Document Title	Specification of Chinese Vehicle-2-X Network
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	991

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R24-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2024-11-27	R24-11	AUTOSAR Release Management	• Completion of IEEE1722 specified tunneling process within the AUTOSAR communication stack for legacy communication (CAN and LIN)
2023-11-23	R23-11	AUTOSAR Release Management	• No content changes
2022-11-24	R22-11	AUTOSAR Release Management	• Initial release

Disclaimer

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

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

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

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

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Contents

Known Limitations

1 Introduction and functional overview

This document specifies the functionality, APIs and the configuration of the AUTOSAR Basic Software module Chinese Vehicle-2-X Network (CnV2xNet).

The Chinese Vehicle-2-X Network (CnV2xNet) together with the Chinese Vehicle-2-X Message (CnV2xMsg), Chinese Vehicle-2-X Management (CnV2xMgt), Chinese Vehicle-2-X Security (CnV2xSec), Linklayer Sdu Routing Module (LSduR) , and AUTOSAR BSW module Ethernet Interface (EthIf) forms the Chinese V2X stack within the AUTOSAR architecture.

The bases for this document are the Chinese LTE-V2X based standards [1] [2]. It is assumed that the reader is familiar with these standards.

1.1 Architecture Overview

CnV2xNet module provides services to upper module CnV2xMsg to transmit or receive V2X messages (i.e. DSMP SDUs), and gets services from the EthIf module to realize the data exchanging with LTE-V2X hardware. It also responsible for source Layer-2 address and destination layer-2 address selection and maintain QoS related mapping relationships between upper layer and lower layer from protocol perspective, the details are explained in chapter 1.2 and chapter 7 of this document.

Positioning of the CnV2xNet module within the AUTOSAR BSW and the Layered Software architecture is shown in below.

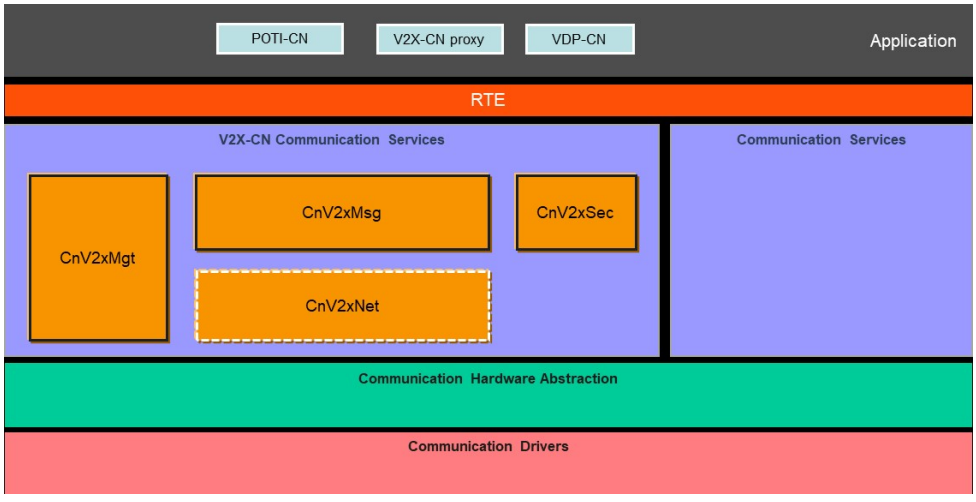

Figure 1.1: AUTOSAR BSW software architecture - CnV2xNet scope

1.2 Functional Overview

The functionality of CnV2xNet module should comply with the standard of Chinese LTE-V2X based Network protocol [2] and Technical Requirements of Vehicular Communication System based on LTE-V2X Direct Communication [1]. The module provides services to the upper CnV2xMsg module specified in [3] for data transmission and reception. In order to communicate with LTE-V2X hardware to realize packet transport services, it relies on the lower EthIf module [4].

From protocol perspective, CnV2xNet module includes DSMP sub-layer and Adaptation sub-layer. Adaptation sub-layer provides parameter adaptation function between LTE-V2X access layer and DSMP sub-layer. DSMP sub-layer is responsible for exchanging data with message layer and Adaptation layer.

CnV2xNet module function also includes: mapping between application layer identifier and destination layer-2 ID, generation/change/maintenance of the source layer-2 ID, mapping between the unicast/multicast address and the layer-2 ID, mapping between the message priority and PPPP, indicating the service period to the lower layer, indicating the channel busy rate or maximum data rate to the upper layer, and so on.

2 Acronyms and Abbreviations

Abbreviation / Acronym:	Description:
API	Application programming Interface
BSW	Basic Software
BSM	Basic safety Message
C-V2X	Cellular based Vehicle to Everything
CCSA	China Communications Standards Association
CnV2xMsg	Chinese Vehicle-2-X Message
CnV2xNet	Chinese Vehicle-2-X Network
CnV2xSec	Chinese Vehicle-2-X Security
DEM	Diagnostic Event Manager
DET	Default Error Tracer
DSMP	Dedicated Short Message Protocol
EcuM	Electronic Control Unit Manager
IF	Interface
NTCAS	National Technical Committee of Auto Standardization
PPPP	ProSe Per-Packet Priority

3 Related documentation

3.1 Input documents & related standards and norms

[1] GB/T: Technical requirements and test methods of vehicular communication system based on LTE-V2X direct communication (Draft Edition: 2022-04-01)
http://www.catarc.org.cn/

[2] YD/T 3707-2020: Technical requirements of network layer of LTE-based vehicular communication
http://www.ccsa.org.cn/

[3] Specification of Chinese Vehicle-2-X Message
AUTOSAR_CP_SWS_ChineseV2XMessage

[4] Specification of Ethernet Interface
AUTOSAR_CP_SWS_EthernetInterface

[5] Specification of Default Error Tracer
AUTOSAR_CP_SWS_DefaultErrorTracer

[6] Specification of ECU State Manager
AUTOSAR_CP_SWS_ECUStateManager

[7] Specification of Linklayer Sdu Routing Module
AUTOSAR_CP_SWS_LSduRouter

4 Constraints and assumptions

4.1 Limitations

Wireless communication supports LTE-V2X only. Other cellular based wireless communication can be extended in future release of AUTOSAR standard. CnV2xNet module support non-IP (i.e. DSMP) transmission only and mainly focus on broadcast based packet transport services in R20-11. The V2X modules follow the guidance regarding the Day-1 V2X allocations defined in [1] [2], which are by NTCAS and CCSA respectively.

4.2 Applicability to car domains

This specification is applicable to all car domains.

5 Dependencies to other modules

This section describes the relations of CnV2xNet module to other modules within the AUTOSAR basic software architecture. It outlines the modules that are required or optional for the realization of CnV2xNet module and services.

5.1 AUTOSAR Default Error Tracer (DET)

In development mode, CnV2xNet module reports errors through the Det_ReportError function of DET Module [5].

5.2 AUTOSAR Ecu State Manager (EcuM)

The EcuM [6] initializes the CnV2xNet module by calling CnV2xNet_Init specified in 8.3.1 in this document.

5.3 AUTOSAR Ethernet Interface (EthIf)

The Ethernet Interface [4] is the lower layer module of the CnV2xNet module for the control flow.

5.4 AUTOSAR Linklayer Sdu Routing Module (LSduR)

The Linklayer Sdu Routing Module [7] is the lower layer module of the CnV2xNet module for the data flow.

5.5 AUTOSAR Chinese Vehicle-2-X Message (CnV2xMsg)

CnV2xMsg is the upper layer module of CnV2xNet module. The callback services called by CnV2xNet module are declared and placed inside the CnV2xMsg module. These callbacks provide receive indication and transmit confirmation services for the CnV2xMsg Module.

