCPIP-366 Fixed known issue: Checksum calculation might fail
- Change configuration parameter TcpIpDefaultRouter to optional
- ASCTCPIP-383 Fixed known issue: TcpIp incorrectly acknowledges a FIN,ACK of the remote host
- TcpIp does not send unexpected acknowledgments in the next Mainfunction after TcpIp_Close is called anymore
- ASCTCPIP-386 Fixed known issue: TcpIp controller statemachine might not switch to OFFLINE if DHCP is used
- Support of keep alive probes according to AUTOSAR 4.2.2
- ASCTCPIP-399 Fixed known issue: TCP retransmits a correctly acknowledged data segment

Module version 3.1.0

2015-06-19

- Support of Transmission Control Protocol (TCP) including Nagle Algorithm
- Support of Dynamic Host Configuration Protocol (DHCPv4)
- Support of Fully Qualified Domain Name (FQDN) Option for Dynamic Host Configuration Protocol for IPv4 Clients
- Support of Dynamic Configuration of IPv4 Link-Local Addresses (Auto-IP)

- Support of ISO 13400-2 recommended timing values for Dynamic Configuration of IPv4 Link-Local Addresses
- Support of Gratuitous ARP.
- Support of configurable upper layer

Module version 3.0.1

2014-12-12

- ASCTCPIP-313 Fixed known issue: Reception and transmission of certain multicast IPv4 datagrams fails

Module version 3.0.0

2014-11-30

- Initial mass production version (limited feature set).

2.2. New features

- Added support for external network IP address to TcpIp_IsConnectionReady
- Added support for Tls

2.3. Elektrobit-specific enhancements

This chapter lists the enhancements provided by the module.

- Ipv4 receive indication callout for ARP insertion

  Description:

  With configuration parameters contained in `TcpIpIpV4ArpPacketFilter()` the user is able to specify a callout which is invoked on successful reception of an IpV4 datagram. The callout decides based on the received ipV4 datagram's content, whether the sender's source address shall be inserted into to the ARP table or not.
- IP fragmentation and reassembly support can be enabled and disabled separately.
- IP fragmentation supports two modes: in-order and out-of-order transmission.
- Handling of atomic packets according to IETF RFC 6946.

  Description:

IETF RFC 6946 states that "atomic" fragments, i.e. packets which have a fragment header, but an 'offset' of zero and the 'more' flag also equal to zero - i.e. fragments that do contain the whole packet, shall NOT collide with partially-assembled packets using the same ID, while the partially-assembled fragments shall NOT be discarded due to this non-collision.

- Simple DHCPv4 Client

  With the configuration parameter `TcpIpUseSimpleDhcpClient` the simple DHCPv4 client can be enabled. The IP address is assigned through an exchange of 2 messages with the DHCP server. Client sends a DHCPDISCOVER with XID set to the lower 4 bytes of the MAC address. If server responses with a DHCPOFFER with the XID set to the client's MAC address, the client sets its own IP address to that given in the YIADDR field.

- Set static ARP/NDP cache entries through API TcpIp_SetRemotePhysAddr

  TcpIp_SetRemotePhysAddr allows to set an entry in the NDP or ARP cache to static, remove a static entry if no longer needed or clear the whole cache.

- Request the assignment of multiple assignment methods through TcpIp_RequestIpAddrAssignment

  If TcpIp_RequestIpAddrAssignment is called with LocalAddrId configured as ANY, all assignment methods of all local addresses configured for the referenced controller are assigned.

  If TcpIp_RequestIpAddrAssignment is called with Type equals TCPIP_IPADDR_ASSIGNMENT_ALL, all assignment methods of the specified LocalAddrId are assigned. See http://www.autosar.org/bugzilla/show_bug.cgi?id=74847

- Mechanism to prevent ARP floods

  After the transmission of an ARP request the TcpIp skips 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. See http://www.autosar.org/bugzilla/show_bug.cgi?id=80210

- Trigger transmissions through TcpIp_MainFunctionTx()

  TcpIp_MainFunctionTx() allows it to trigger an immediate transmission of TCP segments after the call of TcpIp_TcpTransmit(). The API can be enabled through the configuration parameter TcpIpEnableMainFunctionTx.

- Internet Protocol Security (IPsec)

  - Security Architecture for the Internet Protocol according to IETF RFC 4301

    The TcpIp implements the Security Architecture for the Internet Protocol defined in IETF RFC 4301. This includes the configuration of a Security Policy Database (SPD) in which traffic can be configured as BYPASSED, SECURED or DISCARD. For secured traffic the TcpIp allows to configure manual and dynamic security associations in a Security Association Database (SAD). Dynamic security associa-

tions are negoatiated between the TcpIp and the remote host during the start up of the TcpIp through the Internet Key Exchange Protovol Version 2. Features supported by the IKEv2 implementation are listed below. Every time a UDP/TCP/ICMP frame is received or transmitted, the TcpIp consults the SPD and decides if the frame shall be bypassed, secured or discarded.

TcpIp provides the API TcpIp_RequestIpSecMode() to dis/enable security between the TcpIp and the selected remote host. If enabled all SECURED policies configured for the remote host are actived and secured traffic is exchanged between the TcpIp and the remote host. If disabled the SECURED policies are treated as BYPASSED.

- IP Authentication Header according to IETF RFC 4302

  The TcpIp allows to secure traffic by the Authentication Header in transport mode defined in IETF RFC 4302.

- The AES-CMAC-96 Algorithm according to IETF RFC 4494

  The TcpIp supports the AES-CMAC-96 integrity algorithm for the Authentication Header.

- The Galois Message Authentication Code (GMAC) Algorithm according to IETF RFC 4543

  The TcpIp supports the following GMAC integrity algorithm for the Authentication Header:

  - AUTH_AES_128_GMAC
  - AUTH_AES_256_GMAC

- The HMAC Algorithm according to IETF RFC 4868

  The TcpIp supports the following HMAC integrity algorithm for the Authentication Header:

  - HMAC-SHA-256-128

  Note: If HMAC is used to secure IPv4 traffic with a Linux host, the respective transform state i.e. security association needs to be configured with the `align4` flag. For more information please refer to `man ip-xfrm`.

- The Anti-Replay Algorithm according to IETF RFC 6479

  The TcpIp implements Anti-Replay Algorithm for the Authentication Header without the need for bit shifting and it reduces the number of times an anti-replay window is adjusted.

- Extended Sequence Numbers for AH according to IETF RFC 4302

  The TcpIp uses Extended Sequence Numbers for detecting replay attacks and for dynamic security associations negotiated through IKEv2 per default. Note: 32 bit sequence numbers are not supported.

- Recommended algorithms for Authentication Header (AH) according to RFC 8221

  The TcpIp supports the following AH Authentication Algorithms :

  - AES_128_GMAC

- AES_256_GMAC
- HMAC_SHA2_256_128

- IPsec features deviations to the RFC are listed in Section 2.4, “Deviations ”.

- Internet Key Exchange Protocol Version 2 (IKEv2)
  - Internet Key Exchange Protocol Version 2 (IKEv2) according to IETF RFC 7296
    - TcpIp supports the establishment of IKE SAs between the TcpIp and a configurable amount of remote hosts through IKE_SA_INIT/IKE_AUTH exchanges.
    - TcpIp supports the establishment of a single IPsec Security Association per IKE security association through the IKE_AUTH exchange.
    - TcpIp supports the re-authentication of the IKE SA (creating a new IKE SA from scratch by using IKE_SA_INIT/IKE_AUTH exchanges) by the initiator or responder after a configurable amount of time. The new IKE SA is created in parallel to the existing IKE SA and then the old IKE SA is deleted (Make-Before-Break principle)
    - TcpIp supports configurable life time for the IKE security association and deletion after life time expires.
    - TcpIp provides the API TcpIp_RequestIpSecMode() to dis/enable IKE between the TcpIp and the selected remote host.
  - Window size according to IETF RFC 7296

    The TcpIp supports an IKE window size of one.
  - ID types to identify a remote host according to IETF RFC 7296

    The TcpIp supports the following ID types in the configuration of the connection table and the Identification Payload:
    - ID_IPV4_ADDR
    - ID_IPV6_ADDR
    - ID_DER_ASN1_DN
  - Certificate Encoding of supported according to IETF RFC 7296

    The TcpIp supports the following Certificate Encoding of Certificates in the CERT payload:
    - X.509 Certificate - Signature
  - Elliptic Curve Cryptography (ECC) groups according to IETF RFC 5903

    The TcpIp supports the following Diffie-Hellman Group Transforms for use in the Internet Key Exchange version 2 (IKEv2) protocols:
    - 256-Bit Random ECP Group

- 384-Bit Random ECP Group

- Signature Authentication in the Internet Key Exchange Version 2 according to RFC 7427

  The TcpIp supports the following signature algorithms to generate the signature :

  - ECDSA-with-SHA256
  - ECDSA-with-SHA384
  - ECDSA-with-SHA512

- Authentication methods for the authentication payload according to RFC 7427 and RFC 7296

  The TcpIp supports the following authentication methods:

  - Shared Key Message Integrity Code according to RFC 7296
  - Digital Signature according to RFC 7427

- AES-CBC Cipher Algorithm according to RFC 7296 and RFC 3602

  The TcpIp supports the following AES-CBC algorithm and key sizes for the Encryption Algorithm Transform in IKEv2:

  - ENCR_AES_CBC with key size 128bit
  - ENCR_AES_CBC with key size 256bit

- The HMAC Algorithm as a Pseudorandom Function in IKEv2 according to IETF RFC 4868

