Document Title	Specification of UDP Network Management
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	414
Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R21-11

Document Change History

Date	Release	Changed by	Change Description
2021-11-25	R21-11	AUTOSAR Release Management	• Added handling of internal requested Pnc • Improved synchronized Pnc shutdown • NM PDU filter algorithm and aggregation of internal and external requested partial networks is now obsolete and replaced • Traceability directly to RS_Nm
2020-11-30	R20-11	AUTOSAR Release Management	• Updates for CONC 641 VSNM • Updates for Light CONC 685 • Minor changes
2019-11-28	R19-11	AUTOSAR Release Management	• Det error handling corrected • Harmonization of API • Minor corrections • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• Header file cleanup • Minor corrections
2017-12-08	4.3.1	AUTOSAR Release Management	• Node Detection Configuration per channel • Det error handling corrected • Bug fixes and editorial changes
2016-11-30	4.3.0	AUTOSAR Release Management	• Added Trigger Transmit feature • Car Wakeup support completed • Immediate TX Transmission corrected • Editorial changes

Document Change History			
Date	**Release**	**Changed by**	**Change Description**
2015-07-31	4.2.2	AUTOSAR Release Management	• Revised Error Classification • Added support for Car Wakeup • Bug fixes and editorial changes
2014-10-31	4.2.1	AUTOSAR Release Management	• Harmonization of API description • Revised Partial Networking Requirements • Extended Production Errors • Editorial Changes
2014-03-31	4.1.3	AUTOSAR Release Management	• Minor bug fixes • Editorial Changes
2013-10-31	4.1.2	AUTOSAR Release Management	• Revised Spontaneous Transmission • Editorial changes • Removed chapter(s) on change documentation
2013-03-15	4.1.1	AUTOSAR Administration	• Added support for Partial Networking • Added updated production errors • Editorial changes
2011-12-22	4.0.3	AUTOSAR Administration	• Support coordinated shutdown • New traceability mechanism
2010-09-30	3.1.5	AUTOSAR Administration	• ComStack Harmonization • Harmonization of NM interfaces
2010-02-02	3.1.4	AUTOSAR Administration	• Initial Release

Table of Contents

1 Introduction and Functional Overview 7
2 Acronyms and abbreviations 8
3 Related documentation 9
3.1 Input documents 9
3.2 Related standards and norms 10
3.3 Related specification 10
4 Constraints and assumptions 11
4.1 Limitations 11
4.2 Applicability to car domains 11
5 Dependencies on other modules 12
5.1 File Structure 12
5.1.1 Code File Structure 12
6 Requirements traceability 13
7 Functional specification 18
7.1 Coordination algorithm 18
7.2 Operational Modes 19
7.2.1 Network Mode 19
7.2.2 Prepare Bus-Sleep Mode 22
7.2.3 Bus-Sleep Mode 24
7.3 Network states 25
7.4 Initialization 25
7.5 Execution 26
7.5.1 Processor architecture 26
7.5.2 Timing parameters 27
7.6 Communication Scheduling 27
7.6.1 NM Message Transmission 27
7.6.2 NM Message Reception 30
7.7 Additional features 30
7.7.1 Detection of Remote Sleep Indication (optional) 30
7.7.2 User Data (optional) 31
7.7.3 Passive Mode (optional) 32
7.7.4 State change notification (optional) 32
7.7.5 Communication Control (optional) 32
7.7.6 NM Coordinator synchronization support (optional) 34
7.8 Partial Networking 34
7.8.1 Rx Handling of NM PDUs 34
7.8.2 Tx Handling of NM PDUs 36
7.8.3 Handling of Internal Requested Partial Network Clusters 40
7.8.4 NM PDU Filter Algorithm (OBSOLETE) 40
7.8.5 Aggregation of Internal and External Requested Partial Networks (OBSOLETE) 41
7.8.6 Aggregation of External Requested Partial Networks (OBSOLETE) 43

7.8.7 Spontaneous Transmission of NM-PDUs via UdpNm_NetworkRequest 45
7.9 Payload (PDU) Structure 46
7.10 Functional requirements on UdpNm API 48
7.11 Car Wakeup 48
7.12 Error Classification 49
7.12.1 Development Errors 50
7.12.2 Run Time Errors 50
7.12.3 Transient Faults 51
7.12.4 Production Errors 51
7.12.5 Extended Production Errors 51
7.13 Scheduling of the main function 51
7.14 Application notes 51
7.14.1 Wakeup notification 51
7.14.2 Coordination of coupled networks 51
7.15 Version check 51
7.16 Parameter check 52

8 API specification 53

8.1 Imported Types 53
8.2 Type Definitions 54
8.2.1 UdpNm_ConfigType 54
8.2.2 UdpNm_PduPositionType 55
8.3 Function definitions 55
8.3.1 UdpNm_Init 55
8.3.2 UdpNm_PassiveStartUp 56
8.3.3 UdpNm_NetworkRequest 56
8.3.4 UdpNm_NetworkRelease 57
8.3.5 UdpNm_DisableCommunication 58
8.3.6 UdpNm_EnableCommunication 58
8.3.7 UdpNm_SetUserData 59
8.3.8 UdpNm_GetUserData 60
8.3.9 UdpNm_GetNodeIdentifier 61
8.3.10 UdpNm_GetLocalNodeIdentifier 62
8.3.11 UdpNm_RepeatMessageRequest 62
8.3.12 UdpNm_GetPduData 63
8.3.13 UdpNm_GetState 64
8.3.14 UdpNm_GetVersionInfo 64
8.3.15 UdpNm_RequestBusSynchronization 65
8.3.16 UdpNm_CheckRemoteSleepIndication 66
8.3.17 UdpNm_SetSleepReadyBit 67
8.3.18 UdpNm_Transmit 67
8.3.19 UdpNm_PnLearningRequest 68
8.3.20 UdpNm_RequestSynchronizedPncShutdown 69
8.4 Call-back notifications 70
8.4.1 UdpNm_SoAdIfTxConfirmation 70
8.4.2 UdpNm_SoAdIfRxIndication 71
8.4.3 UdpNm_SoAdIfTriggerTransmit 72
8.5 Scheduled Functions 74
8.5.1 UdpNm_MainFunction_<Instance Id> 74

8.6 Expected Interfaces ... 74
8.6.1 Mandatory Interfaces ... 74
8.6.2 Optional Interfaces ... 76
8.6.3 Configurable interfaces ... 77
8.7 Service Interfaces ... 77
8.8 UML State chart diagram ... 78

9 Sequence diagrams and Transition Tables ... 79

9.1 UdpNmTransmission ... 79
9.2 UdpNm Reception ... 79

10 Configuration specification ... 81

10.1 How to read this chapter ... 81
10.2 Containers and configuration parameters ... 81
10.2.1 UdpNm ... 82
10.2.2 UdpNmGlobalConfig ... 82
10.2.3 UdpNmChannelConfig ... 91
10.2.4 UdpNmRxPdu ... 105
10.2.5 UdpNmTxPdu ... 106
10.2.6 UdpNmUserDataTxPdu ... 107
10.2.7 UdpNmPnInfo ... 108
10.2.8 UdpNmPnFilterMaskByte ... 110
10.3 Published parameters ... 111

11 Not applicable requirements ... 112

1 Introduction and Functional Overview

This document describes the concept, core functionality, optional features, interfaces and configuration issues of the AUTOSAR UDP Network Management (UdpNm).
UdpNm is intended to be an optional feature. It is intended to work together with a TCP/IP Stack, independent of the physical layer of the communication system used.
The AUTOSAR UDP Network Management is a hardware independent protocol that can be used on TCP/IP based systems (for limitations refer to chapter 4.1). Its main purpose is to coordinate the transition between normal operation and bus-sleep mode of the network.
In addition to the core functionality optional features are provided e.g. to implement a service to detect all present nodes or to detect if all other nodes are ready to sleep.
The UDP Network Management (UdpNm) function provides an adaptation between Network Management Interface (Nm) and a TCP/IP Stack (TCP/IP). For a general understanding of the AUTOSAR Network Management functionality please refer to [9].

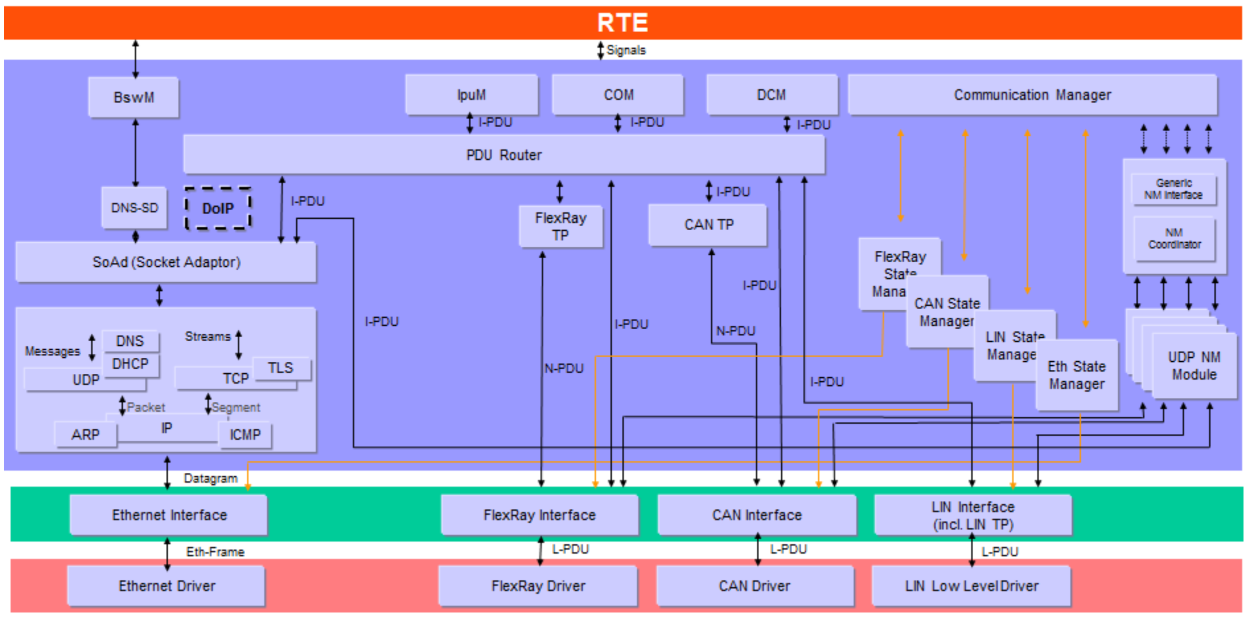

Figure 1: Extended AUTOSAR Communication Stack.

2 Acronyms and abbreviations

Acronym or Abbreviation:	***Description:***
API	Application Programming Interface
BSW	Basic Software
CWU	Car Wakeup
EthIf	Ethernet Interface
DET	Default Error Tracer
IP	Internet Protocol
NM	Network Management
PDU	Protocol Data Unit
PNL	Partial Network Learning
SDU	Service Data Unit
TCP	Transmission Control Protocol
TCP/IP	A family of communication protocols used in computer networks
UDP	User Datagram Protocol
PNI	Partial Network Information
UdpNm	UDP Network Management

Term:	***Description:***
PDU transmission ability is disabled	This means that the NM message transmission has been disabled by the optional service UdpNm_DisableCommunication.
Repeat Message Request Bit Indication	UdpNm_SoAdIfRxIndication finds the Repeat Message Bit set in the Control Bit Vector of a received NM message.
NM PDU	Refers to the payload transmitted in a packet. It contains the NM User Data as well as the Control Bit Vector and the Source Node Identifier.
NM Packet	Refers to an Ethernet Frame containing an IP as well as a UDP header in addition to the data (PDU) transmitted by the NM in the payload section.
NM Message	Most abstract term referring to any single information item transferred within the methodology of the NM algorithm.
Bus-Off state	Refers to a situation where no cable is connected to the Ethernet HW.
Top-level PNC coordinator	The top-level PNC coordinator is an ECU that acts as PNC gateway in the network and that handles at least one PNC as actively coordinated on all assigned channels. If synchronized PNC shutdown is enabled, the top-level PNC coordinator triggers for these PNCs the shutdown, if no other ECU in the network request them.
Intermediate PNC coordinator	An intermediate PNC coordinator is an ECU that acts as PNC gateway in the network and that handles at least one PNC as passively coordinated on at least one assigned channel. If synchronized PNC shutdown is enabled, it forwards received shutdown requests for these PNCs to the corresponding actively coordinated channels and starts their shutdown accordingly.
PNC leaf node	A PNC leaf node is an ECU that acts not as a PNC coordinator at all in the network. It processes PN shutdown message as usual NM messages.
PN shutdown message	A top-level PNC coordinator transmit PN shutdown messages to indicate a synchronized PNC shutdown across the PN topology. A PN shutdown message is as NM message which has PNSR bit in the control bit vector and all PNCs which are indicated for a synchronized shutdown set to '1'.