6 Requirements Tracing

Requirement	Description	Satisfied by
[CP_SRS_CnV2X_-00401]	The network layer of Chinese V2X communication shall support a CCSA compliant Network layer protocol of LTE-based vehicular communication	[CP_SWS_CnV2xNet_00100] [CP_SWS_CnV2xNet_00101] [CP_SWS_CnV2xNet_00102] [CP_SWS_CnV2xNet_00103] [CP_SWS_CnV2xNet_00105] [CP_SWS_CnV2xNet_00106] [CP_SWS_CnV2xNet_00109] [CP_SWS_CnV2xNet_00110] [CP_SWS_CnV2xNet_00111] [CP_SWS_CnV2xNet_00113] [CP_SWS_CnV2xNet_00114] [CP_SWS_CnV2xNet_00115] [CP_SWS_CnV2xNet_00116] [CP_SWS_CnV2xNet_01002] [CP_SWS_CnV2xNet_01003] [CP_SWS_CnV2xNet_01004] [CP_SWS_CnV2xNet_01006] [CP_SWS_CnV2xNet_01007] [CP_SWS_CnV2xNet_01008] [CP_SWS_CnV2xNet_01009] [CP_SWS_CnV2xNet_01010] [CP_SWS_CnV2xNet_01011] [CP_SWS_CnV2xNet_01012] [CP_SWS_CnV2xNet_02001] [CP_SWS_CnV2xNet_02003] [CP_SWS_CnV2xNet_02005] [CP_SWS_CnV2xNet_02009] [CP_SWS_CnV2xNet_02012] [CP_SWS_CnV2xNet_02015] [CP_SWS_CnV2xNet_02020]
[CP_SRS_CnV2X_-00402]	The network layer of Chinese V2X communication shall Select and maintain Source Layer-2 ID and Destination Layer-2 ID	[CP_SWS_CnV2xNet_00119] [CP_SWS_CnV2xNet_00120] [CP_SWS_CnV2xNet_00121]
[CP_SRS_CnV2X_-00403]	The network layer of Chinese V2X communication shall provide the mapping between packet priority and PPPP	[CP_SWS_CnV2xNet_00108] [CP_SWS_CnV2xNet_00117]
[CP_SRS_CnV2X_-00404]	The network layer of Chinese V2X communication shall provide CBR or Max data rate to message Layer	[CP_SWS_CnV2xNet_00118]
[CP_SRS_CnV2X_-00605]	The Chinese V2X communication shall randomize the identifiers related to BSM to in order to support privacy	[CP_SWS_CnV2xNet_00112]
[CP_SRS_CnV2X_-00606]	The Chinese V2X communication shall change pseudonym certificates in order to support privacy	[CP_SWS_CnV2xNet_00122] [CP_SWS_CnV2xNet_00123] [CP_SWS_CnV2xNet_00124] [CP_SWS_CnV2xNet_00125]
[SRS_BSW_00345]	BSW Modules shall support pre-compile configuration	[SWS_CnV2xNet_03001]

Table 6.1: Requirements Tracing

7 Functional specification

7.1 General Functionality

[CP_SWS_CnV2xNet_00100]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈The CnV2xNet module shall implement the network Layer protocols defined in [2] unless specified otherwise in this document.⌋

[CP_SWS_CnV2xNet_00101]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈The network protocol shall meet the network layer related requirements defined in [1].⌋

[CP_SWS_CnV2xNet_00102]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈The CnV2xNet Module shall encapsulate the payload from the CnV2xMsg module with a DSMP header and Adaptation layer header as per [2].⌋

7.2 Message Transmission

[CP_SWS_CnV2xNet_00103]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈The CnV2xNet module shall provide the API CnV2xNet_Transmit () to enable transmit requests from the CnV2xMsg module.⌋

[CP_SWS_CnV2xNet_00105]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈The CnV2xNet module shall only support DSMP based service to CnV2xMsg Module, and employ DSMP and Adaptation layer headers.⌋

[CP_SWS_CnV2xNet_00106]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈The DSMP protocol version shall be set to zero.⌋

[CP_SWS_CnV2xNet_00108]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00403

⌈The CnV2xNet module shall select a PPPP value for a V2X packet to be transmitted based on Priority indicated in CnV2xNet_Transmit () and the mapping relationship defined in Table A.2 of [2].⌋

[CP_SWS_CnV2xNet_00109]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈The CnV2xNet module shall provide traffic period indicated in CnV2xNet_Transmit () (if provided by CnV2xMsg Module) to lower layer using EthIf_SetBufCv2xPC5TxParams().⌋

[CP_SWS_CnV2xNet_00110]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈The CnV2xNet module shall provide transmission parameters to the C-V2X Driver for a V2X Packet to be transmitted via an API call to EthIf_SetBufCv2xPC5TxParams. This has to be done during the CnV2xNet_Transmit context.⌋

[CP_SWS_CnV2xNet_00111]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈

The CnV2xNet module shall transmit packets using the LSduR_V2xGnTransmit () API provided by the LSduR Module. This has to be done during the CnV2xNet_Transmit context.

⌋

[CP_SWS_CnV2xNet_00112]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00605

⌈The CnV2xNet module shall suspend transmission of V2X packet when a pseudonym certificate changes is in preparation.⌋

[CP_SWS_CnV2xNet_00113]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈If the configuration parameter CnV2xNet_TxConfirmation is enabled, the CnV2xNet module shall provide information about the status of the transmission with an associated ID (generated by the CnV2xMsg module and handed down to track the status of the packet) the CnV2xMsg Module via the CnV2xMsg_TxConfirmation () (for details see chapter 8.4.1 in [3])callback.⌋

NOTE: A dedicated EtherType value can be considered for Chinese V2X network layer during in-vehicle communication. It is up to implementation for whether a private value or a registered value is used.

7.3 Message Reception

[CP_SWS_CnV2xNet_00114]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈The CnV2xNet module shall create a unique TransactionId for each received packet. This TransactionId is handed up to track the received packets and is used for verification.⌋

[CP_SWS_CnV2xNet_00115]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈The CnV2xNet module shall indicate received packets via the CnV2xMsg_RxIndication () callback to the CnV2xMsg module.⌋

[CP_SWS_CnV2xNet_00116]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈The CnV2xNet module shall get the reception status of a received packet during the CnV2xNet_RxIndication () from the EthIf module with a call to EthIf_GetBufCv2xPC5RxParams ().⌋

[CP_SWS_CnV2xNet_00117]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00403

⌈The CnV2xNet module shall map PPPP indicated in CnV2xNet_RxIndication () to priority and provides to upper layer through CnV2xMsg_RxIndication ().⌋

[CP_SWS_CnV2xNet_00118]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00404

⌈The CnV2xNet module shall provide the received CBR or max data rate indicated in CnV2xNet_RxIndication () to upper layer through CnV2xMsg_RxIndication ().⌋

7.4 Layer-2 ID selection

[CP_SWS_CnV2xNet_00119]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00402

⌈The CnV2xNet module shall implement the source Layer-2 ID and destination Layer-2 ID selection and maintenance.⌋

[CP_SWS_CnV2xNet_00120]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00402

⌈The CnV2xNet module shall randomly select a source layer-2 ID within [0x010001, 0xFFFFFE] during initialization as per [1]. It shall randomly reselect the source layer-2 ID during the application layer ID changing context.⌋

[CP_SWS_CnV2xNet_00121]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00402

⌈The CnV2xNet module shall select a destination layer-2 ID for a V2X packet to be transmitted based on Aid indicated in CnV2xNet_Transmit () and the mapping relationship defined in Table 3 of [1].⌋

[CP_SWS_CnV2xNet_00122]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00606

⌈The CnV2xNet shall perform the changing of the source Layer-2 ID in two phases, which consist of the prepare phase and the commit/abort phase. The second phase depends on the result of all called modules within the first phase. If the first phase was successful, the commit phase shall be performed. Otherwise, the abort phase shall be triggered.⌋

[CP_SWS_CnV2xNet_00123]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00606

⌈In the prepare phase, the API CnV2xNet_PrepareAppLayerIdChange() shall be called by CnV2xMsg.