  The TcpIp supports the following HMAC algorithm for the Pseudorandom Function Transform in IKEv2:

  - PRF_HMAC_SHA2_256
  - PRF_HMAC_SHA2_384

- The HMAC Algorithm for authentication and integrity verification in IKEv2 according to IETF RFC 4868

  The TcpIp supports the following HMAC algorithm for the Integrity Algorithm Transform in IKEv2:

  - AUTH_HMAC_SHA2_256_128
  - AUTH_HMAC_SHA2_384_192

- The Traffic Selector Type in IKEv2 according to IETF RFC 7296

  The TcpIp supports the following Traffic Selector Types in IKEv2:

  - TS_IPV4_ADDR_RANGE
  - TS_IPV6_ADDR_RANGE

- Message Fragmentation for IKEv2 according to IETF RFC 7383

  The TcpIp supports to enable the IKEv2 message fragmentation and reassembly with the Encrypted and Authenticated Fragment Payload if the message exceeds the MTU

- GCM Algorithm according to RFC 4106

  The TcpIp supports the following GCM Encryption Transforms for use in the Internet Key Exchange version 2 (IKEv2) protocols:

  - AES-GCM with 16-octet ICV and 256 bit key
  - AES-GCM with 12-octet ICV and 256 bit key
- Recommended algorithms in the Internet Key Exchange Version 2 according to RFC 8247

  The TcpIp supports the following encryption algorithm :

  - ENCR_AES_CBC with 128 bit key
  - ENCR_AES_CBC with 256 bit key
  - ENCR_AES_GCM_16 with 256 bit key

  The TcpIp supports the following pseudorandom functions :

  - PRF_HMAC_SHA2_256

  The TcpIp supports the following integrity algorithm :

  - AUTH_HMAC_SHA2_256_128

  The TcpIp supports the following Diffie Hellman groups :

  - 256-bit random ECP

  The TcpIp supports the following authentication methods :

  - Shared Key Message Integrity Code
  - Digital Signature

  The TcpIp supports the following hash functions for IKEv2 digital signature :

  - SHA2-256
  - SHA2-384
  - SHA2-512

  The TcpIp supports the following authentication for IKEv2 digital signature :

  - ecdsa-with-sha256
- IKEv2 features deviations to the RFC are listed in Section 2.4, “Deviations ”.

- Transmission Control Protocol (TCP)

  - Transmission of the zero-window probes according to IETF RFC 1122

    The TCP shall send the first zero-window probe immediately when TcpIp_TcpTransmit is called.

2.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

- [IPV4] Protocol ARP not optional

  Description:

  Deactivation of core protocol ARP is not possible. The configuration parameter `TcpIpArpEnabled` is unused.

  Rationale:

  ARP is a core protocol and mandatory for IPv4. An alternative implementation using preconfigured and static address tables is not available.

  Requirements:

  ECUC_TcpIp_00006

- Certain TCP features are not supported

  Description:

  TcpIp does not support the following TCP features:

  - Slow Start

  - Congestion Avoidance

  - Out of order reception

  - Fast Retransmit/Recovery

  Requirements:

  SWS_TCPIP_00062, SWS_TCPIP_00064, ECUC_TcpIp_00061, ECUC_TcpIp_00063, ECUC_TcpIp_-00062, ECUC_TcpIp_00060, ECUC_TcpIp_00019, SWS_TCPIP_00168

- Some DET errors not supported

  Description:

  Module TcpIp does not support the following development errors:

  - `TCPIP_E_INIT_FAILED`

  Requirements:

  SWS_TCPIP_00042

- [IPV4] Path MTU discovery not supported

  Description:

  Discover the maximum transmission unit (MTU) for a path as defined in IETF RFC 1191 (Path MTU Discovery) is not supported. Configuration parameter `TcpIpPathMtuDiscoveryEnabled` is unused.

  Requirements:

  SWS_TCPIP_00055

- [IPV6] Path MTU discovery not supported

  Description:

  Discover the maximum transmission unit (MTU) for a path as required in IETF RFC 2460 (Path MTU Discovery) is not supported. Configuration parameter `TcpIpIpV6PathMtuDiscoveryEnabled`, `TcpIpIpV6PathMtuEnabled` and `TcpIpIpV6PathMtuTimeout` are unused.

  Requirements:

  SWS_TCPIP_00160, SWS_TCPIP_00158, ECUC_TcpIp_00090, ECUC_TcpIp_00107, ECUC_TcpIp_00105

- [IPV4] IPv4 fragmentation/reassembly mirrors IpV6 fragmentation/reassembly behavior

  Description:

  TcpIp supports IpV4 fragmentation and reassembly in the IPv6 sense only:

  - no overlapping fragments (see rfc5722).
  - honoring of DF=1 to save on IDs and remove the bandwidth limitation incurred from incorrect usage (see rfc6864).
  - non-colliding atomic fragments (see rfc6946, compared to rfc6864 for IpV4).
  - fixed reassembly timeout (no updating from fragments' TTL field) see rfc1122
  - end-to-end fragmentation without re-fragmentation in intermediate routers.

    While this is the default for IpV6, we also employ IPv4 in this mode: the DF-flag is on by default, but can be set to off for packets routed out of the in-car-network. This default avoids the bandwidth limitations that stem from large reassembly timeouts and the 16 bit ID field of IpV4. (see rfc6864 for a discussion)
  - no IpV6 path MTU discovery due to the fully-known and configurable environment.

  Rationale:

  IpV4 fragmentation according to rfc791, amended by rfc1122 to clarify timeout issues, then amended further by rfc6864 to overcome bandwidth limitations from the 16 bit fragmentID counter already comes close to the specification of IPv6's fragmentation as specified by rfc2460, amended by rfc5722 and rfc6946.

Security concerns described in rfc6274 and further addressed in rfc1858 and rfc3128 suggest disallowing overlapping fragments altogether in a controlled automotive network - as specified in rfc5722 for IpV6. IPv4 did not go there, because of legacy hardware and complex routing in combination with re-fragmentation in the wild of the internet, but requires equality checks for the overlapping parts of overlapping fragments according to rfc6864.

Since in-car networks do not have the alternative routing possibilities (too much randomness), the use case for overlapping fragments is non-existent. Since the attack-surface and overhead nevertheless remain, it is best to disallow overlapping fragments within in-car-networks. What remains is virtually identical to the IpV6 specification. IpV6 provides the newer and more consistent specification without the compatibility support for 40 years of hardware and structures, hence more suitable for automotive applications.

Note: This does not affect the fact that IpV4 uses the tuple (SrcIp32,DestIp32,Id16,Protocol8) as unique reassembling ID, whereas IpV6 omits the protocol, i.e.: only uses (SrcIp128,DestIp128,Id32). This is maintained throughout this implementation.

Requirements:

SWS_TCPIP_00054 SWS_TCPIP_00102 SWS_TCPIP_00231

- IPv4 fragmentation/reassembly shares configuration with IpV6 fragmentation/reassembly

Description:

IPv4 and IPv6 share configuration parameters. The TcpIp module uses a single configuration for IP fragmentation and reassembly. It is named `TcpIpIpFragmentationConfig` and is located in the `TcpIpIpConfig` tab. The configuration parameters are:

- `TcpIpIpFragMemReserved`

  Size of internal fragmentation and reassembly data in units of bytes (static memory allocation) - Memory required by post-build configuration must be smaller than this constant.

- `TcpIpIpFragmentationRxEnabled`

  Enables or disables support for reassembling of incoming datagrams that are fragmented according to IETF RFC 815 (IP Datagram Reassembly Algorithms).

  - `true` IP Datagram Reassembly enabled
  - `false` IP Datagram Reassembly disabled

  The following parameters configure the details of the reassembly mechanism:

  - `TcpIpIpReassemblyTimeout`

    Time after which an incomplete datagram gets discarded. RFC1122 (from 1989) suggests a value between 60 and 120 seconds. A large value can quickly lead to reassembly buffer exhaustion if fragments are lost.

- `TcpIpIpReassemblyBufferCount`

  Number of fragmented IP datagrams that can be reassembled in parallel.

- `TcpIpIpReassemblyBufferSize`

  Size of each reassembly buffer.

- `TcpIpIpFragmentationTxEnabled`

  Enables or disables support for fragmenting outgoing datagrams according to IETF RFC 791 / RFC 2460.

  Available choices:

  - `OFF`

    IP Datagram splitting disabled.

  - `OUTOFORDER`

    The header fragment with the checksum will be transmitted last to avoid buffering.

  - `INORDER`

    All data will be buffered in Ethernet transmit buffers, so the first fragment with the header and the checksum can be transmitted first. Additional data will be needed to keep track of the Ethernet buffer handles. This can be configured by the following values:

    - `TcpIpIpTxFragmentBufferCount`

      Maximum number of transmit buffers. Number of fragmented IP datagrams that can be sent in parallel.

    - `TcpIpIpTxFragmentBufferSize`

      Maximum size of a transmitted packet. INORDER fragmentation does not allocate memory for the data, but instead stores the data in Ethernet buffers. The maximum number of Ethernet buffers per packet is configured in TcpIpIpTxFragmentSegmentCount. Multiplying that with the ethernet MTU size is the virtual buffer size, which is the limit for fragmented INORDER transmissions and must be configured here.

    - `TcpIpIpTxFragmentSegmentCount`

      Maximum number of transmit Ethernet buffers (fragments) per IP datagram and socket. Twelve bytes of data will be reserved per fragment and buffer to store the Ethernet buffer handles.