3 Related documentation

3.1 Input documents

[1] Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf

[2] General Requirements on Basic Software Modules
AUTOSAR_SRS_BSWGeneral.pdf

[3] Requirements on Network Management
AUTOSAR_SRS_NetworkManagement.pdf

[4] Specification of Ethernet Interface
AUTOSAR_SWS_EthernetInterface.pdf

[5] Specification of FlexRay Network Management
AUTOSAR_SWS_FlexRayNetworkManagement.pdf

[6] Specification of Communication Stack Types
AUTOSAR_SWS_CommunicationStackTypes.pdf

[7] Specification of ECU Configuration
AUTOSAR_TPS_ECUConfiguration.pdf

[8] Specification of BSW Scheduler
AUTOSAR_SWS_BSW_Scheduler.pdf

[9] Specification of Generic Network Management Interface
AUTOSAR_SWS_NetworkManagementInterface.pdf

[10] Specification of Communication Manager
AUTOSAR_SWS_ComManager.pdf

[11] Specification of ECU State Manager
AUTOSAR_SWS_ECUStateManager.pdf

[12] Specification of Operating System
AUTOSAR_SWS_OS.pdf

[13] Specification of Default Error Tracer
AUTOSAR_SWS_Default ErrorTracer.pdf

[14] Specification of Standard Types
AUTOSAR_SWS_StandardTypes.pdf

[15] Specification of Platform Types
AUTOSAR_SWS_PlatformTypes.pdf

[16] Specification of Compiler Abstraction
AUTOSAR_SWS_CompilerAbstraction.pdf

[17] Basic Software Module Description Template
AUTOSAR_TPS_BSWModuleDescriptionTemplate.pdf

[18] Specification of Socket Adaptor
AUTOSAR_SWS_SocketAdaptor.pdf

[19] Requirements on Ethernet
AUTOSAR_SRS_Ethernet.pdf

[20] List of Basic Software Modules
AUTOSAR_TR_BSWModuleList

[21] General Specification of Basic Software Modules
AUTOSAR_SWS_BSWGeneral.pdf

[22] Specification of the AUTOSAR Network Management Protocol
AUTOSAR_PRS_NetworkManagementProtocol.pdf

[23] Specification of SystemTemplate
AUTOSAR_TPS_SystemTemplate

3.2 Related standards and norms

[24] IEEE
http://www.opengroup.org/onlinepubs/000095399/
[25] ISO 14229 Road Vehicles – Unified Diagnostic Services (UDS)

3.3 Related specification

AUTOSAR provides a General Specification on Basic Software modules [21] (SWS BSW General), which is also valid for UDP Network Management.

Thus, the specification SWS BSW General shall be considered as additional and required specification for UDP Network Management.

4 Constraints and assumptions

4.1 Limitations

1. One instance of UdpNm is associated with only one NM-Cluster in one network. One NM-Cluster can have only one instance of UdpNm in one node.
2. One instance of UdpNm is associated with only one network within the same ECU.
3. UdpNm is only applicable for TCP/IP based systems.

Figure 2 presents an AUTOSAR NM stack within an example ECU belonging to two UDP NM-clusters.

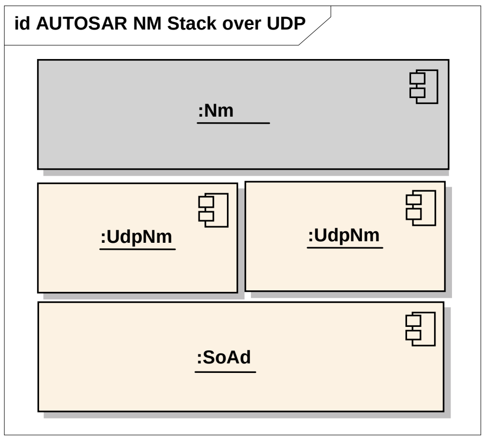

Figure 2: AUTOSAR NM stack within an example ECU belonging to two UDP NM-clusters

[SWS_UdpNm_00131]⌈ The AUTOSAR UdpNm algorithm shall support up to 250 nodes per NM-Cluster by default.
Note: The AUTOSAR UdpNm algorithm can support an arbitrary number of nodes per NM-cluster (even more than default 250 nodes per cluster, if necessary) – it is only a matter of configuration, since the upper limit is not fixed and depends on the trade off between response time, fault-tolerance and resulted bus load configured for the AUTOSAR UdpNm coordination algorithm. This might depend on the physical layer used. ⌋()

4.2 Applicability to car domains

N/A

5 Dependencies on other modules

UDP Network Management (UdpNm) uses services of the TCP/IP Stack and provides services to the Generic Network Management Interface (Nm).

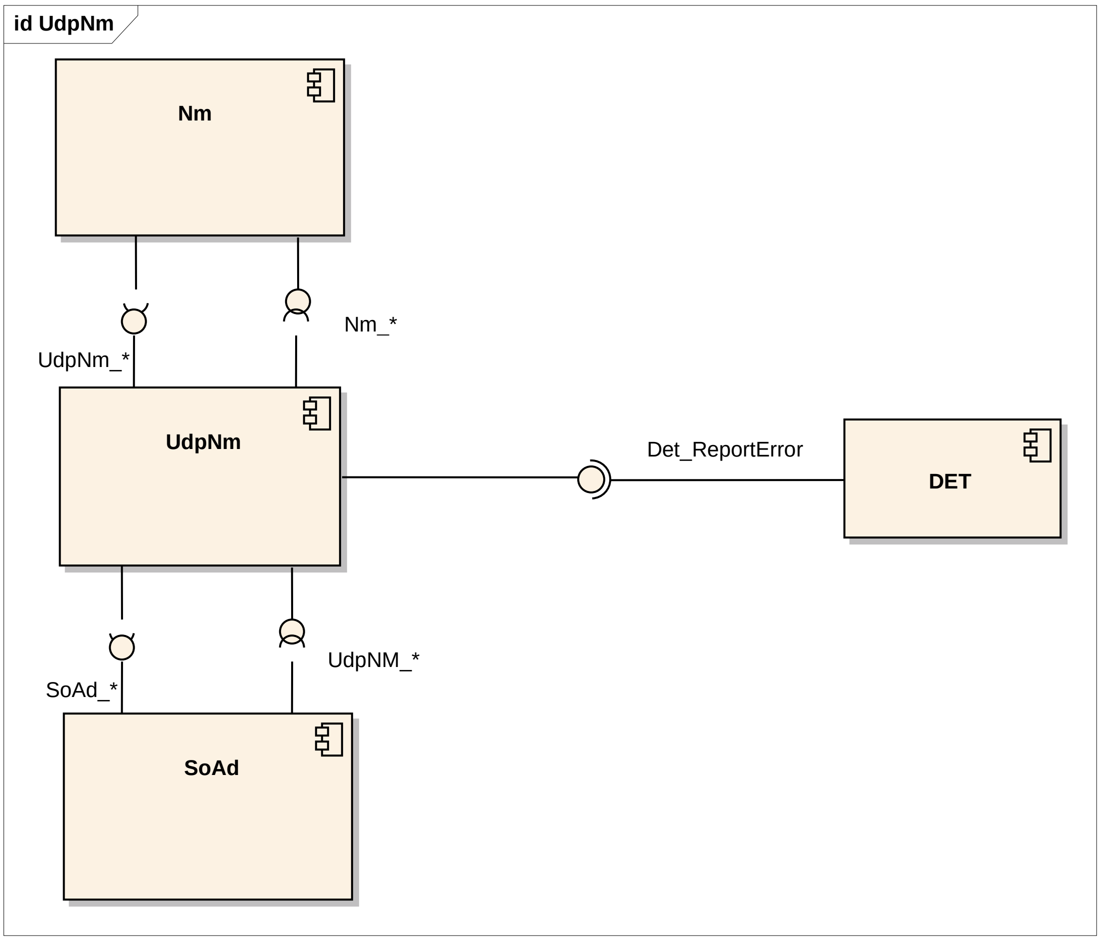

Figure 3: Dependencies on other modules.

5.1 File Structure

5.1.1 Code File Structure

[SWS_UdpNm_00081]⌈ The code file structure shall not be fully defined within this specification. However, the code file structure shall include the following files:

`UdpNm_Lcfg.c` (for link time configurable parameters)

`UdpNm_PBcfg.c` (for post build time configurable parameters)

These files shall contain all link time post build time configurable parameters.
⌋(SRS_BSW_00419, SRS_BSW_00346, SRS_BSW_00308)

6 Requirements traceability

Requirement	Description	Satisfied by
RS_Nm_00046	It shall be possible to trigger the startup of all Nodes at any Point in Time	SWS_UdpNm_NA_00999
RS_Nm_00050	The NM shall provide the current state of NM	SWS_UdpNm_NA_00999
RS_Nm_00052	The NM interface shall signal to the application that all other ECUs are ready to sleep.	SWS_UdpNm_NA_00999
RS_Nm_00054	There shall be a deterministic time from the point where all nodes agree to go to bus sleep to the point where bus is switched off.	SWS_UdpNm_NA_00999
RS_Nm_00137	NM shall perform communication system error handling for errors that have impact on the NM behavior.	SWS_UdpNm_00379, SWS_UdpNm_00466, SWS_UdpNm_00467
RS_Nm_00142	NM shall provide a mechanism to limit its bus load.	SWS_UdpNm_NA_00999
RS_Nm_00144	NM shall support communication clusters of up to 64 ECUs	SWS_UdpNm_NA_00999
RS_Nm_00151	The Network Management algorithm shall allow any node to integrate into an already running NM cluster	SWS_UdpNm_NA_00999
RS_Nm_00153	The Network Management shall optionally provide a possibility to detect present nodes	SWS_UdpNm_00014, SWS_UdpNm_00111, SWS_UdpNm_00112, SWS_UdpNm_00113, SWS_UdpNm_00119, SWS_UdpNm_00120, SWS_UdpNm_00121, SWS_UdpNm_00468, SWS_UdpNm_NA_00999
RS_Nm_00154	The Network Management API shall be independent from the communication bus	SWS_UdpNm_NA_00999
RS_Nm_02503	The NM API shall optionally give the possibility to send user data	SWS_UdpNm_00315, SWS_UdpNm_00317, SWS_UdpNm_00377, SWS_UdpNm_00464, SWS_UdpNm_00479, SWS_UdpNm_00495
RS_Nm_02509	The NM interface shall signal to the application that at least one ECU is not ready to sleep anymore.	SWS_UdpNm_NA_00999
RS_Nm_02512	The NM shall give the possibility to enable or disable the network management related communication configured for an active NM node	SWS_UdpNm_00178, SWS_UdpNm_00215, SWS_UdpNm_00216
RS_Nm_02519	The NM Control Bit Vector shall	SWS_UdpNm_00486, SWS_UdpNm_00489,

	contain a PNI (Partial Network Information) bit.	SWS_UdpNm_00493
RS_Nm_02527	Nm shall implement a filter algorithm dropping all NM messages that are not relevant for the ECU	SWS_UdpNm_00487
RS_Nm_02540	The NM Control Bit Vector shall contain a PN shutdown request bit.	SWS_UdpNm_00045, SWS_UdpNm_00475, SWS_UdpNm_00490
RS_Nm_02541	NM shall define a common layout of NM messages.	SWS_UdpNm_00478, SWS_UdpNm_00488
RS_Nm_02542	The NM of the top-level PNC coordinator shall set the PN shutdown request bit if a least one PNC is released	SWS_UdpNm_00475, SWS_UdpNm_00490
RS_Nm_02545	NM shall handle requests for synchronized PNC shutdown	SWS_UdpNm_00474, SWS_UdpNm_00475, SWS_UdpNm_00476, SWS_UdpNm_00477, SWS_UdpNm_00480, SWS_UdpNm_00490, SWS_UdpNm_00494, SWS_UdpNm_91002
RS_Nm_02546	UdpNm shall support Partial Networking on Ethernet	SWS_UdpNm_00486, SWS_UdpNm_00487, SWS_UdpNm_00489, SWS_UdpNm_00493
RS_Nm_02547	Nm shall be able to propagate and evaluate the need for Partial Networking Learning (optional)	SWS_UdpNm_00486, SWS_UdpNm_00489, SWS_UdpNm_00493
SRS_BSW_00005	Modules of the ÂµC Abstraction Layer (MCAL) may not have hard coded horizontal interfaces	SWS_UdpNm_NA_00999
SRS_BSW_00006	The source code of software modules above the ÂµC Abstraction Layer (MCAL) shall not be processor and compiler dependent.	SWS_UdpNm_NA_00999
SRS_BSW_00010	The memory consumption of all Basic SW Modules shall be documented for a defined configuration for all supported platforms.	SWS_UdpNm_NA_00999
SRS_BSW_00160	Configuration files of AUTOSAR Basic SW module shall be readable for human beings	SWS_UdpNm_NA_00999
SRS_BSW_00161	The AUTOSAR Basic Software shall provide a microcontroller abstraction layer which provides a standardized interface to higher software layers	SWS_UdpNm_NA_00999
SRS_BSW_00162	The AUTOSAR Basic Software shall provide a hardware abstraction layer	SWS_UdpNm_NA_00999