⌋

[CP_SWS_CnV2xNet_00124]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00606

⌈In the commit phase, the API CnV2xNet_CommitAppLayerIdChange() shall be called by CnV2xMsg.

⌋

[CP_SWS_CnV2xNet_00125]
Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00606

⌈In the commit phase, the API CnV2xNet_CommitAppLayerIdChange() shall be called by CnV2xMsg.

⌋

7.5 Error Classification

This chapter lists and classifies all errors that can be detected within this software module. Each error is classified according to relevance (development / production) and related error code. For development errors, a value is defined.

7.5.1 Development Errors

[CP_SWS_CnV2xNet_00126] Development Error Types

Status: DRAFT

⌈The following table lists development errors that shall be distinguished by the CnV2xNet module. CnV2xNet shall report them to the DET, if development error detection (CnV2xNetDevErrorDetect) is enabled⌋

Type of error	*Related error code*	*Value [hex]*
API service called with invalid parameter	CNV2XNET_E_PARAM	0x01
API service called with invalid pointer	CNV2XNET_E_PARAM_POINTER	0x02
API function called before the CnV2xNet module has been fully initialized	CNV2XNET_E_UNINIT	0x03
CnV2xNet initialization failed	CNV2XNET_E_INIT_FAILED	0x04

Table 7.1: Development Error Types for CnV2xNet

7.5.2 Runtime Errors

There are no runtime errors.

7.5.3 Transient Faults

There are no transient faults.

7.5.4 Production Errors

There are no production errors.

7.5.5 Extended Production Errors

There are no extended production errors.

8 API specification

8.1 Imported types

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

[CP_SWS_CnV2xNet_01001] Definition of imported datatypes of module CnV2x Net ⌈

Module	*Header File*	*Imported Type*
CnV2xMsg	CnV2xMsg.h	CnV2xMsg_RxParamsPresenceType (draft)
	CnV2xMsg.h	CnV2xMsg_RxParamsType (draft)
Comtype	ComStack_Types.h	PduIdType
	ComStack_Types.h	PduInfoType
	ComStack_Types.h	PduLengthType
CV2x	CV2x_GeneralTypes.h	CV2x_BufCV2xPC5RxParamIdType (draft)
	CV2x_GeneralTypes.h	CV2x_BufCV2xPC5TxParamIdType (draft)
Std	Std_Types.h	Std_ReturnType
	Std_Types.h	Std_VersionInfoType

⌋

8.2 Type definitions

8.2.1 CnV2xNet_TxParamsType

[CP_SWS_CnV2xNet_01002] Definition of datatype CnV2xNet_TxParamsType

Status: DRAFT

Upstream requirements: CP_SRS_CnV2X_00401

⌈

Name	CnV2xNet_TxParamsType (draft)	
Kind	Structure	
Elements	presence	
	Type	CnV2xNet_TxParamsPresenceType
	Comment	Mark optional child present or not
	Aid	
	Type	uint64
	Comment	The value of the AID (Application Identifier)
	ProtocolType	
	Type	CnV2x_NetworkProtocolType

▽

△

	Comment	Network layer protocol type. Value 4 is used for DSMP protocol and other values are reserved.
	priority	
	Type	uint8
	Comment	Priority of V2X-CN message
	SourceLayer2Id	
	Type	CnV2x_Layer2IdType
	Comment	Source Layer 2 ID of V2X-CN packet(24-bit)
	DestinationLayer2Id	
	Type	CnV2x_Layer2IdType
	Comment	Destination Layer-2 ID of V2X-CN packet(24-bit)
	TrafficPeriod	
	Type	CnV2x_TrafficPeriodType
	Comment	Indicate Traffic Period
	AppLayerIdChangedCount16	
	Type	uint16
	Comment	The order of the Application layer Id Changed. This value is created in the CnV2xMsg module and shall be mapped with parameter pseudonymCount16.
	DsmpHeaderExtensionPtr	
	Type	uint8*
	Comment	Ptr of Dsmp header Extension
	DsmpHeaderExtensionLength	
	Type	uint16
	Comment	Length of Dsmp header Extension
Description	Wraps Message layer parameters from CnV2xMsg **Tags:** atp.Status=draft	
Available via	CnV2xNet.h	

⌋

8.2.2 CnV2xNet_TxParamsPresenceType

[CP_SWS_CnV2xNet_01003] Definition of datatype CnV2xNet_TxParamsPresenceType

Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈

Name	CnV2xNet_TxParamsPresenceType (draft)
Kind	Bitfield
Derived from	uint8

▽

△

Elements	Kind	Name	Mask	Description
	bit	SourceLayer2Id	0x08	Bit 3: Optional child present
	bit	DestinationLayer2Id	0x04	Bit 2: Optional child present
	bit	TrafficPeriod	0x02	Bit 1: Optional child present
	bit	DsmpHeaderExtension	0x01	Bit 0 (LSB): Optional child(Dsmp HeaderExtensionPtr and Dsmp HeaderExtensionLength) present
Description	Presence flags for CnV2xNet_TxParamsType **Tags:** atp.Status=draft			
Available via	CnV2x_GeneralTypes.h			

⌋

8.2.3 CnV2xNet_RxParamsType

[CP_SWS_CnV2xNet_01004] Definition of datatype CnV2xNet_RxParamsType

Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈

Name	CnV2xNet_RxParamsType (draft)	
Kind	Structure	
Elements	presence	
	Type	CnV2xNet_RxParamsPresenceType
	Comment	Mark optional child present or not
	Sourcelayer2Id	
	Type	CnV2x_Layer2IdType
	Comment	Source Layer 2 ID of V2X-CN packet(24bit)
	DestinationLayer2Id	
	Type	CnV2x_Layer2IdType
	Comment	Destination Layer 2 ID of V2X-CN packet(24bit)
	pppp	
	Type	CnV2x_PPPPType
	Comment	ProSe per-packet priority
	cbr	
	Type	CnV2x_CbrType
	Comment	Channel busy rate
	MaxDataRate	
	Type	CnV2x_MaxDataRateType
	Comment	Max data rate

▽

△

Description	Structure containing Access layer parameters related to a received C-V2X packet. **Tags:** atp.Status=draft
Available via	CnV2xNet.h

⌋

8.2.4 CnV2xNet_RxParamsPresenceType

[CP_SWS_CnV2xNet_01005] Definition of datatype CnV2xNet_RxParamsPresenceType

Status: DRAFT

⌈

Name	CnV2xNet_RxParamsPresenceType (draft)			
Kind	Bitfield			
Derived from	uint8			
Elements	***Kind***	***Name***	***Mask***	***Description***
	bit	Cbr	0x02	Bit 1: Optional child present
	bit	maxDataRate	0x01	Bit 0 (LSB): Optional child present
Description	Presence flags for CnV2xNet_RxParamsType **Tags:** atp.Status=draft			
Available via	CnV2x_GeneralTypes.h			

⌋

8.2.5 CnV2x_Layer2IdType

[CP_SWS_CnV2xNet_01006] Definition of ImplementationDataType CnV2x_Layer2IdType

Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈

Name	CnV2x_Layer2IdType (draft)		
Kind	Type		
Derived from	uint32		
Range	0..16777215	–	–

▽

△

Description	The Layer 2 Id (24bit) **Tags:** atp.Status=draft
Variation	–
Available via	CnV2x_GeneralTypes.h