Rationale:

As the implementation is generic within the IP module and mostly independent of the IpV4 and IpV6 specifics, there can only be one configuration read in the init code. This is an IpV6-like configuration, using the IpV6 configuration parameter names (minus the IpV6 reference) and the EB extensions.

Hence the IpV4 and IpV6 parameters and references are unused in favor of the common configuration, which is placed in the IP configuration container.

Requirements:

ECUC_TcpIp_00077, ECUC_TcpIp_00078, ECUC_TcpIp_00079, ECUC_TcpIp_00080, ECUC_TcpIp_-00099, ECUC_TcpIp_00103, ECUC_TcpIp_00114, ECUC_TcpIp_00157, ECUC_TcpIp_00158, ECUC_-TcpIp_00159, ECUC_TcpIp_00160, ECUC_TcpIp_00161, ECUC_TcpIp_00162

- [IPV4] Broadcast addresses must be explicitly configured for reception

Description:

If an IP datagram is received with a destination broadcast address, it is received only if there is an explicit broadcast local address configured, and there are sockets bound to this broadcast local address.

Rationale:

This allows detailed control of filtering the accepted destination IP addresses.

Requirements:

SWS_TCPIP_00106

- [IPV4] Only a single Unicast Internet address per physical/virtual interface is supported

Description:

This TcpIp implementation supports a single (unicast) internet address per physical/virtual interface (as specified in IETF RFC 791, Chapter 2.3).

Requirements:

SWS_TCPIP_00053

- [IPV4] IPv4 option fields are not supported

Description:

This TcpIp implementation ignores received Option fields as part of the IpV4 header (as specified in IETF RFC 791, Chapter 3.1). Transmission of Options as part of the IpV4 header is not supported.

Requirements:

SWS_TCPIP_00053

- [IPV4] Certain ICMP messages are not supported

  Description:

  This TcpIp implementation does not transmit the ICMP messages of type

  - Parameter Problem Message

  Time Exceeded Message

  - Source Quench Message

  - Redirect Message

  - Timestamp Reply Message

  (as specified in IETF RFC 792).

  Requirements:

  SWS_TCPIP_00059

- [IPV4] No local multicast loopback

  Description:

  This TcpIp implementation does not locally loop back transmitted multicast datagrams. Thus, multicast messages transmitted will not be received by the local node, even if it is assigned to the multicast address.

  Rationale:

  This is a use-case for systems with independent processes communicating via TcpIp. In AUTOSAR the local communication is usually performed within the RTE, thus this feature is assumed to be superfluous.

  Requirements:

  SWS_TCPIP_00097

- Unbound sockets will not be automatically closed

  Description:

  If the last EthIf controller reaches the Offline state, unbound sockets will not be automatically closed.

  Rationale:

  It is assumed that the upper layer (e.g. Soad) will close all unbound sockets if the TcpIp calls SoAd_-LocalIpAddrAssignmentChg() with State TCPIP_IPADDR_STATE_UNASSIGNED.

  Requirements:

SWS_TCPIP_00077

- TcpIp_TcpTransmit() does not queue data in state SYN-SENT and SYN-RECEIVED

  Description:

  If TcpIp_TcpTransmit() is called with a socket in state SYN-SENT and SYN-RECEIVED the function reports TCPIP_E_NOTCONN to Det.

  Rationale:

  It is assumed that the upper layer will only transmit data in state ESTABLISHED after <Up>_TcpConnected() or <Up>_TcpAccepted() is called.

  Requirements:

  SWS_TCPIP_00061

- TcpIp_Close() does not report TCPIP_E_NOTCONN to Det if socket is CLOSED

  Description:

  If TcpIp_Close() is called with abort equals TRUE or FALSE and a socket in state CLOSED, <Up>_TcpEvent() with TCPIP_TCP_CLOSED is called.

  Rationale:

  If the socket is unused or bound the upper layer need to be informed that the socket can no longer be used.

  Requirements:

  SWS_TCPIP_00061

- Reception of SYN segment in state TIME-WAIT does not re-establish a connection

  Description:

  If SYN segment is received in state TIME-WAIT a reset is transmitted and the connection is closed.

  Requirements:

  SWS_TCPIP_00104

- Non-compliant deviations in the vendor-specific module definition file

  Description:

  The vendor-specific module definition file (VSMD) has non compliant deviations to the AUTOSAR specification:

The following configuration parameters are in the pre-compile configuration class instead of the link configuration class:

- `TcpIpArpTableSizeMax`
- `TcpIpLocalAddrIpv4EntriesMax`
- `TcpIpLocalAddrIpv6EntriesMax`
- `TcpIpUdpSocketMax`
- `TcpIpTcpSocketMax`

Rationale:

Making the parameters `TcpIpArpTableSizeMax`, `TcpIpLocalAddrIpv4EntriesMax`, `TcpIpLocalAddrIpv6EntriesMax`, `TcpIpUdpSocketMax`, `TcpIpTcpSocketMax` pre-compile configurable allows for significant performance optimizations.

- Non-compliant deviations in the vendor-specific module definition file

Description:

The vendor-specific module definition file (VSMD) has non compliant deviations to the AUTOSAR specification:

The valid multiplicity of the configuration parameter `TcpIpNdpConfig` is from 0 to 1, which exceeds the range of 1 to * defined in the AUTOSAR specification.

Rationale:

Although, the configuration of NDP is required for IPv6, NDP does not need to be configured if the node supports IPv4 only.

- Non-compliant deviations in the vendor-specific module definition file

Description:

The vendor-specific module definition file (VSMD) has non compliant deviations to the AUTOSAR specification:

The valid range of the configuration parameter `TcpIpAssignmentPriority` is from 1 to 4, which exceeds the range of 1 to 3 defined in the AUTOSAR specification.

Rationale:

Extending the range of the configuration parameter `TcpIpAssignmentPriority` allows for the simultaneous configuration of all IPv4 assignment methods for one local address id.

- TcpIp does not support multiple configuration containers

Description:

TcpIp supports the configuration of a single container in the following lists only:

- TcpIpArpConfig

- TcpIpAutoIpConfig

- TcpIpDhcpConfig

- TcpIpIpV6DhcpConfig

- TcpIpIpV6NdpConfig

Requirements:

ECUC_TcpIp_00097, ECUC_TcpIp_00098, ECUC_TcpIp_00100, ECUC_TcpIp_00101, ECUC_TcpIp_-00102

- Router functionality not supported

Description:

TcpIp does not support any router functionality.

Requirements:

SWS_TCPIP_00160, SWS_TCPIP_00163

- TcpIp generates an ICMPv6 error message when receiving a packet sent as a link-layer multicast.

Description:

If the TcpIp receives a packet sent as a link-layer multicast and the packet contains an error the TcpIp will respond with an ICMPv6 error message.

Requirements:

SWS_TCPIP_00163

Rationale:

The destination link layer address is not passed to the TcpIp and therefore it is not possible to detect a link-layer multicast address. Usually, a link-layer multicast address is sent in combination with an IPv6 multicast address. If a packet destined to an IPv6 multicast address is received an ICMPv6 error message is not generated.

- [IPV6] Tunneling mechanism not supported

Description:

TcpIp does not support the encapsulation of IPv4 in the IPv6 header or vice versa.

Requirements:

SWS_TCPIP_00160

- [IPV4] DhcpV4 server not supported

Description:

TcpIp does not implementation a DhcpV4 server.

Requirements:

SWS_TCPIP_00200, SWS_TCPIP_00201, SWS_TCPIP_00218, SWS_TCPIP_00058, ECUC_TcpIp_00183, ECUC_TcpIp_00195, ECUC_TcpIp_00187, ECUC_TcpIp_00190, ECUC_TcpIp_00189, ECUC_TcpIp_00188, ECUC_TcpIp_00191

- Allocated DHCP addresses cannot be stored

Description:

TcpIp does not support to store an IP address allocated through Dhcp in an Nvm block in the Nvm module.

Requirements:

SWS_TCPIP_00219, ECUC_TcpIp_00186

- Stored addresses cannot be reset

Description:

TcpIp does not support API TcpIp_ResetIpAssignment for reseting all IP addresses stored in NvM block

Requirements:

SWS_TCPIP_00215, SWS_TCPIP_00216, SWS_TCPIP_00217, ECUC_TcpIp_00182

- [IPV4] IPv4 packet queuing not supported

Description:

TcpIp does not queue an IPv4 packet if the link layer address of the remote host does not exist in the ARP table and returns TCPIP_E_PHYS_ADDR_MISS to the caller.

Rationale:

If a UDP/ICMP frame is transmitted and an ARP entry does not exist the UDP/ICMP frame will be dropped and an ARP request will be transmitted instead. TCP frames will be retransmitted in the next mainfunction if an ARP entry does not exist. For UDP frames a packet queue can be configured in SoAd through con-

figuration parameter SoAdSocketUdpRetryEnabled in SoAdSocketConnectionGroup. An alternative way would be to configure a TcpIpIpV4ArpPacketFilter callout function which is called for every IPv4 frame which is received and matches the configured IP address. Through this callout it is possible to decide by the return value of this function if an ARP entry shall be created for the remote host. Per default an ARP entry is not created when an IPv4 frame is received if none exists to avoid unecessary ARP entries.