SRS_BSW_00164	The Implementation of interrupt service routines shall be done by the Operating System, complex drivers or modules	SWS_UdpNm_NA_00999
SRS_BSW_00168	SW components shall be tested by a function defined in a common API in the Basis-SW	SWS_UdpNm_NA_00999
SRS_BSW_00170	The AUTOSAR SW Components shall provide information about their dependency from faults, signal qualities, driver demands	SWS_UdpNm_NA_00999
SRS_BSW_00172	The scheduling strategy that is built inside the Basic Software Modules shall be compatible with the strategy used in the system	SWS_UdpNm_NA_00999
SRS_BSW_00305	Data types naming convention	SWS_UdpNm_NA_00999
SRS_BSW_00306	AUTOSAR Basic Software Modules shall be compiler and platform independent	SWS_UdpNm_NA_00999
SRS_BSW_00307	Global variables naming convention	SWS_UdpNm_NA_00999
SRS_BSW_00308	AUTOSAR Basic Software Modules shall not define global data in their header files, but in the C file	SWS_UdpNm_00081
SRS_BSW_00309	All AUTOSAR Basic Software Modules shall indicate all global data with read-only purposes by explicitly assigning the const keyword	SWS_UdpNm_NA_00999
SRS_BSW_00312	Shared code shall be reentrant	SWS_UdpNm_NA_00999
SRS_BSW_00314	All internal driver modules shall separate the interrupt frame definition from the service routine	SWS_UdpNm_NA_00999
SRS_BSW_00321	The version numbers of AUTOSAR Basic Software Modules shall be enumerated according specific rules	SWS_UdpNm_NA_00999
SRS_BSW_00325	The runtime of interrupt service routines and functions that are running in interrupt context shall be kept short	SWS_UdpNm_NA_00999
SRS_BSW_00328	All AUTOSAR Basic Software Modules shall avoid the duplication of code	SWS_UdpNm_NA_00999
SRS_BSW_00330	It shall be allowed to use macros instead of functions where source code is used and runtime is critical	SWS_UdpNm_NA_00999

SRS_BSW_00331	All Basic Software Modules shall strictly separate error and status information	SWS_UdpNm_NA_00999
SRS_BSW_00333	For each callback function it shall be specified if it is called from interrupt context or not	SWS_UdpNm_NA_00999
SRS_BSW_00334	All Basic Software Modules shall provide an XML file that contains the meta data	SWS_UdpNm_NA_00999
SRS_BSW_00335	Status values naming convention	SWS_UdpNm_NA_00999
SRS_BSW_00336	Basic SW module shall be able to shutdown	SWS_UdpNm_NA_00999
SRS_BSW_00341	Module documentation shall contains all needed informations	SWS_UdpNm_NA_00999
SRS_BSW_00346	All AUTOSAR Basic Software Modules shall provide at least a basic set of module files	SWS_UdpNm_00081
SRS_BSW_00347	A Naming seperation of different instances of BSW drivers shall be in place	SWS_UdpNm_NA_00999
SRS_BSW_00375	Basic Software Modules shall report wake-up reasons	SWS_UdpNm_NA_00999
SRS_BSW_00377	A Basic Software Module can return a module specific types	SWS_UdpNm_NA_00999
SRS_BSW_00410	Compiler switches shall have defined values	SWS_UdpNm_NA_00999
SRS_BSW_00413	An index-based accessing of the instances of BSW modules shall be done	SWS_UdpNm_NA_00999
SRS_BSW_00415	Interfaces which are provided exclusively for one module shall be separated into a dedicated header file	SWS_UdpNm_NA_00999
SRS_BSW_00416	The sequence of modules to be initialized shall be configurable	SWS_UdpNm_NA_00999
SRS_BSW_00417	Software which is not part of the SW-C shall report error events only after the DEM is fully operational.	SWS_UdpNm_NA_00999
SRS_BSW_00419	If a pre-compile time configuration parameter is implemented as "const" it should be placed into a separate c-file	SWS_UdpNm_00081
SRS_BSW_00423	BSW modules with AUTOSAR interfaces shall be describable with the means of the SW-C Template	SWS_UdpNm_NA_00999
SRS_BSW_00424	BSW module main processing	SWS_UdpNm_NA_00999

	functions shall not be allowed to enter a wait state	
SRS_BSW_00425	The BSW module description template shall provide means to model the defined trigger conditions of schedulable objects	SWS_UdpNm_NA_00999
SRS_BSW_00426	BSW Modules shall ensure data consistency of data which is shared between BSW modules	SWS_UdpNm_NA_00999
SRS_BSW_00427	ISR functions shall be defined and documented in the BSW module description template	SWS_UdpNm_NA_00999
SRS_BSW_00429	Access to OS is restricted	SWS_UdpNm_NA_00999
SRS_BSW_00432	Modules should have separate main processing functions for read/receive and write/transmit data path	SWS_UdpNm_NA_00999

7 Functional specification

7.1 Coordination algorithm

The AUTOSAR UdpNm is based on decentralized direct network management strategy, which means that every network node performs activities self-sufficient depending only on the UDP packets received and/or transmitted within the communication system.

The AUTOSAR UdpNm coordination algorithm is based on periodic NM packets, which are received by all nodes in the cluster via broadcast transmission. Reception of NM packets indicates that sending nodes want to keep the NM-cluster awake. If any node is ready to go to the Bus-Sleep Mode, it stops sending NM packets, but as long as NM packets from other nodes are received, it postpones transition to the Bus-Sleep Mode. Finally, if a dedicated timer elapses because no NM packets are received anymore, every node initiates transition to the Bus-Sleep Mode.
If any node in the NM-cluster requires bus-communication, it can keep the NM-cluster awake by transmitting NM packets. For more details concerning the wakeup procedure itself, please refer to [10].

The main concept of the AUTOSAR UdpNm coordination algorithm can be defined by the following two key-requirements:

[SWS_UdpNm_00087]⌈ Every network node shall transmit periodic NM PDUs as long as it requires bus-communication; otherwise it shall not transmit NM PDUs. ⌋()

[SWS_UdpNm_00088]⌈If `UdpNmStayInPbsEnabled` is disabled and bus communication in a UdpNm cluster is released and there are no Network Management PDUs on the bus for a configurable amount of time determined by `UdpNmTimeoutTime` + `UdpNmWaitBusSleepTime` (both configuration parameters) transition into the Bus-Sleep Mode shall be performed. ⌋()

The overall state machine of the AUTOSAR UdpNm coordination algorithm can be defined as follows:

[SWS_UdpNm_00089]⌈ The AUTOSAR UdpNm state machine shall contain states, transitions and triggers required for the AUTOSAR UdpNm coordination algorithm as seen from the point of view of one single node in the NM cluster. ⌋()

Note: A UML state chart of the AUTOSAR UdpNm state machine from the point of view of one single node in the NM cluster can be found in the API specifications chapter 8

7.2 Operational Modes

This chapter describes the operational modes of the AUTOSAR UdpNm coordination algorithm.

[SWS_UdpNm_00092]⌈ The AUTOSAR UdpNm shall contain three operational modes visible at the modules interface:

Network Mode

Prepare Bus-Sleep Mode

Bus-Sleep Mode ⌋()

[SWS_UdpNm_00093]⌈ Changes of the AUTOSAR UdpNm operational modes shall be signalled to the upper layer by means of call-back functions. ⌋()

7.2.1 Network Mode

[SWS_UdpNm_00094]⌈ The Network Mode shall consist of three internal states:

Repeat Message State

Normal Operation State

Ready Sleep State ⌋()

[SWS_UdpNm_00095]⌈ When the Network Mode is entered from Bus-Sleep Mode or Prepare Bus-Sleep Mode, by default, the Repeat Message State shall be entered. ⌋()

[SWS_UdpNm_00096]⌈ When the Network Mode is entered, the NM-Timeout Timer shall be started. ⌋()

[SWS_UdpNm_00097]⌈ When the Network Mode is entered, the UdpNm shall notify the upper layer by calling `Nm_NetworkMode`. ⌋()

[SWS_UdpNm_00098]⌈ Upon successful reception of an NM PDU (call of `UdpNm_SoAdIfRxIndication`) in Network Mode, the NM-Timeout Timer shall be restarted. ⌋()

[SWS_UdpNm_00099]⌈ Upon transmission of an NM PDU (call of `UdpNm_SoAdIfTxConfirmation` with `E_OK`) in the Network Mode, the NM-Timeout Timer shall be restarted. ⌋()

Note: As no transmission confirmation is available from the SoAd or the TCP/IP stack it is assumed that each Network Management PDU transmission request results in a successful Network Management PDU transmission.

[SWS_UdpNm_00206]⌈ The NM-Timeout Timer shall be reset every time it is started or restarted. ⌋()

[SWS_UdpNm_00468] {DRAFT} ⌈ If function UdpNm_PnLearningRequest is called on a channel where `UdpNmDynamicPncToChannelMappingEnabled` is set to TRUE and UdpNm is in the Network Mode the UdpNm module shall set the Repeat Message Bit and the Partial Network Learning Bit in the CBV to 1 on this channel and change to or restart the Repeat Message State.⌋(RS_Nm_00153)

[SWS_UdpNm_00469] {DRAFT} ⌈ If the bits Partial Network Learning and Repeat Message Request both are received with value 1 on a channel where `UdpNmDynamicPncToChannelMappingEnabled` is set to TRUE and UdpNm is in the Network Mode the UdpNm module shall set the Partial Network Learning Bit in the CBV to 1 on this channel and change to or restart the Repeat Message State. ⌋()

Note: Restart in SWS_UdpNm_00468 or SWS_UdpNm_00469 means that UdpNm is already in Repeat Message State and then a complete re-entry of the Repeat Message State has to be performed once.

7.2.1.1 Repeat Message State

For nodes that are not in passive mode (refer to chapter 7.7.3) the Repeat Message State ensures, that any transition from Bus-Sleep or Prepare Bus-Sleep to the Network Mode becomes visible for the other nodes on the network. Additionally it ensures that any node stays active for a minimum amount of time (`UdpNmRepeatMessageTime`). Optionally it can be used for detection of present nodes.

[SWS_UdpNm_00100]⌈ When the Repeat Message State is entered from Bus-Sleep Mode, Prepare-Bus-Sleep Mode, Normal Operation State or Ready Sleep State transmission of NM packets shall be (re-) started unless passive mode is enabled. ⌋()

[SWS_UdpNm_00101]⌈ When the NM-Timeout Timer expires in the Repeat Message State, the NM-Timeout Timer shall be restarted. ⌋()

[SWS_UdpNm_00102]⌈ The NM shall stay in the Repeat Message State for a configurable amount of time determined by the `UdpNmRepeatMessageTime` (configuration parameter); after that time the Repeat Message State shall be left. ⌋()

[SWS_UdpNm_00103]⌈ When Repeat Message State is left, the Normal Operation State shall be entered, if the network has been requested (see SWS_UdpNm_00104). ⌋()

[SWS_UdpNm_00106]⌈ When Repeat Message State is left, the Ready Sleep State shall be entered, if the network has been released (see SWS_UdpNm_00105). ⌋()

[SWS_UdpNm_00107]⌈ If `UdpNmNodeDetectionEnabled` is set to `TRUE` UdpNm shall clear the Repeat Message Bit when leaving Repeat Message State. ⌋()

[SWS_UdpNm_00470] {DRAFT}
⌈ If `UdpNmDynamicPncToChannelMappingSupport` is set to TRUE UdpNm shall clear the Partial Network Learning Bit when leaving the Repeat Message State. ⌋()

7.2.1.2 Normal Operation State

The Normal Operation State ensures that any node can keep the NM-cluster awake as long as the network functionality is required.

[SWS_UdpNm_00116]⌈ When the Normal Operation State is entered from Ready Sleep State, transmission of NM PDUs shall be started unless passive mode is enabled or the NM message transmission ability has been disabled. ⌋()

[SWS_UdpNm_00117]⌈ When the NM-Timeout Timer expires in the Normal Operation State, the NM-Timeout Timer shall be restarted. ⌋()

[SWS_UdpNm_00118]⌈ When the network is released and the current state is Normal Operation State, the Normal Operation State shall be left and the Ready Sleep state shall be entered (refer to SWS_UdpNm_00105). ⌋()

[SWS_UdpNm_00119]⌈ If `UdpNmNodeDetectionEnabled` is set to `TRUE` and Repeat Message Request bit is received in the Normal Operation State, UdpNm shall enter Repeat Message State. ⌋(RS_Nm_00153)

[SWS_UdpNm_00120]⌈ If `UdpNmNodeDetectionEnabled` is set to `TRUE` and function `UdpNm_RepeatMessageRequest` is called in the Normal Operation State, UdpNm shall enter Repeat Message State. ⌋(RS_Nm_00153)

[SWS_UdpNm_00121]⌈ If `UdpNmNodeDetectionEnabled` is set to `TRUE` and function `UdpNm_RepeatMessageRequest` is called in the Normal Operation State, UdpNm shall set the Repeat Message Bit. ⌋(RS_Nm_00153)

7.2.1.3 Ready Sleep State

The Ready Sleep State ensures that any node in the NM-cluster waits with transition to the Prepare Bus-Sleep Mode as long as any other node keeps the NM-cluster awake.