⌋

8.2.6 CnV2x_PPPPType

[CP_SWS_CnV2xNet_01007] Definition of datatype CnV2x_PPPPType

Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈

Name	CnV2x_PPPPType (draft)		
Kind	Type		
Derived from	uint8		
Range	1..8	–	–
Description	Prose per-packet priority of V2X message **Tags:** atp.Status=draft		
Available via	CnV2x_GeneralTypes.h		

⌋

8.2.7 CnV2x_NetworkProtocolType

[CP_SWS_CnV2xNet_01008] Definition of datatype CnV2x_NetworkProtocolType

Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈

Name	CnV2x_NetworkProtocolType (draft)		
Kind	Type		
Derived from	uint8		
Range	CNV2X_DSMP_PROTOCOL	0x04	DSMP protocol type
Description	Enumeration Type as defined in CCSA YD/T 3709-2020. **Tags:** atp.Status=draft		
Available via	CnV2x_GeneralTypes.h		

⌋

8.2.8 CnV2x_TrafficPeriodType

[CP_SWS_CnV2xNet_01009] Definition of datatype CnV2x_TrafficPeriodType

Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈

Name	CnV2x_TrafficPeriodType (draft)		
Kind	Type		
Derived from	uint8		
Range	CNV2X_TRAFFIC_PERIOD_20	0x00	Traffic Period: 20ms
	CNV2X_TRAFFIC_PERIOD_50	0x01	Traffic Period: 50ms
	CNV2X_TRAFFIC_PERIOD_100	0x02	Traffic Period: 100ms
	CNV2X_TRAFFIC_PERIOD_200	0x03	Traffic Period: 200ms
	CNV2X_TRAFFIC_PERIOD_300	0x04	Traffic Period: 300ms
	CNV2X_TRAFFIC_PERIOD_400	0x05	Traffic Period: 400ms
	CNV2X_TRAFFIC_PERIOD_500	0x06	Traffic Period: 500ms
	CNV2X_TRAFFIC_PERIOD_600	0x07	Traffic Period: 600ms
	CNV2X_TRAFFIC_PERIOD_700	0x08	Traffic Period: 700ms
	CNV2X_TRAFFIC_PERIOD_800	0x09	Traffic Period: 800ms
	CNV2X_TRAFFIC_PERIOD_900	0x0a	Traffic Period: 900ms
	CNV2X_TRAFFIC_PERIOD_1000	0x0b	Traffic Period: 1000ms
Description	Enumeration Type as defined in CCSA YD/T 3709-2020. **Tags:** atp.Status=draft		
Available via	CnV2x_GeneralTypes.h		

⌋

8.2.9 CnV2x_CbrType

[CP_SWS_CnV2xNet_01010] Definition of ImplementationDataType CnV2x_Cbr Type

Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈

Name	CnV2x_CbrType (draft)		
Kind	Type		
Derived from	uint8		
Range	0..100	–	–
Description	Channel busy rate % **Tags:** atp.Status=draft		
Variation	–		
Available via	CnV2x_GeneralTypes.h		

⌋

8.2.10 CnV2x_MaxDataRateType

[CP_SWS_CnV2xNet_01011] Definition of ImplementationDataType CnV2x_Max DataRateType

Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈

Name	CnV2x_MaxDataRateType (draft)		
Kind	Type		
Derived from	uint32		
Range	0..1585200	–	–
Description	Max date Rate uint: bps **Tags:** atp.Status=draft		
Variation	–		
Available via	CnV2x_GeneralTypes.h		

⌋

8.2.11 CnV2x_NetTxResultType

[CP_SWS_CnV2xNet_01012] Definition of datatype CnV2x_NetTxResultType

Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈

Name	CnV2x_NetTxResultType (draft)		
Kind	Type		
Derived from	uint8		
Range	CNV2X_NETTX_ ACCEPTED	0x00	–
	CNV2X_NETTX_E_ MAXSDUSIZEOVF	0x01	Traffic Period: 50ms
	CNV2X_NETTX_E_AID	0x02	transmit has been rejected due to unsupported AID
	CNV2X_NETTX_E_ PRTOCOLTYPE	0x03	transmit has been rejected due to maximum length exceedance
	CNV2X_NETTX_E_ PRIORITY	0x04	transmit has been rejected due to unsupported priority
	CNV2X_NETTX_E_ LAYER2ID_S	0x05	transmit has been rejected due to uncorrected source Layer 2 ID
	CNV2X_NETTX_E_ LAYER2ID_D	0x06	transmit has been rejected due to uncorrected destination Layer 2 ID
	CNV2X_NETTX_E_TP	0x07	transmit has been rejected due to unsupported traffic period
	CNV2X_NETTX_E_ UNSPECIFIED	0x08	transmit has been rejected due to unspecified reasons
Description	Return Types of API CnV2xNet_Transmit. **Tags:** atp.Status=draft		
Available via	CnV2x_GeneralTypes.h		

⌋

8.3 Function definition

8.3.1 CnV2xNet_Init

[CP_SWS_CnV2xNet_02001] Definition of API function CnV2xNet_Init

Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈

Service Name	CnV2xNet_Init (draft)	
Syntax	`void CnV2xNet_Init ( void* CfgPtr )`	
Service ID [hex]	0x1	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	CfgPtr	Points to a null pointer
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Initialize the CnV2xNet module **Tags:** atp.Status=draft	
Available via	CnV2xNet.h	

⌋

[CP_SWS_CnV2xNet_02030]

Status: DRAFT

⌈The CfgPtr must be a Null pointer and as a rationale, that initialization at runtime (post build init) is not possible.⌋

[CP_SWS_CnV2xNet_02002]

Status: DRAFT

⌈If development error detection is enabled: The function shall check the parameter CfgPtr for containing a valid configuration. If the check fails, the function shall raise the development error CNV2XNET_E_INIT_FAILED.⌋

8.3.2 CnV2xNet_GetVersionInfo

[CP_SWS_CnV2xNet_02003] Definition of API function CnV2xNet_GetVersion Info

Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈

Service Name	CnV2xNet_GetVersionInfo (draft)	
Syntax	`void CnV2xNet_GetVersionInfo ( Std_VersionInfoType* VersionInfoPtr )`	
Service ID [hex]	0x2	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	VersionInfoPtr	Pointer to where to store the version information of this module.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Returns the version information of this module. **Tags:** atp.Status=draft	
Available via	CnV2xNet.h	

⌋

[CP_SWS_CnV2xNet_02004]

Status: DRAFT

⌈If development error detection is enabled: the function shall check the parameter VersionInfoPtr for being valid. If the check fails, the function shall raise the development error CNV2XNET_E_PARAM_POINTER.⌋

8.3.3 CnV2xNet_Transmit

[CP_SWS_CnV2xNet_02005] Definition of API function CnV2xNet_Transmit

Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈

Service Name	CnV2xNet_Transmit (draft)	
Syntax	`CnV2x_NetTxResultType CnV2xNet_Transmit ( uint16 TransactionId16, const CnV2xNet_TxParamsType* TxParams, uint16 Length, const uint8* DataPtr )`	
Service ID [hex]	0x3	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	TransactionId16	ID identifying the payload to be transmitted. This ID is generated by the CnV2xMsg module and is used later to indicate the status of the transmission of the message having this ID to the CnV2x Msg module.
	TxParams	Structure containing all the Network layer parameters used for the transmit request.
	Length	Length of the data pointed by DataPtr.
	DataPtr	Payload of the Message Layer packet to be transmitted
Parameters (inout)	None	
Parameters (out)	None	
Return value	CnV2x_NetTxResultType	`CNV2X_NETTX_ACCEPTED` if no error occurred. `CNV2X_NETTX_E_MAXSDUSIZEOVF` transmit has been rejected due to maximum length exceedance `CNV2X_NETTX_E_AID` transmit has been rejected due to unsupported AID `CNV2X_NETTX_E_PRTOCOLTYPE` transmit has been rejected due to unsupported protocol type `CNV2X_NETTX_E_PRIORITY` transmit has been rejected due to unsupported priority `CNV2X_NETTX_E_LAYER2ID_S` transmit has been rejected due to uncorrected source Layer 2 ID `CNV2X_NETTX_E_LAYER2ID_D` transmit has been rejected due to uncorrected destination Layer 2 ID `CNV2X_NETTX_E_TP` transmit has been rejected due to unsupported traffic period `CNV2X_NETTX_E_UNSPECIFIED` transmit has been rejected due to unspecified reasons
Description	This API is called by the CvxMsgCN module to request sending a Network Layer V2X PDU to the peer Network entity. **Tags:** atp.Status=draft	
Available via	CnV2xNet.h	