Requirements:

SWS_TCPIP_00191, SWS_TCPIP_00192, ECUC_TcpIp_00170

- [IPV6] Certain IPv6 Extension Headers are not supported

Description:

TcpIp does not support the reception and transmission of the following IPv6 Extension Headers:

- Authentication Header

- Encapsulating Security Payload Header

TcpIp does not support the transmission of the following IPv6 Extension Headers:

- Hop-by-Hop Options Header

- Routing Header

- Destination Options Header

Requirements:

SWS_TCPIP_00157, SWS_TCPIP_00160

- AUTOSAR API TcpIp_RequestIpAddrAssignment v4.2.2 not supported

Description:

TcpIp implements the AUTOSAR API TcpIp_RequestIpAddrAssignment according to v4.1.3.

Rationale:

Due to compatibility reasons to other modules (SoAd) the AUTOSAR API TcpIp_RequestIpAddrAssignment will not be updated to v4.2.2

Requirements:

SWS_TCPIP_00037, SWS_TCPIP_00079

- AUTOSAR API TcpIp_UdpTransmit v4.2.2 not supported

Description:

TcpIp implements the AUTOSAR API TcpIp_UdpTransmit according to v4.1.3.

Rationale:

Due to compatibility reasons to other modules (SoAd) the AUTOSAR API TcpIp_UdpTransmit will not be updated to v4.2.2

Requirements:

SWS_TCPIP_00025

- [IPV6] Certain rules of the IPv6 Source Address Selection are not supported

Description:

- The TcpIp does not support Rule 4: Prefer home addresses as described in (IETF RFC 6724). Rationale: IPv6 Mobility (IETF RFC 3375) which introduces home addresses is not required in AUTOSAR.

- The TcpIp does not support Rule 6: Prefer matching label as described in (IETF RFC 6724). Rationale: Prefix policy table cannot be configured in AUTOSAR.

- The TcpIp does not support Rule 7: Prefer temporary addresses. as described in (IETF RFC 6724). Rationale: According to SWS_TCPIP_00166 temporary addresses are not required in AUTOSAR.

Requirements:

SWS_TCPIP_00154

- [IPV6] IPv6 Loop back messages not supported

Description:

TcpIp does not verify if a neighbor solicitation which is sent to probe for a duplicate address is looped back. If a looped back neighbor solicitation is received, the TcpIp will interpret the neighbor solicitation as duplicate and will not assign the IP address to the interface

Requirements:

SWS_TCPIP_00157, SWS_TCPIP_00159

- [IPV6] Autoconfiguration issues related to MLD not supported

Description:

TcpIp does not send Multicast Listener Discovery messages. TcpIp does not support delaying of Neighbor Solicitation messages

Requirements:

SWS_TCPIP_00157

- [IPV6] IPv6 packet queuing not supported

  Description:

  TcpIp does not queue an IPv6 packet if the link layer address of the remote host does not exist in the NDP table and returns TCPIP_E_PHYS_ADDR_MISS to the caller.

  Rationale:

  If a UDP/ICMP frame is transmitted and a NDP entry does not exist the UDP/ICMP frame will be dropped and an Neighbor Soliciation will be transmitted instead. TCP frames will be retransmitted in the next main-function if an NDP entry does not exist. For UDP frames a packet queue can be configured in SoAd through configuration parameter SoAdSocketUdpRetryEnabled in SoAdSocketConnectionGroup.

  Requirements:

  SWS_TCPIP_00164, SWS_TCPIP_00165, SWS_TCPIP_00193, SWS_TCPIP_00194, ECUC_TcpIp_-00171, SWS_TCPIP_00164

- [IPV6] UDP/TCP IPv6 socket does not support IPv4 transmission

  Description:

  - A UDP/TCP IPv6 socket does not allow to bind an IPv4 local address.

  - A UDP IPv6 socket does not support to transmit messages to an IPv4 address embedded in an IPv6 address.

  - A TCP IPv6 socket does not support to connect to an IPv4 address embedded in an IPv6 address.

  Requirements:

  SWS_TCPIP_00162

- [IPV6] The DhcpV6 client does not collect multiple advertise messages

  Description:

  The DhcpV6 client does not wait and buffer multiple advertise messages until the first RT time elapses before responding to advertise message, instead it responds to first valid advertise message that is received.

  Rationale:

  Memory reservation if not required should be avoided.

  Requirements:

  SWS_TCPIP_00166

- [IPV6] API function TcpIp_IcmpV6Transmit() is not supported

Description:

TcpIp does not support the API function TcpIp_IcmpV6Transmit().

Rationale:

TcpIp_IcmpTransmit is used to transmit an icmp message instead of TcpIp_IcmpV6Transmit

Requirements:

SWS_TCPIP_00187, SWS_TCPIP_00230

- [IPV6] IpV6 configuration parameters not supported

Description:

The following IpV6 configuration parameters are not supported:

- TcpIpIpVXCtrl

Requirements:

ECUC_TcpIp_00094

- [IPV6] DhcpV6 configuration parameters not supported

Description:

The following DhcpV6 configuration parameters are not supported:

- TcpIpDhcpV6CnfDelayMax

- TcpIpDhcpV6CnfDelayMin

- TcpIpDhcpV6InfDelayMax

- TcpIpDhcpV6InfDelayMin

- TcpIpDhcpV6SolDelayMax

- TcpIpDhcpV6SolDelayMin

Requirements:

ECUC_TcpIp_00116, ECUC_TcpIp_00117, ECUC_TcpIp_00118, ECUC_TcpIp_00119, ECUC_TcpIp_-00120, ECUC_TcpIp_00121

- [IPV6] Dynamic reconfiguration of MTU via Router Advertisements is not supported

Description:

The TcpIp does not update the MTU according to the value received in a router advertisement, the configured MTU (EthIfCtrlMtu) is used instead.

Requirements:

SWS_TCPIP_00153, SWS_TCPIP_00157, SWS_TCPIP_00160, SWS_TCPIP_00164

- [IPV6] Dynamic reconfiguration of hop limit via Router Advertisements is not supported

Description:

The TcpIp does not update the hop limit according to the value received in a router advertisement, the configured hop limit (TcpIpUdpTtl, TcpIpTcpTtl, TcpIpIcmpTtl, TcpIpIcmpV6HopLimit) is used instead.

Requirements:

SWS_TCPIP_00164, SWS_TCPIP_00157

- [IPV6] Dynamic reconfiguration of reachable time via Router Advertisements is not supported

Description:

The TcpIp does not update the reachable time according to the value received in a router advertisement, the configured reachable time (TcpIpNdpDefaultReachableTime) is used instead.

Requirements:

SWS_TCPIP_00164, SWS_TCPIP_00157

- [IPV6] Dynamic reconfiguration of retransmit timer via Router Advertisements is not supported

Description:

The TcpIp does not update the retransmit timer according to the value received in a router advertisement, the configured retrnasmit timer (TcpIpNdpDefaultRetransTimer) is used instead.

Requirements:

SWS_TCPIP_00164, SWS_TCPIP_00157

- [IPV6] Ndp configuration parameters not supported

Description:

The following Ndp configuration parameters are not supported:

- TcpIpNdpDynamicHopLimitEnabled

- TcpIpNdpDynamicMtuEnabled

- TcpIpNdpDynamicReachableTimeEnabled

- TcpIpNdpDynamicRetransTimeEnabled

- TcpIpNdpAddressResolutionUnrechabilityDetectionEnabled

- TcpIpNdpMinRandomFactor

- TcpIpNdpMaxRandomFactor

- TcpIpNdpDefaultReachableTime

- TcpIpNdpNeighborUnreachabilityDetectionEnabled

- TcpIpNdpRandomReachableTimeEnabled

Requirements:

ECUC_TcpIp_00146, ECUC_TcpIp_00147, ECUC_TcpIp_00148, ECUC_TcpIp_00145, ECUC_TcpIp_-00091, ECUC_TcpIp_00134, ECUC_TcpIp_00135, ECUC_TcpIp_00130, ECUC_TcpIp_00136, ECUC_-TcpIp_00137

- [IPV6] Certain DhcpV6 message types are not supported

  Description:

  DhcpV6 does not support the transmission/reception of the following message types:

  - Information-request Message

  - Release Message

  - Confirm Message

  - Reconfigure Message

  Requirements:

  SWS_TCPIP_00166

- [IPV6] Multiple IA_ADDR options in DhcpV6 messages are not supported

  Description:

  DhcpV6 does not support the transmission of the multiple IA_ADDR options in single IA_NA option. DhcpV6 shall only process last IA_ADDR option and ignore others in single IA_NA option.

  Requirements:

  SWS_TCPIP_00166

- [IPV6] IA_NA options with times T1 == T2 in DhcpV6 messages are not supported

Description:

DhcpV6 discards IA_NA options with times T1 == T2

Requirements:

SWS_TCPIP_00166

- Oversized ICMPv4/v6 Echo Reply is not fragmented

Description:

If the TcpIp receives an Echo Request greater than the MTU the TcpIp will not transmit the Echo Reply in IP fragments, it will truncate the size of the Echo Reply to the MTU instead and send the reply in a single IP frame.

Requirements:

SWS_TCPIP_00163, SWS_TCPIP_00059

- [IPV6] Passing of ICMPv6 error messages to upper layer is not supported

Description:

UDP and TCP do not evaluate an ICMP error message from a remote host. The ICMP message is passed to the configurable Up_IcmpMsgHandler instead.

Requirements:

SWS_TCPIP_00163

- [IPV6] Limitation of ICMPv6 packet transmissions not supported

Description:

TcpIp does not allow to configure a limit for the transmission of ICMPv6 error messages to the the same destination and ICMPv6 error messages transmissions per second.