[SWS_UdpNm_00108] ⌈ When the Ready Sleep State is entered from Repeat Message State or Normal Operation State, transmission of NM PDUs shall be stopped. ⌋()

Note: If passive mode is enabled no NM PDUs are transmited, no action is required. If passive mode is disabled, in some cases NM PDUs have to be transmitted in Ready Sleep State to grant a synchronized shutdown in the network, e.g. re-transmission of PN shutdown messages.

[SWS_UdpNm_00109] ⌈ When the NM-Timeout Timer expires in the Ready Sleep State, the Ready Sleep State shall be left and the Prepare Bus-Sleep Mode shall be entered. ⌋()

[SWS_UdpNm_00110] ⌈ When the network is requested and the current state is the Ready Sleep State, the Ready Sleep State shall be left and the Normal Operation State shall be entered (refer to SWS_UdpNm_00104). ⌋()

[SWS_UdpNm_00111] ⌈ If `UdpNmNodeDetectionEnabled` is set to `TRUE` and Repeat Message Request bit is received in the Ready Sleep State, UdpNm shall enter Repeat Message State. ⌋(RS_Nm_00153)

[SWS_UdpNm_00112] ⌈ If `UdpNmNodeDetectionEnabled` is set to `TRUE` and function `UdpNm_RepeatMessageRequest` is called in the Ready Sleep State, UdpNm shall enter Repeat Message State. ⌋(RS_Nm_00153)

[SWS_UdpNm_00113] ⌈ If `UdpNmNodeDetectionEnabled` is set to `TRUE` and function `UdpNm_RepeatMessageRequest` is called in the Ready Sleep State, UdpNm shall set the Repeat Message Bit. ⌋(RS_Nm_00153)

7.2.2 Prepare Bus-Sleep Mode

The purpose of the Prepare Bus Sleep state is to ensure that all nodes have time to stop their network activity before the Bus Sleep state is entered. Bus activity is calmed down (i.e. queued messages are transmitted in order to empty all Tx-buffers) and finally there is no activity on the bus in the Prepare Bus-Sleep Mode.

[SWS_UdpNm_00114]⌈ When Prepare Bus-Sleep Mode is entered, the UdpNm shall notify the upper layer by calling `Nm_PrepareBusSleepMode`. ⌋()

[SWS_UdpNm_00115]⌈ If `UdpNmStayInPbsEnabled` is disabled UdpNm shall stay in the Prepare Bus-Sleep Mode for a configurable amount of time determined by the `UdpNmWaitBusSleepTime` (configuration parameter); after that time the Prepare Bus-Sleep Mode shall be left and the Bus-Sleep Mode shall be entered.⌋()

Note: This requirement implicitly contains that if `UdpNmStayInPbsEnabled` is enabled UdpNm will never be left due to a timeout, i.e. UdpNm will stay in Prepare Bus-Sleep Mode until either ECU goes to Power Off or any restart reason applies (e.g. see following requirements).

[SWS_UdpNm_00124]⌈ Upon successful reception of an NM PDU in the Prepare Bus-Sleep Mode, the Prepare Bus-Sleep Mode shall be left and the Network Mode shall be entered; by default the Repeat Message State is entered (refer to SWS_UdpNm_00095).⌋()

[SWS_UdpNm_00123]⌈When the network is requested in the Prepare Bus-Sleep Mode, the Prepare Bus-Sleep Mode shall be left and the Network Mode shall be entered; by default the Repeat Message State is entered (refer to SWS_UdpNm_00095)⌋()

[SWS_UdpNm_00122]⌈When the network has been requested (see SWS_UdpNm_00104) in the Prepare Bus-Sleep Mode and the UdpNm module has entered Network Mode and if `UdpNmImmediateRestartEnabled` (configuration parameter) is `TRUE`, the UdpNm module shall transmit a Network Management PDU.⌋()

Rationale: Other nodes in the cluster are still in Prepare Bus-Sleep Mode; in the exceptional situation described above transition into the Bus-Sleep Mode shall be avoided and bus-communication shall be restored as fast as possible.

Caused by the transmission offset for Network Management PDUs in UdpNm, the transmission of the first Network Management PDU in Repeat Message State can be delayed significantly. In order to avoid a delayed re-start of the network the transmission of a Network Management PDU can be requested immediately.

Note: If `UdpNmImmediateRestartEnabled` is `TRUE` and a wake-up line is used, a burst of Network Management PDUs occurs if all network nodes get a network request in Prepare Bus-Sleep Mode.

7.2.3 Bus-Sleep Mode

The purpose of the Bus-Sleep state is to reduce power consumption in the node, when no messages are to be exchanged.

The communication controller is switched to sleep mode, respective wakeup mechanisms are activated and finally power consumption is reduced to the adequate level in the Bus-Sleep Mode.

If `UdpNmStayInPbsEnabled` is disabled and configurable amount of time determined by the `UdpNmTimeoutTime + UdpNmWaitBusSleepTime` (both configuration parameters) is identically configured for all nodes in the network management cluster, all nodes in the network management cluster that are coordinated with use of the AUTOSAR NM algorithm perform the transition into the Bus-Sleep Mode at approximately the same time.

Note: The parameters `UdpNmTimeoutTime` and `UdpNmWaitBusSleepTime` should have the same values within all network nodes of the NM-cluster.
Depending on the specific implementation, transition into the Bus-Sleep Mode takes place approximately at the same time. The time jitter experienced for this transition depends on the following factors:

internal clock precision (oscillator’s drift),

NM-task cycle time (if tasks are not synchronized with a global time),

NM PDUs waiting time in the Tx-queue (if transmission confirmation is made immediately after transmit request).

For a best case estimation only oscillator drift should be taken into account for a configurable amount of time determined by the value `UdpNmTimeoutTime + UdpNmWaitBusSleepTime` (both configuration parameters).

[SWS_UdpNm_00126] ⌈ When Bus-Sleep Mode is entered, the UdpNm shall notify the upper layer by calling `Nm_BusSleepMode`; this shall not be the case if Bus-Sleep Mode is entered by default at initialization. ⌋()

[SWS_UdpNm_00127] ⌈ When the UdpNm module receives successfully Network Management PDU in the Bus-Sleep Mode (call of `UdpNm_SoAdIfRxIndication`), the UdpNm module shall notify the upper layer by calling the callback function `Nm_NetworkStartIndication`. ⌋()

Rationale: To avoid race conditions and state inconsistencys between Network and Mode Management, UdpNm will not automatically perform the transition from Bus-Sleep Mode to Network Mode. UdpNm will only inform the upper layers which have to make the wake-up decision. NM packet reception in Bus-Sleep Mode must be handled depending on the current state of the ECU shutdown or startup process.

[SWS_UdpNm_00128] ⌈ If `UdpNm_PassiveStartUp` is called in the Bus-Sleep Mode or Prepare Bus Sleep Mode, the UdpNm module shall enter the Network Mode; by default the Repeat Message State is entered (refer to SWS_UdpNm_00095 and SWS_UdpNm_00104). ⌋()

Note: In the Prepare Bus-Sleep Mode and Bus-Sleep Mode is assumed that the network is released, unless bus communication is explicitly requested.

[SWS_UdpNm_00129]: ⌈When the network is requested in Bus-Sleep Mode, the UdpNm module shall enter the Network Mode; by default the UdpNm module shall enter the Repeat Message State (refer to SWS_UdpNm_00095 and SWS_UdpNm_00104).⌋()

7.3 Network states

Network states (i.e. 'requested' and 'released') are two additional states of the AUTOSAR UdpNm state machine that exist in parallel to the state machine. Network states denote, whether the software components need to communicate on the bus (the network state is then 'requested'); or whether the software components don't have to communicate on the bus (the bus network state is then 'released'); note that if the network is released an ECU may still communicate because some other ECU still request the network.

[SWS_UdpNm_00104] ⌈The function call `UdpNm_NetworkRequest` shall request the network. I.e. the UdpNm module shall change network state to 'requested'.⌋()

[SWS_UdpNm_00105] ⌈ The function call `UdpNm_NetworkRelease` shall release the network. I.e. the UdpNm module shall change network state to 'released'. ⌋()

7.4 Initialization

[SWS_UdpNm_00141] ⌈ After successful initialization the Network Management state shall be set to BusSleep Mode. ⌋()

Note: The UdpNm module should be initialized after SoAd is initialized and before any other network management service is called.

[SWS_UdpNm_00143] ⌈ When initialized, by default, the UdpNm module shall set the network state to 'released'. ⌋()

[SWS_UdpNm_00144] ⌈ When initialized, by default, the UdpNm module shall enter the Bus-Sleep Mode. ⌋()

[SWS_UdpNm_00145] ⌈If AUTOSAR UdpNm is not initialized it shall not prohibit bus traffic. ⌋()

[SWS_UdpNm_00060] ⌈The function `UdpNm_Init` shall select the active configuration set by means of a configuration pointer parameter being passed (see 8.3.1).
⌋()

[SWS_UdpNm_00033] ⌈After initialization the transmission of NM messages shall be stopped.
⌋()

[SWS_UdpNm_00025] ⌈After initialization each byte of the user data bytes shall be set to 0xFF. ⌋()

[SWS_UdpNm_00085] ⌈After initialization the Control Bit Vector shall be set to `0x00`.
⌋()

[SWS_UdpNm_00485] {DRAFT} ⌈During initialization and if `UdpNmPnEnabled` is `TRUE`, the UdpNm module shall set each byte of the PNC bit vector to `0x00`.⌋()

[SWS_UdpNm_00148] ⌈ All instances of UDP NM on different ECUs in one NM cluster shall use the same UDP receive port.⌋()

7.5 Execution

7.5.1 Processor architecture

[SWS_UdpNm_00146] ⌈ The AUTOSAR UdpNm coordination algorithm shall be processor independent, meaning it shall not rely on any processor specific hardware support and thus shall be realizable on any processor architecture that is within the scope of AUTOSAR. ⌋()

7.5.2 Timing parameters

[SWS_UdpNm_00246]⌈ The configuration parameter `UdpNmTimeoutTime` shall determine the AUTOSAR UdpNm timing parameter NM-Timeout Time. ⌋()

[SWS_UdpNm_00247]⌈ The configuration parameter `UdpNmRepeatMessageTime` shall determine the AUTOSAR UdpNm timing parameter Repeat Message Time. ⌋()

[SWS_UdpNm_00248]⌈ The configuration parameter `UdpNmWaitBusSleepTime` shall determine the AUTOSAR UdpNm timing parameter Wait Bus-Sleep Time. ⌋()

[SWS_UdpNm_00249]⌈ The optional configuration parameter `UdpNmRemoteSleepIndTime` shall determine the AUTOSAR UdpNm timing parameter Remote Sleep Indication Time. ⌋()

7.6 Communication Scheduling

7.6.1 NM Message Transmission

Note: The transmission mechanisms described in this chapter are only relevant if the NM message transmission ability is enabled.

[SWS_UdpNm_00072] ⌈ The transmission of NM messages shall be configurable by means of `UdpNmPassiveModeEnabled` (see chapter 10.2). ⌋()

Note: Passive nodes do not transmit NM messages, i.e. they can not actively influence the shut down decision, but they do receive NM message in order to be able to shut down synchronously.

Note: The transmission mechanisms described in this chapter are only relevant if `UdpNmPassiveModeEnabled` is `FALSE`.

[SWS_UdpNm_00237]⌈ The UdpNm module shall provide the periodic transmission mode. In this transmission mode the UdpNm module shall send Network Management PDUs periodically. ⌋()

Note: The periodic transmission mode is used in the "Repeat Message State" and "Normal Operation State".

[SWS_UdpNm_00005]⌈If the Repeat Message State is not entered via UdpNm_NetworkRequest OR UdpNmImmediateNmTransmissions is zero the transmission of NM PDU shall be delayed by `UdpNmMsgCycleOffset` after entering the repeat message state. ⌋()

Note: This requirement covers also the case if Repeat Message State is entered from Network Operation State or Ready Sleep State due to Repeat Message Request or Bit (see SWS_UdpNm_00111, SWS_UdpNm_00112, SWS_UdpNm_00119, SWS_UdpNm_00120). This means that in this case the immediate transmission is not used (even if UdpNmImmediateNmTransmissions > 0 and independent from configuration of UdpNmPnHandleMultipleNetworkRequests) i.e. UdpNmMsgCycleOffset will always be applied. This mechanism prevents bursts of NM messages.