⌋

[CP_SWS_CnV2xNet_02006]

Status: DRAFT

⌈If development error detection is enabled: the function shall check that the service CnV2xNet_Init was previously called. If the check fails, the function shall raise

the development error CNV2XNET_E_UNINIT otherwise (if DET is disabled) return E_NOT_OK.⌋

[CP_SWS_CnV2xNet_02007]

Status: DRAFT

⌈If development error detection is enabled: the function shall check the parameter TxParams for being valid. If the check fails, the function shall raise the development error CNV2XNET_E_PARAM_POINTER otherwise (if DET is disabled) return E_NOT_OK.⌋

[CP_SWS_CnV2xNet_02008]

Status: DRAFT

⌈If development error detection is enabled: the function shall check the parameter DataPtr for being valid. If the check fails, the function shall raise the development error CNV2XNET_E_PARAM_POINTER otherwise (if DET is disabled) return E_NOT_OK.⌋

8.3.4 CnV2xNet_PrepareAppLayerIdChange

[CP_SWS_CnV2xNet_02009] Definition of API function CnV2xNet_PrepareAppLayerIdChange

Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈

Service Name	CnV2xNet_PrepareAppLayerIdChange (draft)	
Syntax	`Std_ReturnType CnV2xNet_PrepareAppLayerIdChange ( uint8 TransmissionClass, uint16 ApplayerIdChangedCount16 )`	
Service ID [hex]	0x4	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	TransmissionClass	Transmission Class Indication
	ApplayerIdChanged Count16	Order of the Application layer identifier changed correspond to specific message type. This count value is created in the CnV2x Sec module.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: operation successful `E_NOT_OK`: operation failed
Description	By this API primitive the CnV2xNet module gets an indication that Application Layer Id is about to change and hereby source Layer-2 ID is about to be changed. **Tags:** atp.Status=draft	
Available via	CnV2xNet.h	

⌋

[CP_SWS_CnV2xNet_02010]

Status: DRAFT

⌈The function shall prepare the setting of source Layer-2 ID used for packet transmission.⌋

[CP_SWS_CnV2xNet_02011]

Status: DRAFT

⌈If development error detection is enabled: the function shall check that the service CnV2xNet_Init was previously called. If the check fails, the function shall raise the development error CNV2XNET_E_UNINIT otherwise (if DET is disabled) return E_NOT_OK.⌋

8.3.5 CnV2xNet_CommitAppLayerIdChange

[CP_SWS_CnV2xNet_02012] Definition of API function CnV2xNet_CommitAppLayerIdChange

Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈

Service Name	CnV2xNet_CommitAppLayerIdChange (draft)	
Syntax	`Std_ReturnType CnV2xNet_CommitAppLayerIdChange ( uint8 TransmissionClass, uint16 ApplayerIdChangedCount16 )`	
Service ID [hex]	0x5	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	TransmissionClass	Transmission Class Indication
	ApplayerIdChanged Count16	Order of the Application layer identifier changed correspond to specific message type. This count value is created in the CnV2x Sec module.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: operation successful `E_NOT_OK`: operation failed
Description	The CnV2xMsg module calls this function when all modules are OK with the pseudonym certificate change and the change is to be committed. **Tags:** atp.Status=draft	
Available via	CnV2xNet.h	

⌋

[CP_SWS_CnV2xNet_02013]

Status: DRAFT

⌈The function shall update the new source Layer-2 ID.⌋

[CP_SWS_CnV2xNet_02014]

Status: DRAFT

⌈If development error detection is enabled: the function shall check that the service CnV2xNet_Init was previously called. If the check fails, the function shall raise the development error CNV2XNET_E_UNINIT otherwise (if DET is disabled) return E_NOT_OK.⌋

Note: The function requires previous preparation of the pseudonym certificate via an API call to CnV2xNet_PrepareAppLayerIdChange.

8.3.6 CnV2xNet_AbortAppLayerIdChange

[CP_SWS_CnV2xNet_02015] Definition of API function CnV2xNet_AbortApp LayerIdChange

Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈

Service Name	CnV2xNet_AbortAppLayerIdChange (draft)	
Syntax	`Std_ReturnType CnV2xNet_AbortAppLayerIdChange ( uint8 TransmissionClass, uint16 ApplayerIdChangedCount16 )`	
Service ID [hex]	0x6	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	TransmissionClass	Transmission Class Indication
	ApplayerIdChanged Count16	Order of the Application layer identifier changed correspond to specific message type. This count value is created in the CnV2x Sec module.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: operation successful `E_NOT_OK`: operation failed
Description	The CnV2xMsg module calls this function when not all modules are OK with the pseudonym certificate change and the change is to be rolled back. **Tags:** atp.Status=draft	
Available via	CnV2xNet.h	

⌋

[CP_SWS_CnV2xNet_02016]

Status: DRAFT

⌈The function shall roll back the prepared source Layer-2 ID change.⌋

[CP_SWS_CnV2xNet_02017]

Status: DRAFT

⌈If development error detection is enabled: the function shall check that the service CnV2xNet_Init was previously called. If the check fails, the function shall raise the development error CnV2xNet_E_UNINIT otherwise (if DET is disabled) return E_NOT_OK.⌋

Note: The function requires previous preparation of the pseudonym certificate via an API call to CnV2xNet_PrepareAppLayerIdChange.

8.4 Callback notifications

8.4.1 CnV2xNet_RxIndication

[CP_SWS_CnV2xNet_91000] Definition of callback function CnV2xNet_RxIndication

Status: DRAFT

⌈

Service Name	CnV2xNet_RxIndication (draft)	

Syntax	`void CnV2xNet_RxIndication (`
	`  PduIdType RxPduId,`
	`  const PduInfoType* PduInfoPtr`
	`)`	

Service ID [hex]	0x42	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in)	RxPduId	ID of the received PDU.
	PduInfoPtr	Contains the length (SduLength) of the received PDU, a pointer to a buffer (SduDataPtr) containing the PDU, and the MetaData related to this PDU.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Indication of a received PDU from a lower layer communication interface module. **Tags:** atp.Status=draft	
Available via	CnV2x.h	

⌋

[CP_SWS_CnV2xNet_02023]

Status: DRAFT

⌈The function CnV2xNet_RxIndication shall get reception parameters of the C-V2X Driver for a C-V2X Packet received via an API call to EthIf_GetBufCv2xPC5RxParams.⌋

[CP_SWS_CnV2xNet_02024]

Status: DRAFT

⌈If development error detection is enabled: the function shall check that the service CnV2xNet_Init was previously called. If the check fails, the function shall raise the development error CNV2XNET_E_UNINIT.⌋

[CP_SWS_CnV2xNet_02025]

Status: DRAFT

⌈If development error detection is enabled: the function shall check the parameter DataPtr for being valid. If the check fails, the function shall raise the development error CNV2XNET_E_PARAM_POINTER.