Requirements:

SWS_TCPIP_00163

- [IPV6] IcmpV6 Time Exceeded Message not supported

Description:

TcpIp does not transmit a IcmpV6 Time Exceeded Message when the TcpIp cannot complete the reassembly due to missing fragments within the configured time limit. It will discard the datagram only.

Requirements:

SWS_TCPIP_00161, SWS_TCPIP_00163

- Scalability classes are not supported

  Description:

  The configuration parameter TcpIpScalabilityClass is not supported.

  Requirements:

  SWS_TCPIP_00148, SWS_TCPIP_00149, SWS_TCPIP_00150, ECUC_TcpIp_00169

- [IPv6] ICMP destination unreachable not supported

  Description:

  The TcpIp does not transmit a ICMP destination unreachable with code 3 (Address Unreachable) for each packet queued for address resolution if the node does not receive a Neighbor Advertisement after the transmission of the maximal number of Multicast Neighbor Solicitations. Instead, the neighbor is removed from the neighbor cache.

  Requirements:

  SWS_TCPIP_00164

- [IPV6] Automatic assignment of IPv6 link local address not supported

  Description:

  The TcpIp does not automatically assign the IPv6 link local address to every configured controller

  Rationale:

  The IPv6 link local address is configurable for every interface

  Requirements:

  SWS_TCPIP_00162

- [IPV6] identification via interface ID not supported

  Description:

  TcpIp does not use interface ID to identify interfaces on a link

  Requirements:

  SWS_TCPIP_00162

- [IPV6] loopback address not supported

Description:

This TcpIp implementation does not locally loop back messages transmitted to the loop back address and does prevent the assignment of a loop back address

Requirements:

SWS_TCPIP_00162

- [IPV6] multicast address reserved fields not supported

Description:

This TcpIp implementation ignores the reserved field of an IPV6 multicast address

Requirements:

SWS_TCPIP_00162

- [IPv6] DUID-LL shall not be used if network interface is not permanently attached to the device

Description:

DUID-LL (DHCP Unique Identifier - Link-layer Address) is used by the TcpIp to identify a server in messages where a server needs to be identified.

Rationale:

DUID-LL (DHCP Unique Identifier - Link-layer Address) is the only way to implement a DUID in TcpIp because there is no parameter in AUTOSAR to set an Enterprise Number [DUID-EN] and no clock to generate a DUID-LLT (DHCP Unique Identifier - Link-layer address plus time). Moreover the network interface of an ECU will most likely not change over time

Requirements:

SWS_TCPIP_00166

- [IPv6] Source Address Selection of unbound IPv6 UDP sockets

Description:

TcpIp does not select an interface that has a local address (TcpIpAddrId) which uses the same network prefix as the destination address if data is transmitted using an unbound IPv6 UDP socket and then performs source address selection for the selected interface. Instead, the TcpIp loops over all local addresses and performs source address selection.

Requirements:

SWS_TCPIP_00185

- [IPv6] Anycast addresses not supported

  Description:

  TcpIp does not support the assignment of Anycast addresses to a TcpIp controller. Messages can still be transmitted to an Anycast address.

  Requirements:

  SWS_TCPIP_00269, SWS_TCPIP_00162
- [IPv6] IPv6 Redirect message not supported

  Description:

  TcpIp does not process received IPv6 Redirect messages, it discards the messages instead

  Requirements:

  SWS_TCPIP_00281, SWS_TCPIP_00164
- Configurable Path MTU discovery not supported

  Description:

  The path MTU discovery cannot be turned on or off for a socket through TcpIp_ChangeParamter and paramId

  - TCPIP_PARAMID_PATHMTU_ENABLE

  Requirements:

  SWS_TCPIP_00267, SWS_TCPIP_00268
- [IPV6] Certain sections of the IPv6 Subnet Model are not supported

  Description:

  - The TcpIp does not support section 4: Host Rules in (IETF RFC 5942).

  - The TcpIp does not support Section 6: updated definition of "on-link" in (IETF RFC 5942).

  Requirements:

  SWS_TCPIP_00265
- Some runtime errors not supported

  Description:

  Module TcpIp does not support the following runtime errors:

- TCPIP_E_TIMEDOUT
- TCPIP_E_CONNREFUSED
- TCPIP_E_HOSTUNREACH
- TCPIP_E_PACKETTOBIG
- TCPIP_E_DADCONFLICT

Requirements:

SWS_TCPIP_00157, SWS_TCPIP_00255, SWS_TCPIP_00256, SWS_TCPIP_00257, SWS_TCPIP_-00258, SWS_TCPIP_00259, SWS_TCPIP_00282

- [IPv6] On-link prefix list not supported

  Description:

  TcpIp does not support the following configuration parameters for on-link prefix configuration:

  - TcpIpNdpPrefixList

  - TcpIpNdpPrefixListEntry

  - TcpIpNdpPrefixListEntryPrefixLength

  - TcpIpNdpPrefixListEntryPrefixAddress

  Requirements:

  ECUC_TcpIp_00205, ECUC_TcpIp_00206, ECUC_TcpIp_00207, ECUC_TcpIp_00208

- [IPv6] TcpIp does not support notifications in case of a detected address conflict

  Description:

  The TcpIp does not notify the configuring agent when an IpV6 address conflict is detected during Ongoing Address Conflict Detection.

  Requirements:

  SWS_TCPIP_00283

- [IPv6] TcpIp allows IPv6 UDP packets with zero checksum.

  Description:

  Since introduction of configurable UDP checksum calculation UDP packets with checksum field set to zero are accepted if UDP checksum calculation is disabled

  Rationale:

Per default the UDP packet is discarded if it contains a zero checksum but zero checksum fields can be allowed through TcpIp_ChangeParameter.

Requirements:

SWS_TCPIP_00185

- TcpIp does not support variant handling for the following TcpIp parameters

Description:

The TcpIp does not support postbuild selectable or loadable for

- TcpIpSocketOwnerUpperLayerType

Rationale:

Due to the fact that the underlying socket owner parameter, e.g. TcpIpSocketOwnerCopyTxDataName, TcpIpSocketOwnerLocalIpAddrAssignmentChgName, ... are link time configurable it is not applicable that TcpIpSocketOwnerUpperLayerType is postbuild selectable or loadable

Requirements:

ECUC_TcpIp_00174

- [IPsec] TcpIp does not support ESP

Description:

TcpIp does not support ESP. If an ESP Header is detected, the frame will be dropped!

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301, 4302, 4543, 5282, 4106.

- [IPsec] TcpIp does not support Tunnel mode

Description:

TcpIp does not support Tunnel mode. If an inner IP header is detected, the frame will be dropped!

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- [IPsec] TcpIp does not support filtering by DSCP

Description:

Secuirty Assocation database entry does not contain DSCP values and DSCP-specific filtering is not applied

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- [IPsec] TcpIp does not support to configure Mutlicast Security Associations

  Description:

  Multicast communication can only be performed unprotected and shall be configured as bypassed in the Security Policy Table

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 4301.

- [IPsec] TcpIp does not support nested SAs

  Description:

  TcpIp allows to apply a single AH to a frame in the SPD only and does not support to combine multiple AHs or ESPs. If the TcpIp encounters more than one AH or an ESP in a frame, the frame will be dropped

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 4301.

- [IPsec] TcpIp does not support Security Gateways

  Description:

  TcpIp does not support the following Security Gateway (SG) features:

  - Discovery of and communication through SGs.
  - Acting as a SG.

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 4301.

- [IPsec] TcpIp does not support additional SPD creation or changes during the runtime

  Description:

  TcpIp only allows the configuration of static security policies. Security policies do not change during runtime and cannot be updated, added or deleted through an API

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 4301.

- TcpIp does not support mobility

Description:

Mobility Support in IPv6 as defined in IETF RFC 6275 and IP Mobility Support for IPv4 as defined in IETF RFC 5944 is not implemented

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- [IPsec] Certain IPsec error are not logged

Description:

TcpIp does not provide any kind of audit logs and does not log the following IPsec error:

- Invalid SPI received
- Sequence number overflow
- IP Fragment passed to AH processing
- No security association found
- ICV validation failed

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301 and 4302.

- [IPsec] Certain HMAC algorithms for AH defined in IETF RFC 4868 are not supported

Description:

TcpIp does not support the following HMAC integrity algorithm for AH

- HMAC-SHA-384-192
- HMAC-SHA-512-256

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4868.

- [IPsec] Certain GMAC algorithms defined in IETF RFC 4543 are not supported

Description:

TcpIp does not support the following GMAC integrity algorithm for AH

- AUTH_AES_192_GMAC

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4543.

- [IPsec] GMAC Initialization Vector can repeat

  Description:

  TcpIp allows use of GMAC with statically configured keys, without re-keying IV can repeat

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 4543.
- [IPsec] TcpIp does not provide support for partially matching incoming packets to SAD entries.

  Description:

  TcpIp only supports full matching of incoming packets to Security Association Database entry, i.e. SPI, destination address and source address of the incoming packet has to the values of an SA in the database for a successful match. Partial matches e.g. SPI and destination address or SPI only are not supported.

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 4302.
- [IPsec] TcpIp does not provide support PFP flag

  Description:

  TcpIp does not support to configure the "populate from packet" flag for an SPD entry which states if the value (e.g. next header protocol, ip address) for an SAD entry shall be taken from the packet or the SPD. The TcpIp always takes the value from the SPD entry.

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 4301.
- [IPsec] TcpIp does not support ICMP filtering based on ICMP Code and Type

  Description:

  TcpIp does not support to configure an SPD entry based on the ICMP Code and Type. It is only possible to configure all ICMP frames for a specific local and remote IP address to be either secured, bypassed or discarded.