[SWS_UdpNm_00334] ⌈When entering the Repeat Message State from Bus Sleep Mode or Prepare Bus Sleep Mode because of `UdpNm_NetworkRequest()` (active wakeup) and if `UdpNmImmediateNmTransmissions` is greater zero, the NM PDUs shall be transmitted using `UdpNmImmediateNmCycleTime` as cycle time. The transmission of the first NM PDU shall be triggered as soon as possible. After the transmission the Message Cycle Timer shall be reloaded with `UdpNmImmediateNmCycleTime`. The `UdpNmMsgCycleOffset` shall not be applied in this case. ⌋()

[SWS_UdpNm_00006] ⌈If Normal Operation State is entered from Ready Sleep State the transmission of NM PDUs shall be started immediately. ⌋()

[SWS_UdpNm_00454] ⌈If `UdpNmPnHandleMultipleNetworkRequests` is set to TRUE `UdpNm_NetworkRequest` shall trigger a state transition from Network Mode to Repeat Message state. If PDU transmission ability is enabled the NM PDUs shall be transmitted using `UdpNmImmediateNmCycleTime` as cycle time. The transmission of the first NM PDU shall be triggered as soon as possible. After the transmission the Message Cycle Timer shall be reloaded with `UdpNmImmediateNmCycleTime`. The `UdpNmMsgCycleOffset` shall not be applied in this case. ⌋()

Note: `UdpNmImmediateNmTransmissions` has to be greater zero in this case due to ECUC_UdpNm_00075.

[SWS_UdpNm_00330] ⌈If NM PDUs shall be transmitted with `UdpNmImmediateNmCycleTime` (See SWS_UdpNm_00334 and SWS_UdpNm_00454), UdpNm shall ensure that `UdpNmImmediateNmTransmissions` (including first immediate transmission) with this timing are requested successfully. If a transmission request to SoAd fails (`E_NOT_OK` is returned), UdpNm shall retry the transmission request in the next main function. Afterwards UdpNm shall continue transmitting NM PDUs using the `UdpNmMsgCycleTime`. ⌋()

Note: While transmitting NM PDUs using the `UdpNmImmediateNmCycleTime` no other Nm PDUs shall be transmitted (i.e. the `UdpNmMsgCycleTime` transmission cycle is stopped).

[SWS_UdpNm_00032] {DRAFT} ⌈ If transmission of NM PDUs has been started, the UdpNm Message Cycle Timer expires and when `UdpNmSynchronizedPncShutdownEnabled` is set to either `FALSE` or if set to `TRUE` and no request for a synchronized PNC shutdown is pending, then the UdpNm modul shall transmit an NM PDU by calling `SoAd_IfTransmit`. ⌋()

[SWS_UdpNm_00472] {DRAFT} ⌈ If transmission of NM PDUs has been started, the UdpNm Message Cycle Timer expires and `UdpNmSynchronizedPncShutdownEnabled` is set to `TRUE` and requests for synchronized PNC shutdown are pending, the transmission of the NM PDU shall be postponed to the next corresponding UdpNm_Mainfunction_<Instance Id> call. ⌋()

Note:

- The synchronized PNC shutdown has to be sent immediately and therefore processing of cylic NM messages transmitted with `UdpNmMsgCycleTime` has to be delayed. In rare cases this could lead to a delay of more than one main function cycle time.
- The NM timing has to consider that an NM message transmitted with `UdpNmMsgCycleTime` may be delayed for more than one main function cycle time. Therefore the following condition has to be fulfilled to tolerate multiple delays of those NM Messages:
  `(NmPnResetTime - UdpNmMsgCycleTime) > n * UdpNmMainFunctionPeriod`, where n denotes the number of tolerated delays before the PnResetTime expires, if no NM message is received.

[SWS_UdpNm_00040] ⌈ If the UdpNm Message Cycle Timer expires it shall be restarted with `UdpNmMsgCycleTime`. ⌋()

[SWS_UdpNm_00051] ⌈ If transmission of NM PDUs has been stopped the UdpNm Message Cycle Timer shall be canceled. ⌋()

[SWS_UdpNm_00007] ⌈ If parameter `UdpNmRetryFirstMessageRequest` (see ECUC_UdpNm_00085) is TRUE and if the first transmit request after transition from Bus Sleep to Repeat Message State is not accepted by SoAd, the message request shall be repeated in the next main function until one transmit request is accepted by SoAd. ⌋()

Note: This feature can be used in case of partial network wakeup filter to avoid a blocking of all messages in case of passive start-up and first message request is not accepted by SoAd due to EthSM could not enable transmission path fast enough (e.g. in case of asynchronous transceiver handling).

[SWS_UdpNm_00379] ⌈ If `UdpNm_SoAdIfTxConfirmation` is called with result `E_NOT_OK`, UdpNm shall call the function `Nm_TxTimeoutException`. ⌋ (RS_Nm_00137)

7.6.2 NM Message Reception

If an NM message has been successfully received, the SoAd will call `UdpNm_SoAdIfRxIndication`.

[SWS_UdpNm_00035] ⌈ Upon a call of `UdpNm_SoAdIfRxIndication`, the UdpNm module shall copy the data of the Network Management PDU referenced in the function parameter to an internal buffer. ⌋()

[SWS_UdpNm_00037] ⌈ When an NM PDU has been received, the Nm function `Nm_PduRxIndication` shall be called, if `UdpNmPduRXIndicationEnabled` (configuration parameter) is `TRUE`. ⌋()

7.7 Additional features

7.7.1 Detection of Remote Sleep Indication (optional)

The “Remote Sleep Indication” denotes a situation, where a node in Normal Operation State finds all other nodes in the cluster are ready to sleep. The node still in Normal Operation State will still keep the bus awake.

[SWS_UdpNm_00149] ⌈ Detection of remote sleep indication shall be statically configurable with use of the `UdpNmRemoteSleepIndEnabled` switch (configuration parameter). ⌋()

[SWS_UdpNm_00150] ⌈ If no NM PDUs are received in the Normal Operation State for a configurable amount of time determined by the `UdpNmRemoteSleepIndTime` (configuration parameter), the NM shall notify the Generic Network Management Interface that all other nodes in the cluster are ready to sleep (the so-called ‘Remote Sleep Indication’) by calling `Nm_RemoteSleepIndication`. ⌋()

[SWS_UdpNm_00151] ⌈ If Remote Sleep Indication has been previously detected and if an NM PDU is received in the Normal Operation State or Ready Sleep State again, the NM shall notify the Generic Network Management Interface that some nodes in the cluster are not ready to sleep anymore (the so-called ‘Remote Sleep Cancellation’) by calling `Nm_RemoteSleepCancellation`. ⌋()

[SWS_UdpNm_00152]⌈ If Remote Sleep Indication has been previously detected and if Repeat Message State is entered from Normal Operation State or Ready Sleep State, the UdpNm shall notify the Generic Network Management Interface that some nodes in the cluster are not ready to sleep anymore (the so-called 'Remote Sleep Cancellation') by calling `Nm_RemoteSleepCancellation`. ⌋()

[SWS_UdpNm_00154]⌈ The NM shall reject a check of Remote Sleep Indication in Bus-Sleep Mode, Prepare Bus-Sleep Mode and Repeat Message State; the service shall not be executed and `E_NOT_OK` shall be returned. ⌋()

7.7.2 User Data (optional)

[SWS_UdpNm_00158]⌈ Support of NM user data shall be statically configurable using the `UdpNmUserDataEnabled` switch (configuration parameter). ⌋()

[SWS_UdpNm_00159]⌈ When `UdpNm_SetUserData` is called, the NM user data for NM packets transmitted next on the bus shall be set; operation of setting the NM user data shall guarantee data consistency. ⌋()

[SWS_UdpNm_00160]⌈ When `UdpNm_GetUserData` is called, the NM user data contained in the payload of the most recently received NM PDU shall be provided; operation of providing the NM user data shall guarantee data consistency. ⌋()

Note: If NM user data is configured it will be sent for sure in the Repeat Message State. In Ready Sleep State the user data will not be sent.

[SWS_UdpNm_00312]⌈ If `UdpNmComUserDataSupport` is enabled the API `UdpNm_SetUserData` shall not be available. ⌋()

[SWS_UdpNm_00317]⌈ If `UdpNmComUserDataSupport` is enabled and NM-PDU is not configured for triggered transmission in SoAd (`SoAdBswModules/SoAdIfTriggerTransmit` = `FALSE`), the UdpNm shall collect the NM User Data from the referenced NM I-PDU by calling `PduR_UdpNmTriggerTransmit` and combine the user data with the further NM bytes each time before it requests the transmission of the corresponding NM message. ⌋(RS_Nm_02503)

Note: In case of triggered transmission no data is needed at the transmission request, just the length is needed. The data will be collected within `UdpNm_SoAdIfTriggerTransmit` (see chapter 8.4.3 UdpNm_SoAdIfTriggerTransmit).

[SWS_UdpNm_00464]⌈ If `UdpNmComUserDataSupport` is enabled and if UdpNm is in RepeatMessage state or NormalOperation state and if `UdpNm_Transmit` is called, UdpNm shall request an additional transmission of the NM PDU with the current data. ⌋(RS_Nm_02503)

Note: The call of `UdpNm_Transmit` request to transmit a NM PDU between the periodic transmissions with the current data (e.g. system bytes, user data and PNC bit vector)

7.7.3 Passive Mode (optional)

In Passive Mode the node is only receiving NM messages but not transmitting any NM messages.

[SWS_UdpNm_00161]⌈ Passive Mode shall be statically configurable with use of the `UdpNmPassiveModeEnabled` switch (configuration parameter). ⌋()

[SWS_UdpNm_00162]⌈ Passive Mode shall be statically configured consistent for all instances within one ECU. ⌋()

[SWS_UdpNm_00163] ⌈ If Passive Mode is used (configuration parameter `UdpNmPassiveModeEnabled`) the following options must not be used:

Bus Synchronization
(configuration parameter `UdpNmBusSynchronizationEnabled`)

Remote Sleep Indication
(configuration parameter `UdpNmRemoteSleepIndEnabled`)

Node Detection
(configuration parameter `UdpNmNodeDetectionEnabled`) ⌋()

7.7.4 State change notification (optional)

[SWS_UdpNm_00166]⌈ All changes of the AUTOSAR UdpNm states shall be notified to the upper layer by calling `Nm_StateChangeNotification` if the callback `Nm_StateChangeNotification` is enabled (configuration parameter `UdpNmStateChangeIndEnabled` is `TRUE`). ⌋()

7.7.5 Communication Control (optional)

[SWS_UdpNm_00168]⌈ Communication Control shall be statically configurable with use of the `UdpNmComControlEnabled` switch (configuration parameter). ⌋()

[SWS_UdpNm_00170] ⌈ The optional service `UdpNm_DisableCommunication` shall disable the NM PDU transmission ability. ⌋()

Note: The NM coordination algorithm cannot work correctly if NM PDU transmission ability is disabled. Therefore it has to be ensured that the ECU is not shutdown as long as the NM PDU transmission ability is disabled.

If `UdpNm_NetworkRelease` is called and NM PDU transmission ability has been disabled, ECU will shut down. This ensures that ECU can shut down also in case of race conditions (e.g. diagnostic session left shortly before enabling communication) or a wrong usage of communication control.

[SWS_UdpNm_00172] ⌈ The optional service `UdpNm_DisableCommunication` shall return `E_NOT_OK`, if the current mode is not Network Mode. ⌋()

[SWS_UdpNm_00173] ⌈ When the Network Management PDU transmission ability is disabled, the UdpNm module shall stop the UdpNm Message Cycle Timer in order to stop the transmission of Network Management PDUs. ⌋()

[SWS_UdpNm_00174] ⌈ When the NM PDU transmission ability is disabled, the NM-Timeout Timer shall be stopped. ⌋()

[SWS_UdpNm_00175] ⌈ When the NM PDU transmission ability is disabled, the detection of Remote Sleep Indication Timer shall be suspended. ⌋()

[SWS_UdpNm_00178] ⌈ When the Network Management PDU transmission ability is enabled, the transmission of NM PDUs shall be started latest within the next NM main function. ⌋(RS_Nm_02512)

[SWS_UdpNm_00179] ⌈ When the NM PDU transmission ability is enabled, the NM-Timeout Timer shall be restarted. ⌋()

[SWS_UdpNm_00180] ⌈ When the NM PDU transmission ability is enabled, the detection of Remote Sleep Indication Timer shall be resumed. ⌋()

[SWS_UdpNm_00181] ⌈ The optional service `UdpNm_RequestBusSynchronization` shall return `E_NOT_OK` if the NM PDU transmission ability is disabled. ⌋()

7.7.6 NM Coordinator synchronization support (optional)

When having more than one coordinator connected to the same bus a special bit in the CBV, the `NmCoordinatorSleepReady` bit is used to indicate that the main coordinator requests to start shutdown sequence. The main functionality of the algorithm is described in the Nm module.