If development error detection is enabled: the function shall check the parameter PduInfoPtr for being valid. If the check fails, the function shall raise the development error CNV2XNET_E_PARAM_POINTER.

⌋

8.4.2 CnV2xNet_TxConfirmation

[CP_SWS_CnV2xNet_91001] Definition of callback function CnV2xNet_TxConfirmation

Status: DRAFT

⌈

Service Name	CnV2xNet_TxConfirmation (draft)	
Syntax	`void CnV2xNet_TxConfirmation ( PduIdType TxPduId, Std_ReturnType result )`	
Service ID [hex]	0x40	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in)	TxPduId	ID of the PDU that has been transmitted.
	result	E_OK: The PDU was transmitted. E_NOT_OK: Transmission of the PDU failed.

▽

△

Parameters (inout)	None
Parameters (out)	None
Return value	None
Description	The lower layer communication interface module confirms the transmission of a PDU, or the failure to transmit a PDU. **Tags:** atp.Status=draft
Available via	CnV2x.h

⌋

[CP_SWS_CnV2xNet_02026]

Status: DRAFT

⌈If development error detection is enabled: the function shall check that the service CnV2xNet_Init was previously called. If the check fails, the function shall raise the development error CNV2XNET_E_UNINIT.⌋

8.5 Scheduled functions

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

8.5.1 CnV2xNetMainFunction

[CP_SWS_CnV2xNet_02020] Definition of scheduled function CnV2xNet_MainFunction

Status: DRAFT
Upstream requirements: CP_SRS_CnV2X_00401

⌈

Service Name	CnV2xNet_MainFunction (draft)
Syntax	`void CnV2xNet_MainFunction ( void )`
Service ID [hex]	0x9
Description	Main function of the CnV2xNet module for periodical execution of protocol operations. **Tags:** atp.Status=draft
Available via	SchM_CnV2xNet.h

⌋

8.6 Expected interfaces

8.6.1 Mandatory interfaces

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

[CP_SWS_CnV2xNet_02021] Definition of mandatory interfaces required by module CnV2xNet ⌈

API Function	*Header File*	*Description*
CnV2xMsg_RxIndication (draft)	CnV2xMsg.h	By this API primitive the CnV2xMsg module gets a confirmation that the V2X message with a certain ID was send successfully. This API primitive is called by the CnV2xNet module providing the data and the Network parameters of a received DSMP packet to CnV2xMsg module. **Tags:** atp.Status=draft
EthIf_GetBufCV2xPC5RxParams	EthIf.h	Read out values related to the receive direction of the Cellular V2X for a received packet. For example, this could be CBR belonging to one single packet.
EthIf_GetBufCV2xPC5TxParams	EthIf.h	Read out values related to the transmit direction of the Cellular V2X for a transmitted packet. For example, this could be transaction ID belonging to one single packet.
EthIf_SetBufCV2xPC5TxParams	EthIf.h	Set values related to the transmit direction of the Cellular V2X for a specific buffer (packet to be sent). For example, this can be the desired ProSe per-packet priority belonging to one single packet.
LSduR_CnV2xNetTransmit (draft)	LSduR_CnV2xNet.h	Requests transmission of a PDU.

⌋

8.6.2 Optional interfaces

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

[CP_SWS_CnV2xNet_02022] Definition of optional interfaces requested by module CnV2xNet ⌈

API Function	*Header File*	*Description*
CnV2xMsg_TxConfirmation (draft)	CnV2xMsg.h	By this API primitive, the CnV2xMsg module gets a confirmation that the V2X message with a certain ID was send successfully. **Tags:** atp.Status=draft
Det_ReportError	Det.h	Service to report development errors.

⌋

9 Sequence diagrams

9.1 RxIndication

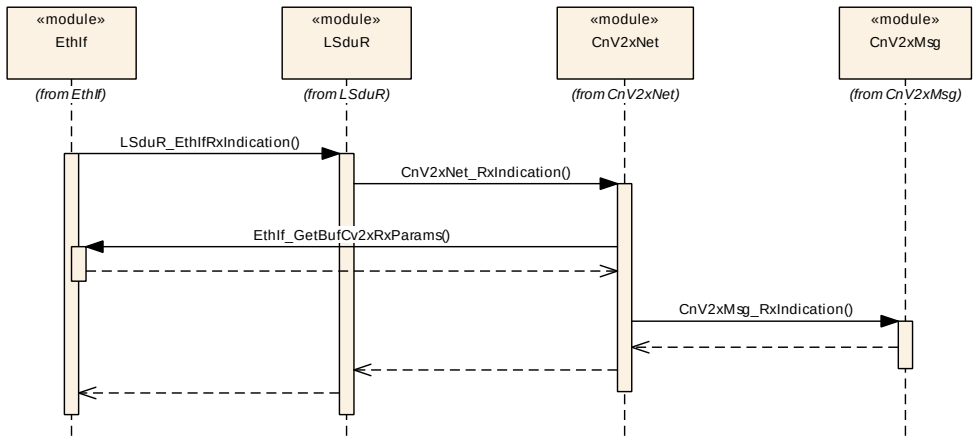

Figure 9.1: RxIndication

9.2 Transmission

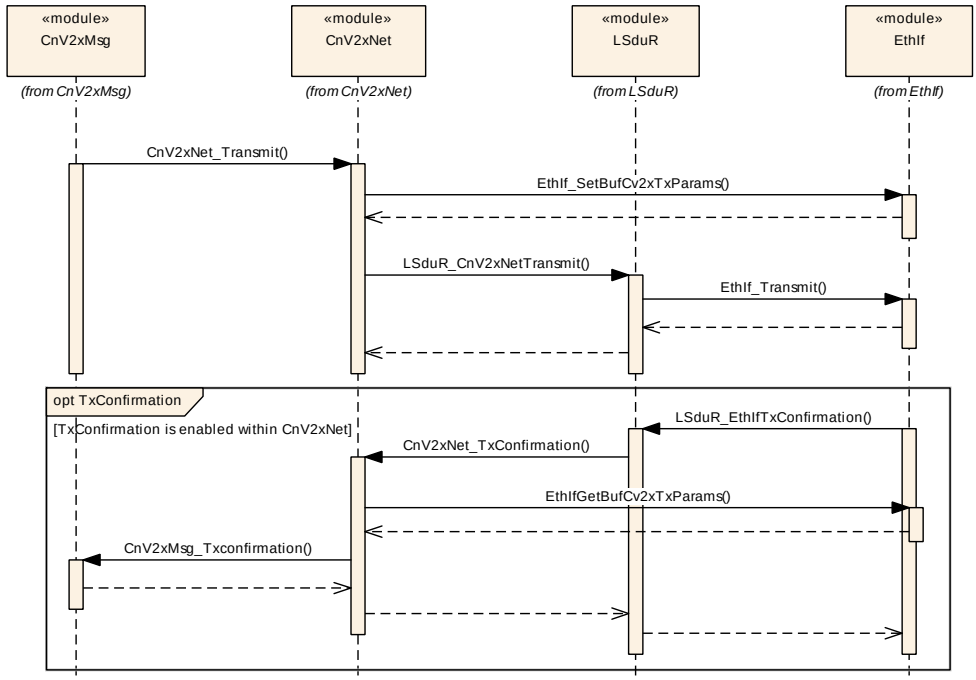

Figure 9.2: Transmission

10 Configuration specification

10.1 Containers and configuration parameters

The following chapters summarize all configuration parameters. The detailed meanings of the parameters are described in Chapter 7 and Chapter 8.