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 4301.
- [IPsec] TcpIp does not support certain identifier types

  Description:

TcpIp does not support configuration in the SPD and identification of a remote host by the following identifier types:

- Fully qualified DNS name
- Fully qualified user name string (email)
- X.500 distinguished name
- Byte string

TcpIp identifies a remote host by an IP address only.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- [IPsec] TcpIp does not support the Sequence Counter Overflow in SAD

Description:

TcpIp does not support the configuration of a Sequence Counter Overflow flag which indicates if sequence number overflow is permitted. For security association which are manually configured the sequence number is not checked and it can overflow.

Rationale:

Manually configured security association shall not use anti-reply and Extended sequence number because the security association might not be synchronized (e.g. one of the two hosts might restart and reset the sequence number counter)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- [IPsec] TcpIp does not send ICMP Error Message when received frame is discarded

Description:

TcpIp does not send an ICMP error message when a frame is discarded in the following cases:

- No matching SPD entry was found
- The TcpIp reached the remote peer but was unable to negotiate the SA required by the SPD entry matching the packet because the remote peer is administratively prohibited from communicating with the initiator, the initiating peer was unable to authenticate itself to the remote peer, the remote peer was unable to authenticate itself to the initiating peer, or the SPD at the remote peer did not have a suitable entry.
- The TcpIp was unable to set up the SA required by the SPD entry matching the packet because the IPsec peer at the other end of the exchange could not be contacted.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- [IPsec] TcpIp does not support SPD cache

Description:

TcpIp does not support caching of recently used/created SPD. Additional entries cannot be added during runtime. Statically configured table is used for lookup.

Rationale:

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- [IPsec] TcpIp does not support SPD ID

Description:

TcpIp does not support SPD-ID instead it directly searches SPD-S and SPD-O

Rationale:

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- [IKEv2] Certain ECP Groups defined in IETF RFC 5903 are not supported

Description:

TcpIp does not support the following Diffie-Hellman Group Transforms for IKEv2

- 521-bit Random ECP Group

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 5903.

- [IKEv2] Certain algorithms defined in IETF RFC 8247 are not supported

Description:

TcpIp does not support the following encryption algorithms for IKEv2

- ENCR_CHACHA20_POLY1305
- ENCR_AES_GCM_16 with 128 bit key

TcpIp does not support the following pseudorandom function for IKEv2

- PRF_HMAC_SHA2_512
- PRF_HMAC_SHA1

TcpIp does not support the following integrity algorithm for IKEv2

- AUTH_HMAC_SHA2_512_256
- AUTH_HMAC_SHA1_96

TcpIp does not support the following Diffie-Hellman group for IKEv2

- 2048-bit MODP Group

TcpIp does not support the following authentication method for IKEv2

- RSA Digital Signature

TcpIp does not support the following digital signature authentication method for IKEv2

- RSASSA-PSS with SHA-256

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 8247.

- [IKEv2] Certain algorithms defined in IETF RFC 8221 are not supported

Description:

TcpIp does not support the following algorithms for AH authentication

- HMAC_SHA2_512_256
- HMAC_SHA1_96
- AES_XCBC_96
- AUTH_NONE

TcpIp does not support any encryption or authentication algorithms for ESP

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 8221.

- [IKEv2] Certain algorithms defined in IETF RFC 7427 are not supported

Description:

TcpIp does not support the following algorithms for signature authentication

- ECDSA-with-SHA1
- RSA-with-SHA1

- RSA-with-SHA256
- RSA-with-SHA384
- RSA-with-SHA512
- DSA-with-SHA1
- DSA-with-SHA256
- RSASSA-with-EmptyParameters
- RSASSA-with-DefaultParameters
- RSASSA-with-SHA256

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7427.

- [IKEv2] Repeated Authentication in IKEv2 Protocol defined in RFC 4478 is not supported

Description:

AUTH_LIFETIME notification is not transmitted when TcpIp is responder and is ignored when TcpIp is initiator

Rationale:

TcpIp does not use EAP and Configuration payloads meaning that creation of a new IKE SA can be initiated by either party (initiator or responder in the original IKE SA)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4478.

- [IKEv2] Additional transforms of known type are ignored

Description:

Received proposals are accepted as long as the locally configured transforms constitute a subset of the received transforms and all additional transforms are either ENCR, PRF, INTEG, D-H or ESN. E.g. if a proposal is received with combined cipher mode and integrity transform different from "NONE" the proposal is accepted as long as the locally configured proposal is a subset of the received proposal.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Certain HMAC Algorithm for Pseudorandom Function in IKEv2 according to IETF RFC 4868 are not supported

Description:

TcpIp does not support the following HMAC algorithm for the Pseudorandom Function Transform in IKEv2:

- PRF_HMAC_SHA2_512

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4868.

- [IKEv2] Certain HMAC Algorithm for authentication and integrity verification in IKEv2 according to IETF RFC 4868 are not supported

Description:

TcpIp does not support the following HMAC algorithm for the Integrity Algorithm Transform in IKEv2:

- AUTH_HMAC_SHA2_512_256

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4868.

- [IKEv2] Re-keying of IPsec Security Association not supported

Description:

TcpIp does not allow to re-key an IPsec security association through the CREATE_CHILD_SA exchange after a configured amount of time, specified counter value such as number of received bytes or when the maximal sequence number is reached. The life time of an IPSec security association is equal to the life time of the IKE security association.

If a CREATE_CHILD_SA request is received which requests re-keying of the IPSec SA the TcpIp will response with a NO_ADDITIONAL_SAS notification.

Rationale:

An IPsec security association can be refreshed through the re-authentication of the IKE security association. (creating a new IKE SA from scratch by using IKE_SA_INIT/IKE_AUTH exchanges)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301, 7296.

- [IKEv2] Re-keying of IKE Security Association not supported

Description:

TcpIp does not allow to re-key an IKE security association through the CREATE_CHILD_SA exchange after a configured amount of time.

If a CREATE_CHILD_SA request is received which requests re-keying of the IKE SA the TcpIp will response with a NO_ADDITIONAL_SAS notification.

Rationale:

An IKE security association can be refreshed through the re-authentication of the IKE security association. (creating a new IKE SA from scratch by using IKE_SA_INIT/IKE_AUTH exchanges)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Certain Id types to identify a remote host in the connection table are not supported

Description:

TcpIp does not support the Id types to identify a remote host in IKEv2:

- DNS name (specific or partial)
- RFC 822 email address (complete or partially qualified)
- Key ID (exact match only)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- [IKEv2] Only Key Exchange version 2 (IKEv2) is supported

Description:

TcpIp does not support any version of the Internet Key Exchange other than 2. In particular version 1 (IKEv1) defined in IETF RFC 4109 is not supported.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301, 7296.

- [IPsec] Anti-replay service not configurable for Security Association

Description:

- For dynamically configured security association (through IKEv2) the Anti-replay service is always turned on per default and cannot be turned off.
- For manually configured security association the Anti-replay service is always turned off per default and cannot be turned on.

Rationale:

Manually configured security association shall not use anti-reply and Extended sequence number because the synchronization of the security association cannot be guaranteed (e.g. one of the two hosts might restart and reset the sequence number counter)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- [IPsec] Extended Sequence Number (ESN) not configurable for Security Association

Description:

- TcpIp supports 64 bit sequence number only, 32 bit seq numbers are not supported.
- For dynamically configured security association (through IKEv2) the ESN is always turned on per default and cannot be turned off.
- For manually configured security association the ESN is always turned off per default and cannot be turned on.

Rationale:

Manually configured security association shall not use anti-reply and Extended sequence number because the security association might not be synchronized (e.g. one of the two hosts might restart and reset the sequence number counter)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- [IKEv2] IKE notification INVALID_SELECTORS not supported

Description:

The TcpIp does not send the IKE notification INVALID_SELECTORS to the sender (IPsec peer), which indicates that the received packet was discarded because of failure to pass selector checks. If the TcpIp receives such a IKE notification, e.g. in an INFORMATIONAL exchange the notification is ignored.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4301.

- [IPsec] Synchronization due to Significant Packet Loss not supported

Description:

If there is an undetected packet loss of 2^32 or more consecutive packets on a single SA, then the transmitter and receiver will lose synchronization of the high-order bits. The TcpIp will not try re-synchronize the sequence number.

Rationale:

The possibility of data loss in a vehicle is very low. IKE will detect that a remote host is not reachable anymore through Dead Peer Detection before 2^32 packets are lost and will try to re-initiate the creation of a new IKE SA.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 4302 and RFC 6479.

- [IKEv2] Mixed preshared key and certificate authentication

Description:

When host 1 uses preshared key authentication and host 2 uses certificate authentication method "digital signature" then only one SIGNATURE_HASH_ALGORITHMS notify is sent from the host 1 to host 2 containing the supported signature hash algorithms. No SIGNATURE_HASH_ALGORITHMS notify is sent from host 2 to host 1.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7427.

- [IKEv2] Certain algorithms defined in IETF RFC 5282 and RFC 4106 are not supported

Description:

TcpIp does not support the following GCM encryption algorithms for IKEv2

- AES-GCM with 16-octet ICV and 128 bit key
- AES-GCM with 16-octet ICV and 192 bit key
- AES-GCM with 12-octet ICV and 128 bit key
- AES-GCM with 12-octet ICV and 192 bit key
- AES-GCM with 8-octet ICV and 128 bit key
- AES-GCM with 8-octet ICV and 192 bit key
- AES-GCM with 8-octet ICV and 256 bit key

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 5282, IETF RFC 4106.