[SWS_UdpNm_00320] ⌈If the UdpNm called `NM_CoordReadyToSleepIndication` and is still in Network Mode it shall notify the Nm by calling `Nm_CoordReadyToSleepCancellation` on the first reception of a NM message with the `NmCoordinatorSleepReady` bit (see CBV) set it to 0
⌋()

[SWS_UdpNm_00364] ⌈If UdpNm has entered Network mode or called `Nm_CoordReadyToSleepCancellation` before it shall notify the NM by calling `Nm_CoordReadyToSleepIndication` on the first reception of NM message with the `NmCoordinatorSleepReady` bit (see CBV) set to 1
⌋()

[SWS_UdpNm_00321] ⌈If `UdpNmCoodinatorSyncSupport` is set to `TRUE` and the API `UdpNm_SetSleepReadyBit` is called UdpNm shall set the "NM Coordinator Sleep Ready Bit" bit to passed value and trigger a single Network Management PDU.⌋()

[SWS_UdpNm_00322] ⌈The API `UdpNm_SetSleepReadyBit()` and the feature "Coordinated Bus Shutdown" shall only be available if `UdpNmCoordinatorSyncSupport` is set to `TRUE`.⌋()

7.8 Partial Networking

7.8.1 Rx Handling of NM PDUs

[SWS_UdpNm_00328]⌈ If the `UdpNmPnEnabled` is `FALSE`, the UdpNm shall perform the normal Rx Indication handling and the partial networking extensions shall be disabled.⌋()

[SWS_UdpNm_00329] ⌈If `UdpNmPnEnabled` is `TRUE`, the PNI bit in the received NM-PDU is 0 and `UdpNmAllNmMessagesKeepAwake` is `TRUE`, the UdpNm module shall perform the normal Rx Indication handling and omitting the extensions for partial networking.⌋()

[SWS_UdpNm_00462]⌈ If `UdpNmPnEnabled` is `TRUE`, the PNI bit in the received NM-PDU is 0 and `UdpNmAllNmMessagesKeepAwake` is `FALSE`, the UdpNm module shall ignore the received NM-PDU. ⌋()

[SWS_UdpNm_00331] {OBSOLETE replaced by SWS_UdpNm_00486 } ⌈If `UdpNmPnEnabled` is `TRUE` and the PNI bit in the received NM-PDU is 1, UdpNm module shall process the Partial Networking Information of the NM-PDU as described in chapter 7.8.3 to 7.8.6.⌋()

[SWS_UdpNm_00486] {DRAFT} ⌈ If `UdpNmPnEnabled` is set to `TRUE`, the PNI bit in the received NM-PDU is set to 1 and the PNSR bit is set to 0, UdpNm module shall extract the PNC bit vector from the received NM-PDU according to the partial network configuration (`NmPncBitVectorOffset` and `NmPncBitVectorLength` of the corresponding NM-channel) and forward the PNC bit vector by calling `Nm_PncBitVectorRxIndication`⌋ (RS_Nm_02546, RS_Nm_02519, RS_Nm_02547)

[SWS_UdpNm_00487] {DRAFT} ⌈If `UdpNmPnEnabled` is set to `TRUE` and `Nm_PncBitVectorRxIndication` was called, then a received NM PDU shall only be considered for further processing under the following conditions:

- UdpNmAllNmMessagesKeepAwake is set to `TRUE` OR
- the output value of RelevantPncRequestDetectedPtr is set to `TRUE`

⌋(RS_Nm_02546, RS_Nm_02527)

Note:

- UdpNmAllNmMessagesKeepAwake is required to enable a gateway to stay awake on any kind of NM-PDU.
- As consequence of SWS_UdpNm_00487, a NM PDU is not considered for further processing if not all messages shall keep the ECU awake or no relevant PN information was detected.

Example:

- `UdpNmPduCbvPosition = 0`
- `UdpNmPduNidPosition = 1`
- `NmPncBitVectorOffset = 4`
- `NmPncBitVectorLength = 4`
- Calculated length of user data range `= 2`

Byte 2 and Byte 3 of the NM PDU contain user data and
Byte 4 to Byte 7 of the NM PDU contain the PNC bit vector:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
CBV	NID	User Data		PNC bit vector			
0x40	0x00	0xFF	0xFF	0x12	0x8E	0x80	0x01

Figure 7-1 Example NM PDU containing PN information

For this example four NmPnFilterMaskBytes shall be defined. The values of the PN filter mask are used according to the partial network design e.g:

- NmPnFilterMaskByteIndex = 0 with NmPnFilterMaskByteValue = 0x01
- NmPnFilterMaskByteIndex = 1 with NmPnFilterMaskByteValue = 0x97
- NmPnFilterMaskByteIndex = 2 with NmPnFilterMaskByteValue = 0x00
- NmPnFilterMaskByteIndex = 3 with NmPnFilterMaskByteValue = 0x00

Note: The offset for the PNC bit vector is derived from the Nm module (`NmPncBitVectorOffset`). The PNC bit vector length is derived form the Nm module per NM-channel (`NmPncBitVectorLength`). The PN filter mask (`NmPnFilterMaskByteIndex` and `NmPnFilterMaskByteValue`) located and used in the Nm module.

[SWS_UdpNm_00473] {DRAFT} ⌈ If `UdpNmSynchronizedPncShutdownEnabled` is `TRUE`, the PNI bit in the received NM-PDU is 1, the PNSR bit in the received NM-PDU is 1 and the corresponding ComMChannel configured via `UdpNmComMNetworkHandleRef` where this NM-PDU was received is actively coordinated (ComMPncGatewayType set to COMM_GATEWAY_TYPE_ACTIVE), then the UdpNm module shall ignore the received NM-PDU. Additionally, the UdpNm module shall:

- report the runtime error `UDPNM_E_INVALID_PN_SYNC_SHUTDOWN_REQUEST` to the Default Error Tracer
- request a transmission of a NM-PDU with the current PN information lastet in the next main function call of the affected UdpNm-Channel, If `UdpNmPnSyncShutdownErrorReactionEnabled` is set to `TRUE`

⌋()

[SWS_UdpNm_00488] {DRAFT} ⌈ If `UdpNmSynchronizedPncShutdownEnabled` is `TRUE`, the PNI bit in the received NM-PDU is set to 1 and the PNSR bit is set to 1, UdpNm module shall extract the PNC bit vector from the received NM-PDU according to the partial network configuration (`NmPncBitVectorOffset` and `NmPncBitVectorLength` of the corresponding NM-channel) and forward the PNC bit vector by calling `Nm_ForwardSynchronizedPncShutdown`. ⌋(RS_Nm_02541)

Note: PNSR Bit set to 1 is only possible if a synchronized PNC shutdown is requested. A synchronized PNC shutdown should be handled across the PN topology. Therefore, it is assumed that either all coordinators have the synchronized PNC shutdown enabled or all coordinators have the synchronized PNC shutdown disabled. A mixture of both would lead to an unsynchronized PNC shutdown, which has to be avoided.

7.8.2 Tx Handling of NM PDUs

[SWS_UdpNm_00332]⌈If `UdpNmPnEnabled` is `TRUE` the UdpNm module shall set the value of the transmitted PNI bit in the CBV to 1.⌋()

Note: The usage of the CBV is mandatory in case Partial Networking is used.

[SWS_UdpNm_00333] ⌈If `UdpNmPnEnabled` is `FALSE` the UdpNm module shall set the value of the transmitted PNI bit in the CBV always to 0.⌋()

[SWS_UdpNm_00489] {DRAFT} ⌈ If `UdpNmPnEnabled` is TRUE, NM-PDU is not configured for triggered transmission in SoAd (`SoAdBswModules/SoAdIfTriggerTransmit` set to `FALSE`), no requests for synchronized PNC shutdown are pending and a NM-PDU has to be transmitted, the UdpNm module shall perform the following actions in the given order:

- Call `Nm_PncBitVectorTxIndication`(<NM-channel>, <buffer to store the unfiltered PNC bit vector of aggregated internal PNC requests>) to indicate the transmission request and to retrieve internal PNC requests.
- Copy the received PNC bit vector for internal PNC requests to the NM-PDU by considering `NmPncBitVectorOffset` and `NmPncBitVectorLength` of the corresponding NM-channel
- If user data is enabled, fetch the available data (either from Com if `UdpNmComUserDataSupport` is enabled or from internal storage) and copy the data in the user data range of the NM-PDU.
- Trigger transmission of the NM-PDU by calling SoAd_IfTransmit

⌋(RS_Nm_02546, RS_Nm_02519, RS_Nm_02547)

[SWS_UdpNm_00474] {DRAFT} ⌈ If `UdpNmSynchronizedPncShutdownEnabled` is set to `TRUE` and the UdpNm module is indicated via `UdpNm_RequestSynchronizedPncShutdown`, the UdpNm module shall store the given PNC (pncId) per given UdpNm-Channel (nmChannelHandle) as pending request for a synchronized PNC shutdown ⌋(RS_Nm_02545)

Note: The aggregation of all PNCs which are requested for a synchronized PNC shutdown and the transmission as PN shutdown message (set the PNSR bit in the CBV to 1) is done asynchronously in the context of the corresponding UdpNm_Mainfunction.

[SWS_UdpNm_00475] { OBSOLETE replaced by SWS_UdpNm_00490 } ⌈ If `UdpNmSynchronizedPncShutdownEnabled` is set to `TRUE`, requests for synchronized PNC shutdown are pending and no transmission confirmation (indicated via `UdpNm_TxConfirmation`) of a previous call is pending, then the UdpNm module shall request in the next main function call of the corresponding NM-channel a transmission of a NM message by calling `SoAd_IfTransmit`. In case the NM-PDU is not configured for a triggered transmission in the SoAd (`SoAdBswModules/SoAdIfTriggerTransmit = FALSE`), UdpNm shall set for this message additionally the following data beneath the normal data:

- Set the `PNSR` bit in the CBV to 1
- Overwrite the PN information in the user data (after NM User Data has been fetched, if `UdpNmComUserDataSupport` is enabled) by setting bits that corresponds to PNC IDs stored as pending request for a synchronized PNC shutdown to 1 and all other bits to 0.

⌋(RS_Nm_02540, RS_Nm_02542, RS_Nm_02545)

[SWS_UdpNm_00490] {DRAFT} ⌈ If UdpNmSynchronizedPncShutdownEnabled is set to TRUE, requests for synchronized PNC shutdown are pending and no transmission confirmation (indicated via UdpNm_TxConfirmation) of a previous call is pending, then the UdpNm module shall request in the next main function call a transmission of a NM-PDU as PN shutdown message by calling SoAd_IfTransmit. In case the NM-PDU is not configured for triggered transmission in SoAd (SoAdBswModules/SoAdIfTriggerTransmit = FALSE), UdpNm shall set for this message additionally the following data beneath the normal data:

- Set the PNSR bit in the CBV to 1
- If user data is enabled, fetch the available data (either from Com if UdpNmComUserDataSupport is enabled or from internal storage) and copy the data in the user data range of the NM-PDU
- Write the PNC bit vector with respect to NmPncBitVectorOffset and NmPncBitVectorLength of the corresponding NM-channel by setting bits that corresponds to PNC IDs stored as pending request for a synchronized PNC shutdown to 1 and all other bits to 0

⌋(RS_Nm_02540, RS_Nm_02542, RS_Nm_02545)

Note: The UdpNm modul has to aggregate all PNCs which were indicated for a synchronized PNC shutdown and transfer the pncId's to a byte array (PNC bit vector). Each bit (PNC bit) of the PNC bit vector represent a particular PNC. The byteIndex and bitindex within the PNC bit vector of PNC bit shall be determined as follows:

- byteIndex = (PncId div 8) - NmPncBitVectorOffset
- bitIndex = (PncId mod 8)

[SWS_UdpNm_00481]⌈ If UdpNmPnShutdownMessageRetransmissionDuration is configured and transmission of a PN shutdown message is requested (refer to SWS_UdpNm_00475) for the first time, then the corresponding retransmission timer for PN shutdown messages shall be started with UdpNmPnShutdownMessageRetransmissionDuration on all affected NM-channels.⌋()

[SWS_UdpNm_00476] {DRAFT} ⌈If UdpNmSynchronizedPncShutdownEnabled is set to TRUE, the UdpNm module has requested a transmission of a NM-PDU as PN shutdown message (see SWS_UdpNm_00490) and UdpNm_TxConfirmation is called with result E_OK, the UdpNm shall consider those PNC IDs stored as pending request for a synchronized PNC shutdown of the corresponding NM-channel as completed and remove them from storage. Additionally, if UdpNmPnShutdownMessageRetransmissionDuration is configured, then UdpNm shall cancel the retransmission timer for PN shutdown messages of the affected NM-channel.⌋(RS_Nm_02545)

Note: UdpNm has to ensure that new request for a synchronized PNC shutdown (indicated via UdpNm_RequestSynchronizedPncShutdown) are not lost, during an on going transmission of a PN shutdown NM frame.