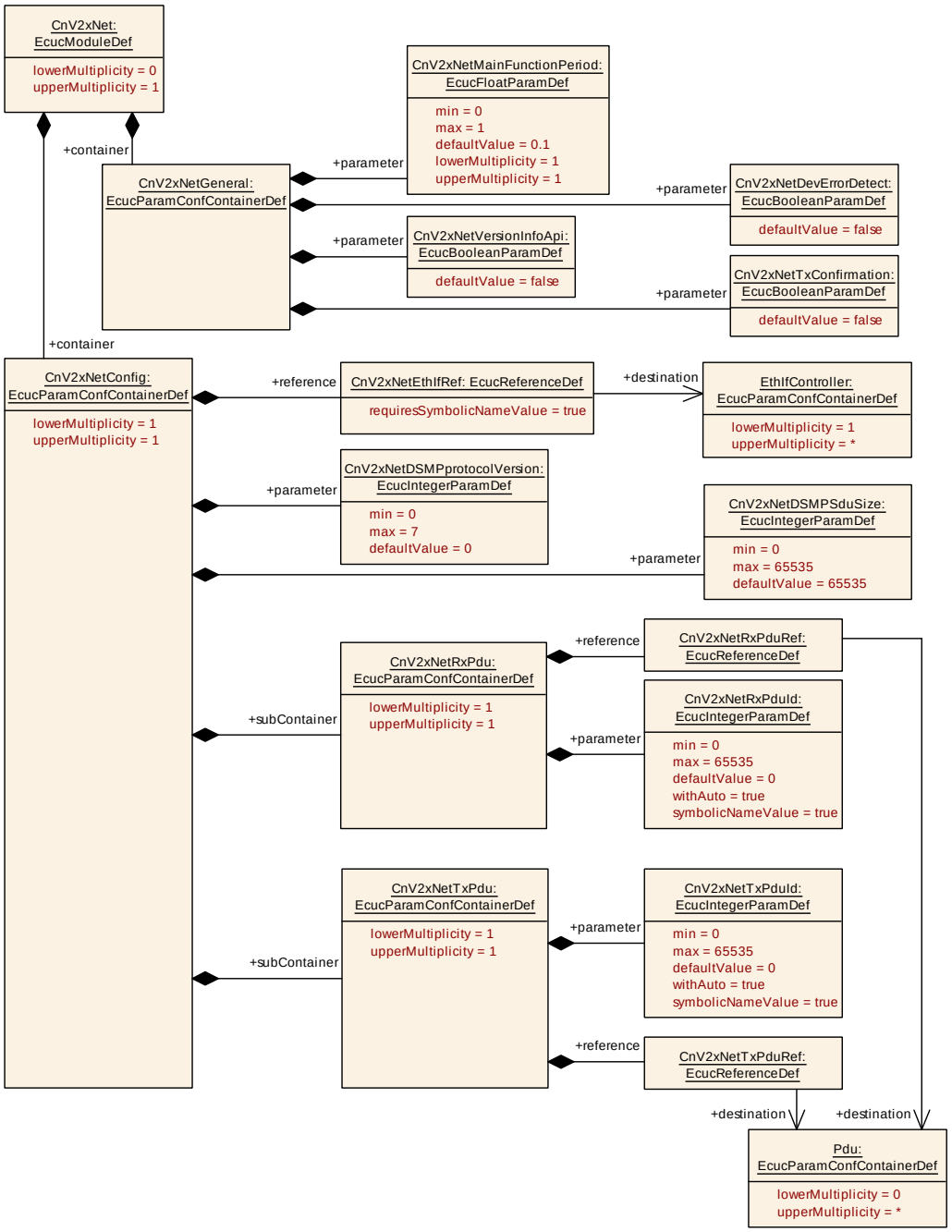

Figure 10.1: CnV2xNet

10.1.1 Variant

[SWS_CnV2xNet_03001]
Status: DRAFT
Upstream requirements: SRS_BSW_00345

⌈The CnV2xNet module only supports VARIANTPRECOMPILE⌋

10.1.2 CnV2xNet

[ECUC_CnV2xNet_00001] Definition of EcucModuleDef CnV2xNet

Status: DRAFT

⌈

Module Name	CnV2xNet
Description	Configuration of the CnV2xNet module.
Post-Build Variant Support	false
Supported Config Variants	VARIANT-PRE-COMPILE

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
CnV2xNetConfig	1	This container contains the configuration parameters and sub containers of the CnV2xNet module. **Tags:** atp.Status=draft
CnV2xNetGeneral	1	This container contains the configuration parameters of the BSW module CnV2xNet. **Tags:** atp.Status=draft

⌋

10.1.3 CnV2xNetGeneral

[ECUC_CnV2xNet_00002] Definition of EcucParamConfContainerDef CnV2xNet General

Status: DRAFT

⌈

Container Name	CnV2xNetGeneral
Parent Container	CnV2xNet
Description	This container contains the configuration parameters of the BSW module CnV2xNet. **Tags:** atp.Status=draft
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
CnV2xNetDevErrorDetect	1	[ECUC_CnV2xNet_00004]
CnV2xNetMainFunctionPeriod	1	[ECUC_CnV2xNet_00003]
CnV2xNetTxConfirmation	1	[ECUC_CnV2xNet_00006]
CnV2xNetVersionInfoApi	1	[ECUC_CnV2xNet_00005]

No Included Containers

⌋

[ECUC_CnV2xNet_00004] Definition of EcucBooleanParamDef CnV2xNetDevErrorDetect

Status: DRAFT

⌈

Parameter Name	CnV2xNetDevErrorDetect		
Parent Container	CnV2xNetGeneral		
Description	Switches the Default Error Tracer (Det) detection and notification ON or OFF. - true: enabled (ON) - false: disabled (OFF) **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_CnV2xNet_00003] Definition of EcucFloatParamDef CnV2xNetMainFunctionPeriod

Status: DRAFT

⌈

Parameter Name	CnV2xNetMainFunctionPeriod		
Parent Container	CnV2xNetGeneral		
Description	This parameter defines the schedule period of CnV2xNet_BsmBs_Main Function.Unit:[s] **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	]0 .. 1[		
Default value	0.1		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_CnV2xNet_00006] Definition of EcucBooleanParamDef CnV2xNetTxConfirmation

Status: DRAFT

⌈

Parameter Name	CnV2xNetTxConfirmation		
Parent Container	CnV2xNetGeneral		
Description	When enabled, transmission status information will be forwarded to the upper layer. - true: enabled (ON) - false: disabled (OFF) **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_CnV2xNet_00005] Definition of EcucBooleanParamDef CnV2xNetVersionInfoApi

Status: DRAFT

⌈

Parameter Name	CnV2xNetVersionInfoApi		
Parent Container	CnV2xNetGeneral		
Description	Enable/disables the API for reading the version information of the CnV2xNet Module. - true: enabled (ON) - false: disabled (OFF) **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

10.1.4 CnV2xNetConfig

[ECUC_CnV2xNet_00010] Definition of EcucParamConfContainerDef CnV2xNetConfig

Status: DRAFT

⌈

Container Name	CnV2xNetConfig
Parent Container	CnV2xNet
Description	This container contains the configuration parameters and sub containers of the CnV2x Net module. **Tags:** atp.Status=draft
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
CnV2xNetDSMPprotocolVersion	1	[ECUC_CnV2xNet_00008]
CnV2xNetDSMPSduSize	1	[ECUC_CnV2xNet_00009]
CnV2xNetEthIfRef	1	[ECUC_CnV2xNet_00007]

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
CnV2xNetRxPdu	1	Represents the received PDU. This PDU is usually linked to the EthIf via LSduR. It consumes meta data items of the types BROADCAST_8 and ETHERNET_MAC_64. **Tags:** atp.Status=draft
CnV2xNetTxPdu	1	Represents the transmitted PDU. This PDU is usually linked to the EthIf via LSduR. It produces meta data items of the type ETHERNET_MAC_64. **Tags:** atp.Status=draft