- [IKEv2] Extensible Authentication Protocol not supported

Description:

TcpIp does not support authentication using Extensible Authentication Protocol (EAP)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] IP Compression not supported

  Description:

  TcpIp does not support IP Compression (IPComp)

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Configuration Payload not supported

  Description:

  TX: TcpIp does not transmit Configuration Payloads. RX: When receiving an INFORMATIONAL request with Configuration Payloads (CP) an empty INFORMATIONAL response is sent back.

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Creating additional IPsec Security Association not supported

  Description:

  TcpIp does not allow creating additional IPsec security association through the CREATE_CHILD_SA exchange. Only one IPsec security association is created in the AUTH exchange.

  If a CREATE_CHILD_SA request is received TcpIp will response with a NO_ADDITIONAL_SAS notification.

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] IKE cookies not supported

  Description:

  TcpIp does not support the use of cookies in the IKE exchange and is not protected against flooding attacks

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] IKE address allocation not supported

Description:

TcpIp does not support address allocation to an IPsec Remote Access Client (IRAC) trying to tunnel into a network protected by an IPsec Remote Access Server (IRAS)

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] IKE ignores messages outside of an IKE SA

Description:

To avoid flooding and reduce number of messages TcpIp discards messages outside of an IKE SA

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] IKE notification SET_WINDOW_SIZE not supported

Description:

The TcpIp does not send the IKE notification SET_WINDOW_SIZE indicating the sending endpoint is capable of keeping state for multiple outstanding exchanges

TcpIp as Initiator will never indicate it supports window size greater than 1 and will never send multiple requests before receiving response to the first.

If TcpIp as responder receives SET_WINDOW_SIZE indicating the remote host supports windows size greater than 1 it will never send multiple requests before receiving response to the first.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] No IKE SA is created without an associated child SA

Description:

If no child SA is created during the AUTH exchange due to e.g. NO_PROPOSAL_CHOSEN the underlying IKE SA will be deleted since only one child SA is supported per IKE SA and there is no way to create additional child SA via CREATE_CHILD_SA exchange.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Only a complete match of the traffic selectors is supported

Description:

When negotiating a Child SA during the IKE_AUTH exchange the traffic selectors contained in the IKE_-AUTH response must match the traffic selectors in the IKE_AUTH request completely. No narrowing shall be done by the responder. In case a complete match is not found an TS_UNACCEPTABLE notify is sent in the IKE_AUTH response.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] IP Compression is not supported

  Description:

  IP Compression is not supported by TcpIp. IPCOMP_SUPPORTED notifications are never send and ignored on receipt.

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] INVALID_SPI notification is not supported

  Description:

  TX: TcpIp does not transmit INVALID_SPI notification when AH packet with unrecognized SPI is received. The packet with unrecognized SPI in AH header is silently dropped. RX: INVALID_SPI notification is considered an error condition and the IKE SA is closed

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] INVALID_IKE_SPI notification is not supported

  Description:

  TX: TcpIp does not transmit INVALID_IKE_SPI notification if IKE message packet arrives on port 500 with an unrecognized IKE SPI. The message with unrecognized SPI in IKE header is silently dropped. RX: INVALID_IKE_SPI notification is considered an error condition and the IKE SA is closed

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] INVALID_MAJOR_VERSION notification is not supported

  Description:

TX: TcpIp does not transmit INVALID_MAJOR_VERSION notification when receiving an IKE message with major version number different from 2. The message with invalid major version in IKE header is silently dropped. RX: INVALID_MAJOR_VERSION notification is considered an error condition and the IKE SA is closed

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] INVALID_MESSAGE_ID notification is not supported

  Description:

  TX: TcpIp does not transmit INVALID_MESSAGE_ID notification when an IKE Message ID outside the supported window is received. The received message with invalid message ID is silently dropped. RX: INVALID_MESSAGE_ID notification is considered an error condition and the IKE SA is closed

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] UNSUPPORTED_CRITICAL_PAYLOAD notification is not supported

  Description:

  Critical unsupported payloads are not expected from the remote host and their presence is considered an error condition and IKE SA is closed. Due to security concerns there is no UNSUPPORTED_CRITICAL_PAYLOAD notification send in the response in order to mitigate the posibility of flooding attacks. TX: TcpIp does not transmit UNSUPPORTED_CRITICAL_PAYLOAD notification when an IKE Message is received with unrecognized payload type whose critical flag is set. RX: UNSUPPORTED_CRITICAL_PAYLOAD notification is considered an error condition and the IKE SA is closed

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] TEMPORARY_FAILURE notification is not supported

  Description:

  TX: TcpIp does not transmit TEMPORARY_FAILURE notification when an IKE Message is received that cannot be completed due to a temporary condition RX: TEMPORARY_FAILURE notification is considered an error condition and the IKE SA is closed

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] SINGLE_PAIR_REQUIRED notification is not supported

Description:

TX: TcpIp does not transmit SINGLE_PAIR_REQUIRED notification when an IKE Message is received with TS payload that contains more than a single pair. RX: SINGLE_PAIR_REQUIRED notification is considered an error condition and the IKE SA is closed

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] CHILD_SA_NOT_FOUND notification is not supported

Description:

TX: TcpIp does not transmit CHILD_SA_NOT_FOUND notification when an CREATE_CHILD_SA request is received for rekeying an non-existant child SA. Instead all CREATE_CHILD_SA requests are responded to with NO_ADDITIONAL_SAS. RX: CHILD_SA_NOT_FOUND notification is considered an error condition and when received in an response the IKE SA is closed

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Only port 500 is open for IKE exchange

Description:

Ike exchange can only occur over port 500. Messages received at port 4500 are silently dropped

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] No replicated entities are allowed

Description:

One-to-one correspondence between identities and hosts.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Identification of IKE SAs

Description:

A given IKE SA is not identified only by the local SPI as recommended by the Rfc but is identified by the initiator SPI + remote IP address for init exchanges and initiator SPI + responder SPI + remote IP address for other exchanges.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Identification of Child SAs

  Description:

  A given Child SA is not identified only by the pair of SAs as recommended by the Rfc but is identified by the single SA that have remote and local SPI.

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Transmission of Delete AH request

  Description:

  TcpIp does not transmit Delete AH INFORMATIONAL Request because CHILD SA will be removed with an IKE SA

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Transforms with restricted key space

  Description:

  TcpIp does not support transform algorithms for which not all values are valid keys.

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Transport Mode NAT Traversal is not supported

  Description:

  The current implementation does not support NAT Traversal

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Transform Attribute

  Description:

  Only the "key length" attribute is currently supported.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Diffie-Hellman transform with ID "NONE"

  Description:

  A Diffie-Hellman transform with ID "NONE" (numerical value 0) is never offered nor accepted in an IKE SA proposal.

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Supported Identification IDs

  Description:

  The following ID types are the onyl ones supported in Identification payloads: ID_IPV4_ADDR, ID_IPV6_-ADDR, ID_DER_ASN1_DN

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Supported Certificate Encoding

  Description:

  The TcpIp does only support the following Certificate Encoding of Certificates in the CERT payload:

  - X.509 Certificate - Signature (value 4)

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Supported Authentication Method

  Description:

  The TcpIp does only support the following Authentication Method in the AUTH payload:

  - Shared Key Message Integrity Code (value 2)
  - Digital Signature (14)

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] No logging of unrecognized notification types

Description:

Unrecognized notify types of notify payloads in any received IKE request or response are not logged.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Unsupported notification types

Description:

The following notification types are handeled as unrecognized by the current implementation:

- INTERNAL_ADDRESS_FAILURE
- FAILED_CP_REQUIRED (see deviation "Configuration Payload not supported")
- ADDITIONAL_TS_POSSIBLE (see deviation "Only a complete match of the traffic selectors is supported")
- IPCOMP_SUPPORTED (see deviation "IP Compression not supported")
- NAT_DETECTION_SOURCE_IP (see deviation "Transport Mode NAT Traversal is not supported")
- NAT_DETECTION_DESTINATION_IP (see deviation "Transport Mode NAT Traversal is not supported")
- COOKIE (see deviation "IKE cookies not supported")
- HTTP_CERT_LOOKUP_SUPPORTED (see deviation "Supported Certificate Encoding")
- REKEY_SA (see deviation "Re-keying of IPsec Security Association not supported")
- ESP_TFC_PADDING_NOT_SUPPORTED (see deviation "TcpIp does not support ESP")
- NON_FIRST_FRAGMENTS_ALSO (see deviation "Creating additional IPsec Security Association not supported")

TX: TcpIp does not transmit these notification types RX: error notifications are considered fatal (IKE SA is closed), status notifications are ignores

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] The vendor ID payload is not supported

Description:

The vendor ID payload is never send and ignored on receipt.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] MODP Diffie-Hellman groups for IKEv2 are not supported

  Description:

  TcpIp does not support MODP Diffie-Hellman groups

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Error notifications in request messages

  Description:

  All error notifications in request messages are considered to be fatal error conditions causing the IKE SA to be closed.

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Address narrowing and port narrowing is not supported

  Description:

  TcpIp Does not support any type of narrowing.

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Certificate validity check is not supported

  Description:

  TcpIp does not support certificate validity check, i.e. if the certificate has expired.

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Immediate reconnection

  Description:

  If the INIT request is responded by an error notify this leads to an immediate reconnection attempt in case there is currently no other IKE SA active for the given connection. In case there is an active connection the IKE SA is closed without further reconnection attempts. The idea of this "never give up" approach is that the implementation should always try to establish a connection.

  Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] No error handling for INFORMATIONAL messages

  Description:

  If an error is detected during processing of an IKE INFORMATIONAL message or an error notification other than AUTHENTICATION_FAILED is present in the received INFORMATIONAL message the message is dropped without further error handling.

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Transmitted delete request messages

  Description:

  INFORMATIONAL IKE SA Delete request are only send 1. after reauthentication was succesfull (and the old IKE SA is deleted) 2. after receiving a CHILD SA delete message 3. after stopping an established connection (by calling TcpIp_IkeV2_stop)

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Multiple identities per host not supported

  Description:

  The IDr payload in AUTH requests are ignored since ECUs are assumed to have only one identity.

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] IP address ranges in traffic selectors not supported

  Description:

  TcpIp module does not support configuration of Ip address ranges for traffic selectors. Instead, IpSec connection remote IP address is set for both Starting Address and Ending Address fields for traffic selector.

  Requirements:

  No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] Ports of traffic selectors using ANY protocol can be configured.

  Description:

TcpIp supports configuration of port range in Traffic Selectors using protocol ANY.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] IKE Responder forgetting the response not supported

Description:

TcpIp Ike does not support forgetting the response after some time. Currently the response is forgotten if new message is transmitted or the connection is restarted.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] IKE SAs are not closed with invalid Message ID

Description:

TcpIp Ike does not support closing or rekeying IKE SA when Message ID overflows.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296.

- [IKEv2] IKE can't receive payloads before Encrypted

Description:

TcpIp Ike does not support transmitting and receiving payloads before Encypted and Authenticated or Encrypted Fragmented payload.

Requirements:

No AUTOSAR requirement. Deviation from IETF RFC 7296, IETF RFC 7383.

- TcpIp_IsConnectionReady() can't bind the socket

Description:

TcpIp_IsConnectionReady() does not support binding the UDP sockets to local resources. Even though TcpIp_UdpTransmit() would do the transmission and transmit the packet on the unbound socket, TcpIp_-IsConnectionReady() will not return OK nor bind the socket.

Requirements:

No AUTOSAR requirement. Deviation from EB requirement created from SWS_TCPIP_00122.

2.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- Limitation on number of entries in container `TcpIpLocalAddr`

  Description:

  The TcpIp can only handle 253 local addresses. This limitation applies to configuration parameter `TcpIpLocalAddr`.

  Rationale:

  LocalAddrId 254 and 255 are reserved for special values.

- Limitation on configuration parameter `TcpIpArpTableEntryTimeout`

  Description:

  The range for the parameter `TcpIpArpTableEntryTimeout` is restricted to 1..65535 seconds or Infinity. Infinity indicates that when an entry is created in the ARP table it will never be removed.

  Rationale:

  0 cannot be configured and corresponds to Infinity.

- Limitation on number of predefined, static, unicast assignments

  Description:

  The TcpIp can only handle a single unicast assignment with assignment method `TCPIP_STATIC` per EthIf controller.

  Rationale:

  This limitation allows a reduced code complexity. Concurrencies between multiple static assignments of described type must not be handled.

- Handling of illegal option length

  Description:

  If the TcpIp encounters a TCP segment with an illegal option length it will drop the segment but will not transmit a reset as suggested in http://tools.ietf.org/html/rfc1122, chapter 4.2.2.5.

- TCP Quiet Time Concept

  Description:

If the TcpIp crashes, it will not delay emitting any TCP segments for at least the agreed Maximum Segment Lifetime (MSL) as suggested in http://tools.ietf.org/html/rfc793, chapter 3.3 "TCP Quiet Time Concept".

- Precedence and Security

  Description:

  The Tcp does not evaluate the Precedence and Security of receiving TCP segments and does not include the options in the IP header in any TCP segments.

- IP Identification

  Description:

  If a retransmitted TCP segment is identical to the original packet, the TCP uses a different IP Identification field.

- DHCP options

  Description:

  The DHCP client does not support the following DHCP Options and BOOTP Vendor Extensions:

  - Time Offset
  - Time Server Option
  - Name Server Option
  - Domain Name Server Option
  - Log Server Option
  - Cookie Server Option
  - LPR Server Option
  - Impress Server Option
  - Resource Location Server Option
  - Host Name Option
  - Boot File Size Option
  - Merit Dump File
  - Domain Name
  - Swap Server
  - Root Path
  - Extensions Path
  - IP Forwarding Enable/Disable Option

- Non-Local Source Routing Enable/Disable Option
- Policy Filter Option
- Maximum Datagram Reassembly Size
- Default IP Time-to-live
- Path MTU Aging Timeout Option
- Path MTU Plateau Table Option
- Interface MTU Option
- All Subnets are Local Option
- Broadcast Address Option
- Perform Mask Discovery Option
- Mask Supplier Option
- Perform Router Discovery Option
- Router Solicitation Address Option
- Static Route Option
- Trailer Encapsulation Option
- ARP Cache Timeout Option
- Ethernet Encapsulation Option
- TCP Default TTL Option
- TCP Keepalive Interval Option
- TCP Keepalive Garbage Option
- Network Information Service Domain Option
- Network Information Servers Option
- Network Time Protocol Servers Option
- Vendor Specific Information
- NetBIOS over TCP/IP Name Server Option
- NetBIOS over TCP/IP Datagram Distribution Server Option
- NetBIOS over TCP/IP Node Type Option
- NetBIOS over TCP/IP Scope Option
- X Window System Font Server Option
- X Window System Display Manager Option
- Message
- Maximum DHCP Message Size

- Class-identifier
- Client-identifier

- DHCPINFORM messages

  Description:

  The DHCP client does not support the DHCPINFORM message. The DHCP client does not inform a DHCP server when obtaining an IP address through other means (e.g. manual configuration).

  RFC 2131 describes the optional mechanism of the DHCPINFORM message. (see chapter 3.4.). This DHCP client implementation does not support this option.

- DHCPRELEASE messages

  Description:

  The DHCP client does not support the DHCPRELEASE message. The DHCP client does not inform a DHCP server that an IP address is no longer used.

  RFC 2131 describes the optional mechanism of the DHCPRELEASE message. (see chapter 3.1.). This DHCP client implementation does not support this option.

- Reusing previously allocated network addresses

  Description:

  The DHCP client does not support to reuse a previously allocated network address to omit some of the steps for obtaining a network address. If the DHCP client wants to obtain a network address it always starts with sending a DHCPDISCOVER message.

  RFC 2131 describes the optional mechanism for reusing a previously allocated network addresses. (see chapter 3.2.). This DHCP client implementation does not support this option.

- The DHCP client continues using the previous network address

  Description:

  The DHCP client silently discards a DHCPACK message with a different acknowledged network address than the IP address in the preceding DHCP Request. If the DHCP client wants to obtain a network address it always starts with sending a DHCPDISCOVER message.

- Simple DHCPDISCOVER message transmission

  Description:

  The simple DHCP client will transmit a DHCPDISCOVER message to the MAC and IP broadcast.

- DAD duplicate address reinitialization

Description:

When static or link local ipv6 address is detected as duplicate during Duplicate address detection when TcpIpNdpSlaacOptimisticDadEnabled is enabled, it can only be reassigned by reinitialization of the TcpIp.

- Parsing of IKE and IpSec proposals

  Description:

  The received list of proposals in IKE_SA_INIT requests and IKE_AUTH requests is parsed sequentially until a match with the locally configured proposal of highest priority is found. Proposals that come after the match are not parsed and thereby any malformatted proposal that might come after the match is ignored.

- Supported PRF transform

  Description:

  The PRF transforms that can be negotiated are PRF_HMAC_SHA2_256 and PRF_HMAC_SHA2_384

- Mulitple UDP socket binds on same local address id and local port

  Description:

  If a UDP socket is bound to a Unicast local address A and a local port B, a second UDP socket cannot be bound to the same local address A and local port B.

  If a UDP socket is bound to a Multicast address A and a local port B it is possible to bind multiple UDP sockets to the same Multicast address A and local port B.

  If a UDP socket is bound to a controller ANY address A and a local port B it is possible to bind multiple UDP socket to the same controller ANY address A and local port B. However if additional UDP sockets are bound to Unicast local address C or controller ANY address A and a local port B, only one of the sockets can receive messages addressed to the Unicast local address C. The same applies when multiple UDP sockets are bound to TCPIP_LOCALADDRID_ANY

- Range restriction on configuration parameter `TcpIpTcpKeepAliveProbesMax`

  Description:

  The range for the parameter `TcpIpTcpKeepAliveProbesMax` is restricted to 0..255 keep alive probes.

- Handling of IKE Reauthentication

  Description:

  EAP and Configuration payloads not supported by the current implementation, because of this there is no difference between reauthentication and a new IKE exchange. Reauthentication can be started by either initiator or responder.

- Discarding DhcpV4 messages because of the lease time

Description:

If received DhcpV4 lease time is greater than 0xFFFFFFFF/7 (0x24924924) seconds, the message shall be silently discarded.

- Deleting Established IKE SA without transmitting a DELETE request

  Description:

  If address using IKE is abandoned due to being a duplicate, Ike SA will be deleted without sending a DELETE request.

  If address using IKE is released, Ike SA will be deleted without sending a DELETE request.

  If controller of the address using IKE is requested OFFLINE, Ike SA will be deleted without sending a DELETE request.

- TLS secured sockets close after TLS handshake

  Description:

  If a socket using TLS requested to close via TcpIp_Close without aborting, TcpIp will first notify TLS and wait for its approval.

2.6. Open-source software

TcpIp does not use open-source software.