[SWS_UdpNm_00477] {DRAFT} ⌈ If UdpNmSynchronizedPncShutdownEnabled is set to TRUE, UdpNmPnShutdownMessageRetransmissionDuration is configured, the UdpNm module has requested a transmission due to synchronized PNC shutdown (see SWS_UdpNm_00475), UdpNm_SoAdIfTxConfirmation is called with result E_NOT_OK or or the transmissions request for this PN shutdown message was not accepted (SoAd_IfTransmit returned E_NOT_OK), then the UdpNm module shall keep those PNC IDs stored as pending request for a synchronized PNC shutdown and perform a retransmission in the next main function. ⌋(RS_Nm_02545)

Note:

- UdpNm has to perform a retry transmission handling for PN shutdown messages in the context of the corresponding main function calls, if transmission of the PN shutdown message was not confirmed by the lower layer (either with E_NOT_OK or UdpNm_SoAdIfTxConfirmation was not called).The retry transmission requests should cover error cases, were the lower layer cannot transmit the Nm messages. In the worst case this collide with a post poned NM message transmitted with UdpNmMsgCycleTime (see SWS_UdpNm_00472). But in any case, if the capability to transmitted NM messages is not re-covered within the PN reset time (EIRA), the PNCs will shutdown not synchronized, which might lead to timeout errors on application level.
- The dependency to a pending transmission confirmation indicated by the lower layer, should support reliable communication, e.g. ensure PN shutdown message was transmitted on the network or avoid transmissions of outdated PN shutdown messages, if for example queueing in the lower layer is configured.

[SWS_UdpNm_00482]⌈ If UdpNmSynchronizedPncShutdownEnabled is set to TRUE and the UdpNm module has stored PNC IDs as pending request for a synchronized PNC shutdown, then the UdpNm shall remove those PNC IDs from storage which are either externally or internally requested again:

- UdpNm shall check on reception of an NM-message, if externally requested PNCs are received
- UdpNm shall check up front to each transmission of an PN shutdown message if internal PNC requests are available by deriving the internal PNC requests from the corresponding ComPdu (see UdpNmTxUserDataPduRef) ⌋()

[SWS_UdpNm_00483]⌈ If UdpNmSynchronizedPncShutdownEnabled is set to TRUE, UdpNmPnShutdownMessageRetransmissionDuration is not configured, the UdpNm module has requested a transmission due to synchronized PNC shutdown (see SWS_UdpNm_00475), UdpNm_TxConfirmation is called with result E_NOT_OK or the transmissions request for this PN shutdown message was not accepted (SoAdIf_Transmit returned E_NOT_OK), then the UdpNm shall remove the PNC IDs stored as pending request for a synchronized PNC shutdown of the corresponding NM-channel and report the runtime error UDPNM_E_TRANSMISSION_OF_PN_SHUTDOWN_MESSAGE_FAILED to DET. ⌋()

[SWS_UdpNm_00484]⌈ If `UdpNmSynchronizedPncShutdownEnabled` is set to `TRUE` and a retransmission timer for a PN shutdown message (see ECUC_UdpNm_00098) expires, then UdpNm shall remove the pending request for a synchronized PNC shutdown of the corresponding NM-channel from the storage and report the runtime error `UDPNM_E_TRANSMISSION_OF_PN_SHUTDOWN_MESSAGE_FAILED` to DET.⌋()

7.8.3 Handling of Internal Requested Partial Network Clusters

All internal PNC requests are maintained by ComM. ComM forwards the aggregated internal PNC requests per channel as PNC bit vector to NmIf. This PNC bit vector carries the so-called "Internal Request Array". The UdpNm has to retrieve the latest IRA from NmIf every time an NM_PDU is transmitted. NmIf provides the IRA information to UdpNm and updates the PNC reset timer (each time a relevant PNC is transmitted, the PNC reset timer is re-started).

Note: For all configured NM-channel where UdpNmPnEnabled is set to TRUE, the UdpNm will call Nm_PncBitVectorTxIndication(<NM-channel>, <buffer to store the unfiltered PNC bit vector of aggregated internal PNC requests>) (see SWS_UdpNm_00489 and SWS_UdpNm_00493) to indicate the transmission and to retrieve the current internal PNC requests as PNC bit vector with respect to the configured NmPncBitVectorLength. The UdpNm will copy received internal PNC requests to the PNC bit vector bytes of the NM-PDU.

7.8.4 NM PDU Filter Algorithm (OBSOLETE)

[SWS_UdpNm_00335] {OBSOLETE}⌈The range (in bytes) that contains the PN request information (PN Info Range) in the received NM-PDU is defined by `UdpNmPnInfoOffset` (in bytes) starting from byte 0 and `UdpNmPnInfoLength` (in bytes). This range is called PN Info Range.⌋()

Example:

- UdpNmPnInfoOffset	= 3
- UdpNmPnInfoLength	= 2

Only Byte 3 and Byte 4 of the NM message contains PN request information

[SWS_UdpNm_00336] {OBSOLETE}⌈Every bit of the PN Info Range represents one Partial Network. If the bit is set to 1 the Partial Network is requested. If the bit is set to 0 there is no request for this PN.⌋()

[SWS_UdpNm_00337] {OBSOLETE}⌈By means of the configuration parameter `UdpNmPnFilterMaskByte` the UdpNm is able to detect which PN is relevant for the ECU and which not.
Each bit of `UdpNmPnFilterMasskByte` has the following meaning:

0 The PN request is irrelevant for the ECU. The communication stack of the ECU is not kept awake if this bit is set in a received NM-PDU.

1 The PN request is relevant for the ECU. The communication stack of the ECU is kept awake if this bit is set in a received NM-PDU.⌋()

[SWS_UdpNm_00338] {OBSOLETE}⌈ Each PN filter mask byte shall be mapped (bitwise AND)to the corresponding byte in the PN info range of the NM message. ⌋()

[SWS_UdpNm_00339] {OBSOLETE replaced by SWS_UdpNm_00487 }⌈If at least one bit within the PN Info Range of the received NM-PDU matches with a bit in the NM filter mask the PN request information is relevant for the ECU⌋()

[SWS_UdpNm_00460] {OBSOLETE replaced by SWS_UdpNm_00487 }⌈ If no relevant PN is requested in the received NM-PDU and `UdpNmAllNmMessagesKeepAwake` is `FALSE` the PDU shall be dropped from further processing. ⌋()

[SWS_UdpNm_00461] {OBSOLETE replaced by SWS_UdpNm_00487 }⌈ If no relevant PN is requested in the received NM-PDU and `UdpNmAllNmMessagesKeepAwake` is `TRUE` the PDU shall not be dropped from further Rx Indication handling. ⌋()

7.8.5 Aggregation of Internal and External Requested Partial Networks (OBSOLETE)

Note: This feature is used by every ECU that has to switch I-PDU-Groups because of the activity of partial networks. (e.g. to prevent false timeouts) I-PDU-Groups shall be switched on if the corresponding PN is requested internally or externally. I-PDU-Groups shall not be switched off until all internal and external requests for the corresponding PN are released.

The logic for switching the IPDU-Groups is implemented by ComM. The UdpNm only provides the information if a PN is requested or not. The COM module is used to transfer the data to the upper layers.

To switch the I-PDU-Groups synchronously on all direct connected ECUs, UdpNm shall provide the information of a request change to the upper layer at (almost) the same time on every ECU. This is why the reset timer is restarted on every received and every sent NM message (see below).

The aggregated state of the internal/external requested PNs is called External Internal Requests Aggregated (EIRA).

[SWS_UdpNm_00344] {OBSOLETE}⌈If `UdpNmPnEiraCalcEnabled` is `TRUE,` the UdpNm shall provide the possibility to store external and internal requested PNs combined over all relevant channels (all UdpNm Channels where `UdpNmPnEnabled` is `TRUE`). At initialization the values of all PNs shall be set to 0 (not requested) ⌋()

[SWS_UdpNm_00347] {OBSOLETE} ⌈If

- `UdpNmPnEiraCalcEnabled` is `TRUE`
- a NM-PDU is received
- PNs are requested within this message (bits are set to 1)
- And the requested PNs are set to 1 within the [configured PN filter mask] then UdpNm shall store the request information (value 1) for these PNs ⌋()

[SWS_UdpNm_00348] {OBSOLETE} ⌈If

- `UdpNmPnEiraCalcEnabled` is `TRUE`
- NM-PDU is being requested to send by UdpNM
- PNs are requested within this message(bits are set to 1)
- And the requested PNs are set to 1 within the [configured PN filter mask] then UdpNm shall store the request information (value 1) for these PNs.⌋()

[SWS_UdpNm_00345] {OBSOLETE} ⌈If `UdpNmPnEiraCalcEnabled` is `TRUE`, the UdpNm module shall provide a possibility to monitor each PN, if this PN is still externally or internally requested on at least one of the relevant channels.⌋()

Note: This means, only one timer is required to handle one PN on multiple connected physical channels. For example: only 8 EIRA reset timers are required to handle the requests of a Gateway with 6 physical channels and 8 partial networks.
This is possible because the switch of PN PDU-Groups is done global for the ECU and not dependent of the physical channel.

[SWS_UdpNm_00349] {OBSOLETE} ⌈If `UdpNmPnEiraCalcEnabled` is `TRUE` and a PN is requested by message reception or sending (see SWS_UdpNM_00347 and SWS_UdpNm_00348) the monitoring for this PN shall be restarted with respect to `UdpNmPnResetTime`.⌋()

Note: `UdpNmPnResetTime` shall be configured to a value greater than `UdpNmMsgCycleTime`. If `UdpNmPnResetTime` is configured to a value smaller than
`UdpNmMsgCycleTime` and only one ECU requests the PN, the request state toggles in the EIRA because request state is rested before the requesting ECU is able to send the next NM message.

Note: `UdpNmPnResetTime` shall be configured to a value smaller than `UdpNmTimeoutTime` to avoid that the timer could elapse after NM already changed to Prepare Bus Sleep.

[SWS_UdpNm_00351] {OBSOLETE} ⌈If `UdpNmPnEiraCalcEnabled` is `TRUE` and a PN is not requested again within `UdpNmPnResetTime` the corresponding stored value for this PN shall be set to 0 (not requested) ⌋()

[SWS_UdpNm_00352] {OBSOLETE} ⌈If `UdpNmPnEiraCalcEnabled` is `TRUE` and the stored value for a PN is set to requested or back to not requested (see SWS_UdpNm_00347, SWS_UdpNm_00348 and SWS_UdpNm_00351) UdpNm shall inform upper layers by calling `PduR_UdpNmRxIndication()` for the configured EIRA PDU (i.e changed EIRA information shall be passed to COM). ⌋()

Note: If a PN shutdown message is received (PNSR is set to 1), no special handling is needed, because the according PNC state machines need to stay in COMM_PNC_READY_SLEEP. Only the ERA PDU is handled in a different way (see SWS_UdpNm_00478)

[SWS_UdpNm_00372] {OBSOLETE} ⌈If `UdpNmPnEiraCalcEnabled` is `TRUE` and `UdpNmPnEraCalcEnabled` is `TRUE`, the PN status information shall be stored separately for both, the EIRA and ERA information (compare to SWS_UdpNM_00344 and SWS_UdpNM_00355).⌋()

7.8.6 Aggregation of External Requested Partial Networks (OBSOLETE)

Note: This feature is used by the Gateways to collect only the external PN requests. The external PN requests are mirrored back to the requesting bus and provided to other (required) physical channels of a central gateway.

In case of a sub gateway the requests bit must not be mirrored back to the requesting physical channel in order to avoid static waking between central- and sub gateways. This logic shall be implemented by the ComM.

The UdpNm module provides the information if the PN is externally requested or not. The COM module is used for data transmission to the upper layer.The aggregated state of the external requested PNs is called “External Requests Aggregated” (ERA).

[SWS_UdpNm_00355] {OBSOLETE} ⌈If `UdpNmPnEraCalcEnabled` is `TRUE`, the UdpNM shall provide the possibility to store external requested PNs on each relevant channel. At initialization the values of all PNs shall be set to 0 (not requested) ⌋()

[SWS_UdpNm_00357] {OBSOLETE} ⌈If

- `UdpNmPnEraCalcEnabled` is `TRUE`
- a NM-PDU is received
- PNSR bit is 0
- PNs are requested within this message (bits are set to 1)

- and the requested PNs are set to 1 within the [configured PN filter mask] then UdpNm shall store the request information (value 1) for these PNs⌋()

Note: PNSR Bit set to 1 is only possible if a synchronized PNC shutdown is requested. A synchronized PNC shutdown should be handled across the PN topology, therefore it is assumed that either all coordinators have the synchronized PNC shutdown enabled or all coordinators have the synchronized PNC shutdown disabled. A mixture of both would lead to an unsynchronized PNC shutdown, which has to be avoided.