⌋

[ECUC_CnV2xNet_00008] Definition of EcucIntegerParamDef CnV2xNetDSMP-protocolVersion

Status: DRAFT

⌈

Parameter Name	CnV2xNetDSMPprotocolVersion		
Parent Container	CnV2xNetConfig		
Description	DSMP protocol version as defined in chapter 4.3.3.1 [13] **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 7		
Default value	0		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_CnV2xNet_00009] Definition of EcucIntegerParamDef CnV2xNet DSMPSduSize

Status: DRAFT

⌈

Parameter Name	CnV2xNetDSMPSduSize
Parent Container	CnV2xNetConfig
Description	Maximum size of DSMP-SDU in [Byte] **Tags:** atp.Status=draft
Multiplicity	1
Type	EcucIntegerParamDef

▽

△

Range	0 .. 65535		
Default value	65535		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_CnV2xNet_00007] Definition of EcucReferenceDef CnV2xNetEthIfRef

Status: DRAFT

⌈

Parameter Name	CnV2xNetEthIfRef		
Parent Container	CnV2xNetConfig		
Description	This is represents the reference to the Ethernet interface taken to transmit the C-V2X packets to. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	Symbolic name reference to EthIfController		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

10.1.5 CnV2xNetRxPdu

[ECUC_CnV2xNet_00011] Definition of EcucParamConfContainerDef CnV2xNetRxPdu

Status: DRAFT

⌈

Container Name	CnV2xNetRxPdu
Parent Container	CnV2xNetConfig
Description	Represents the received PDU. This PDU is usually linked to the EthIf via LSduR. It consumes meta data items of the types BROADCAST_8 and ETHERNET_MAC_64. **Tags:** atp.Status=draft
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
CnV2xNetRxPduId	1	[ECUC_CnV2xNet_00013]
CnV2xNetRxPduRef	1	[ECUC_CnV2xNet_00014]

No Included Containers

⌋

[ECUC_CnV2xNet_00013] Definition of EcucIntegerParamDef CnV2xNetRxPduId

Status: DRAFT

⌈

Parameter Name	CnV2xNetRxPduId		
Parent Container	CnV2xNetRxPdu		
Description	PDU identifier used for RxIndication from LSduR. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	0		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU withAuto = true		

⌋

[ECUC_CnV2xNet_00014] Definition of EcucReferenceDef CnV2xNetRxPduRef

Status: DRAFT

⌈

Parameter Name	CnV2xNetRxPduRef		
Parent Container	CnV2xNetRxPdu		
Description	Reference to the global PDU. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	Reference to Pdu		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

⌋

10.1.6 CnV2xNetTxPdu

[ECUC_CnV2xNet_00012] Definition of EcucParamConfContainerDef CnV2xNetTxPdu

Status: DRAFT

⌈

Container Name	CnV2xNetTxPdu
Parent Container	CnV2xNetConfig
Description	Represents the transmitted PDU. This PDU is usually linked to the EthIf via LSduR. It produces meta data items of the type ETHERNET_MAC_64. **Tags:** atp.Status=draft
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
CnV2xNetTxPduId	1	[ECUC_CnV2xNet_00015]
CnV2xNetTxPduRef	1	[ECUC_CnV2xNet_00016]

No Included Containers

⌋

[ECUC_CnV2xNet_00015] Definition of EcucIntegerParamDef CnV2xNetTxPduId

Status: DRAFT

⌈

Parameter Name	CnV2xNetTxPduId		
Parent Container	CnV2xNetTxPdu		
Description	PDU identifier used for TxConfirmation from LSduR. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	0		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU withAuto = true		

⌋

[ECUC_CnV2xNet_00016] Definition of EcucReferenceDef CnV2xNetTxPduRef

Status: DRAFT

⌈

Parameter Name	CnV2xNetTxPduRef		
Parent Container	CnV2xNetTxPdu		
Description	Reference to the global PDU. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	Reference to Pdu		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

⌋

10.2 Constraints

[CP_SWS_CnV2xNet_CONSTR_00001] Support of PDUs with `KeepLocalPduBuffer` set to FALSE ⌈The configuration of `CnV2xNetTxPdu` and `CnV2xNetRxPdu` shall refer to PDUs where `KeepLocalPduBuffer` is set to FALSE. Otherwise the configuration shall be rejected as invalid.⌋

A Not applicable requirements

B Change history of AUTOSAR traceable items

Please note that the lists in this chapter also include traceable items that have been removed from the specification in a later version. These items do not appear as hyperlinks in the document.

B.1 Traceable item history of this document according to AUTOSAR Release R24-11

B.1.1 Added Specification Items in R24-11

Number	Heading
[CP_SWS_CnV2xNet_91000]	Definition of callback function CnV2xNet_RxIndication
[CP_SWS_CnV2xNet_91001]	Definition of callback function CnV2xNet_TxConfirmation
[ECUC_CnV2xNet_00010]	Definition of EcucParamConfContainerDef CnV2xNetConfig
[ECUC_CnV2xNet_00011]	Definition of EcucParamConfContainerDef CnV2xNetRxPdu
[ECUC_CnV2xNet_00012]	Definition of EcucParamConfContainerDef CnV2xNetTxPdu
[ECUC_CnV2xNet_00013]	Definition of EcucIntegerParamDef CnV2xNetRxPduId
[ECUC_CnV2xNet_00014]	Definition of EcucReferenceDef CnV2xNetRxPduRef
[ECUC_CnV2xNet_00015]	Definition of EcucIntegerParamDef CnV2xNetTxPduId
[ECUC_CnV2xNet_00016]	Definition of EcucReferenceDef CnV2xNetTxPduRef

Table B.1: Added Specification Items in R24-11

B.1.2 Changed Specification Items in R24-11

Number	Heading
[CP_SWS_CnV2xNet_00111]	
[CP_SWS_CnV2xNet_01001]	Definition of imported datatypes of module CnV2xNet
[CP_SWS_CnV2xNet_02005]	Definition of API function CnV2xNet_Transmit
[CP_SWS_CnV2xNet_02021]	Definition of mandatory interfaces required by module CnV2xNet
[CP_SWS_CnV2xNet_02025]	
[ECUC_CnV2xNet_00001]	Definition of EcucModuleDef CnV2xNet
[ECUC_CnV2xNet_00002]	Definition of EcucParamConfContainerDef CnV2xNetGeneral
[ECUC_CnV2xNet_00007]	Definition of EcucReferenceDef CnV2xNetEthIfRef
[ECUC_CnV2xNet_00008]	Definition of EcucIntegerParamDef CnV2xNetDSMPprotocol Version

▽

△

Number	Heading
[ECUC_CnV2xNet_00009]	Definition of EcucIntegerParamDef CnV2xNetDSMPSduSize

Table B.2: Changed Specification Items in R24-11

B.1.3 Deleted Specification Items in R24-11

Number	Heading
[CP_SWS_CnV2xNet_00104]	
[CP_SWS_CnV2xNet_02018]	Definition of callback function CnV2xNet_RxIndication
[CP_SWS_CnV2xNet_02019]	Definition of callback function CnV2xNet_TxConfirmation

Table B.3: Deleted Specification Items in R24-11

B.1.4 Added Constraints in R24-11

Number	Heading
[CP_SWS_CnV2xNet_CONSTR_00001]	Support of PDUs with `KeepLocalPduBuffer` set to FALSE

Table B.4: Added Constraints in R24-11

B.1.5 Changed Constraints in R24-11

none

B.1.6 Deleted Constraints in R24-11

none

B.2 Traceable item history of this document according to AUTOSAR Release R23-11

B.2.1 Added Specification Items in R23-11

none

B.2.2 Changed Specification Items in R23-11

none

B.2.3 Deleted Specification Items in R23-11

none