[SWS_UdpNm_00358] {OBSOLETE} ⌈ If `UdpNmPnEraCalcEnabled` is `TRUE`, the UdpNm module shall provide a possibility to monitor each relevant channel and for each PN if this PN is still externally requested.⌋()

Note: This means, a separate timer is required to handle one PN on multiple physical channels.
For example: 48 ERA reset timers are required to handle the requests of a gateway with 6 physical channels and 8 partial networks. It is not possible to combine the reset timer like EIRA timers, because the external request mustn't be mirrored back to the requesting bus by a sub gateway. Thus it is required to detect the physical channel that is the source of the request bit.

[SWS_UdpNm_00359] {OBSOLETE} ⌈If `UdpNmPnEraCalcEnabled` is `TRUE` and a PN is requested by message reception (see SWS_UdpNM_00357) the monitoring for this PN shall be restarted with respect to the `UdpNmPnResetTime`.⌋()

Note: `UdpNmPnResetTime` shall be configured to a value greater than `UdpNmMsgCycleTime`. If `UdpNmPnResetTime` is configured to a value smaller than `UdpNmMsgCycleTime` and only one ECU requests the PN, the request state toggles in the ERA because request state is rested before the requesting ECU is able to send the next NM-PDU.

Note: `UdpNmPnResetTime` shall be configured to a value smaller than `UdpNmTimeoutTime` to avoid that the timer could elapse after NM already changed to Prepare Bus Sleep.

[SWS_UdpNm_00360] {OBSOLETE} ⌈If `UdpNmPnEraCalcEnabled` is `TRUE` and PN is not requested again within `UdpNmPnResetTime` then the corresponding stored value for this PN shall be set to not requested (value 0)⌋()

[SWS_UdpNm_00361] {OBSOLETE} ⌈If `UdpNmPnEraCalcEnabled` is `TRUE` and the stored value for a PN changes to requested or back to not requested (see SWS_UdpNm_00357 and SWS_UdpNm_00360), the UdpNm module shall inform the upper layers by calling `PduR_UdpNmRxIndication()` for the configured ERA PDU (i.e. changed ERA information shall be passed to the COM module).⌋()

Note: In case both UdpNmPnEiraCalcEnabled and UdpNmPnEraCalcEnabled are set to TRUE, the PN status information is handled as specified in SWS_UdpNm_00372.

[SWS_UdpNm_00478] {OBSOLETE replaced by SWS_UdpNm_00488 } ⌈ If `UdpNmSynchronizedPncShutdownEnabled` is TRUE and the PNSR bit of the received NM PDU is set to 1, then the UdpNm module shall set the bits in the ERA PDU to 0 of the corresponding bits which are set to 1 in the received PN info range, stop the according monitoring for these externally requested PNs (see SWS_UdpNm_00359) and inform the upper layers in the given order:

- call `PduR_UdpNmRxIndication()` for the configured ERA PDU
- call Nm_ForwardSynchronizedPncShutdown() with the configured NetworkHandle (UdpNmComMNetworkHandleRef) ⌋(RS_Nm_02541)

Note: The PN information of a received PN shutdown message shall be used to release the PNCs for a synchronized shutdown and pass this ERA information to the ComM module by writing the corresponding ERA PDU. The synchronized PNC shutdown has to be handled as fast as possible, therefore the Nm module is informed immediately.

7.8.7 Spontaneous Transmission of NM-PDUs via UdpNm_NetworkRequest

[SWS_UdpNm_00362]⌈ If `UdpNm_NetworkRequest` is called, `UdpNmPnHandleMultipleNetworkRequests` is set to `TRUE` and UdpNm is in Ready Sleep State, Normal Operation State or Repeat Message State, UdpNm shall change to or restart the Repeat Message State⌋() .

Note: If `UdpNmPnHandleMultipleNetworkRequests` is set to TRUE the UdpNm feature 'Immediate Transmission' is mandatory.

Note: The PNC Control Module (e.g. ComM) is responsible to call UdpNm_NetworkRequest if the PNC bits change.

7.9 Payload (PDU) Structure

The figure below shows an example for n bytes PDU length where the source node identifier is located in the first byte, the control bit vector in the second byte, user data is used and partial network is enabled. User data range is located between the system bytes and the PNC bit vector:

	Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
Byte 0	Source Node Identifier (default)							
Byte 1	Control Bit Vector (default)							
Byte 2	User data 0							
Byte 3	User data 1							
Byte 4	…							
Byte i+2	User data i							
Byte i+3	PNC bit vector - byte 0							
Byte i+4	PNC bit vector - byte 1							
…	…							
Byte n	PNC bit vector - byte j							

Figure 2: NM packet payload (NM PDU) default format.

Note:
The length of the Network Management PDU (NM PDU) is defined by the PduLength parameter in the "global" ECUC module ([EcuC003_Conf], see Ecu Configuration specification).

[SWS_UdpNm_00074] ⌈ The location of the source node identifier shall be configurable by means of `UDPNM_PDU_NID_POSITION` to Byte 0, Byte 1, or off. ⌋()

[SWS_UdpNm_00075] ⌈ The location of the control Bit vector shall be configurable by means of `UDPNM_PDU_CBV_POSITION` to Byte 0, Byte 1, or off. ⌋()

Note: The location of the PNC bit vector is configurable by means of NmPncBitVectorOffset and NmPncBitVectorLength of the corresponding NM-channel. The location of the PNC bit vector is placed after the system bytes (control bit vector and source node identifier) and within the PduLenght of the NM-PDU.

[SWS_UdpNm_00491] {DRAFT} ⌈ The remaining bytes not assigned to Nm System Bytes or PNC bit vector shall be available for User Data. ⌋()

Note: According to [23] (TPS_SYST_03069, TPS_SYST_03070, TPS_SYST_03071, TPS_SYST_03072) the use and location of user data is configurable. If user data are used, the user data are placed within the PduLenght of the NM-PDU and do not overlap with the range of system bytes or PNC bit vector. If partial network functionaliy is enabled (UdpNmPnEnabled is set to TRUE) and user data are used, the user data range is exclusively located either between the system bytes and the PNC bit vector or between the PNC bit vector and the end of the NM-PDU. The length of user data range shall be calculated according the following restrictions:

- If the user data range resides between the system bytes and the PNC bit vector, then the length of the user data range is determined by the difference of the PNC bit vector offset and the length of the system bytes.
- If the user data range resides between the PNC bit vector and the end of the NM-PDU, then the length of the user data range is determined by the difference of the NM-PDU length and the position/index of the last byte of the PNC bit vector (defined by PNC bit offset + PNC bit vector length)

If partial network functionaliy is disabled (UdpNmPnEnabled is set to FALSE) and user data are used, the user data range is determined by the difference of NM-PDU length and the length of the system bytes.

[SWS_UdpNm_00076] {DRAFT} ⌈ The length of an NM packet shall not exceed the MTU(Maximum Transmission Unit)of the underlying physical transport layer. ⌋()

The figure below describes the format of the Control Bit Vector:

	Bit 7	***Bit 6***	***Bit 5***	***Bit 4***	***Bit 3***	***Bit 2***	***Bit 1***	***Bit 0***
CBV	Res	PNI Bit	Partial Network Learning Bit	Active Wakeup Bit	NM Coordinator Sleep Ready	Res	PN Shutdown Request Bit	Repeat Message Request

Figure 3: Control Bit Vector.

Note: Bit 1 and 2 were used in R3.2 as NM Coordinator ID (Low Bit)

[SWS_UdpNm_00045] {DRAFT} ⌈ The Control Bit Vector shall consist of:

Bit 0: Repeat Message Request Bit
0: Repeat Message State not requested
1: Repeat Message State requested

Bit 1: PN Shutdown Request Bit (PNSR)
0: NM message does not contain synchronized Partial Network shutdown request
1: NM message does contain synchronized Partial Network shutdown request for at least one PNC

Bit 3 :NM Coordinator Sleep Bit
0: Start of synchronized shutdown is not requested by main coordinator
1: Start of synchronized shutdown is requested by main coordinator

Bit 4 Active Wakeup Bit
0: Node has not woken up the network (passive wakeup)
1: Node has woken up the network (active Wakeup)

Bit 5 Partial Network Learning Bit (PNL)
0: PNC learning is not requested
1: PNC learning is requested

Bit 6 Partial Network Information Bit (PNI)

0: NM message contains no Partial Network request information
1: NM message contains Partial Network request information

Bit 2,7 are reserved for future extensions
0 : Disabled / Reserved for future usage ⌋(RS_Nm_02540)

Note: The Control Bit Vector is initialized with `0x00` during initialization (also refer to SWS_UdpNm_00085).

[SWS_UdpNm_00013] ⌈ The source node identifier shall be set with the configuration parameter `UDPNM_NODE_ID` unless `UDPNM_PDU_NID_POSITION` is set to off. ⌋()

[SWS_UdpNm_00366]⌈ If the UdpNm performs a state change from BusSleep state or PrepareBusSleep state to NetworkMode due to a call to `UdpNm_NetworkRequest()` (i.e. due to an active wakeup) and `UdpNmActiveWakeupBitEnabled` is `TRUE`, the UdpNm shall set the ActiveWakeupBit in the CBV. ⌋()

[SWS_UdpNm_00367]⌈ If the UdpNm module leaves the NetworkMode and `UdpNmActiveWakeupBitEnabled` is `TRUE`, the UdpNm module shall clear the ActiveWakeupBit in the CBV.⌋()

7.10 Functional requirements on UdpNm API

[SWS_UdpNm_00014] {DRAFT}⌈If `UdpNmRepeatMsgIndEnabled` is set to TRUE and the Repeat Message Request bit set to 1 is received UdpNm module shall call the callback function Nm_RepeatMessageIndication. In case the Partial Network Learning Bit is also received and `UdpNmDynamicPncToChannelMappingEnabled` is set to TRUE the parameter pnLearningBitSet in this function call shall be set to TRUE, otherwise to FALSE. ⌋(RS_Nm_00153)

7.11 Car Wakeup

[SWS_UdpNm_00373]⌈ The position of the Car Wakeup bit in the NM-PDU is defined by the configuration parameters `UdpNmCarWakeUpBytePosition` and `UdpNmCarWakeUpBitPosition`.⌋()

[SWS_UdpNm_00374]⌈ If the Car Wakeup bit within any received NM-PDU is 1, `UdpNmCarWakeUpRxEnabled` is `TRUE`, and `UdpNmCarWakeUpFilterEnabled` is

FALSE UdpNm shall call Nm_CarWakeUpIndication and perform the standard Rx indication handling.⌋()

[SWS_UdpNm_00375]⌈ If UdpNm_GetPduData is called in the context of Nm_CarWakeUpIndication and if UdpNmNodeDetectionEnabled or UdpNmUserDataEnabled or UdpNmNodeIdEnabled is set to TRUE, UdpNm shall return the PDU data of the PDU that causes the call of Nm_CarWakeUpIndication. ⌋()

Note: This is required to enable ECU to identify detail about the sender of the Car Wakeup request

[SWS_UdpNm_00376]⌈ If UdpNmCarWakeUpFilterEnabled is TRUE, the Car Wakeup bit within any received NM-PDU is 1, UdpNmCarWakeUpRxEnabled is TRUE and the Node ID in the received NM-PDU is equal to UdpNmCarWakeUpFilterNodeId the UdpNm module shall call Nm_CarWakeUpIndication and perform the standard Rx Indication handling⌋()

Note: The Car Wakeup filter is necessary to realize sub gateways that only consider the Car Wakeup of the central Gateway to avoid wrong wakeups

7.12 Error Classification

This section describes how the UdpNm module has to manage the error classes that may occur during the life cycle of this basic software.
The general requirements document of AUTOSAR [2] specifies that all basic software modules must distinguish (according to the product life cycle) two error types:

Development errors: these errors should be detected and fixed during the development phase. In most cases, these errors are software errors. The detection errors that should only occur during development can be switched off for production code (by static configuration, namely preprocessor switches).

Production errors: these errors are hardware errors and software exceptions that cannot be avoided and are expected to occur in the production (i.e. series) code. This kind of error is commonly known as a run-time error.

[SWS_UdpNm_00223]⌈ On errors and exceptions, the UdpNm module shall not modify its current module state. ⌋()

7.12.1 Development Errors

[SWS_UdpNm_00018]⌈

Type of error	***Related error code***	***Error value***
API service used without module initialization	UDPNM_E_UNINIT	0x01
API service called with wrong channel handle	UDPNM_E_INVALID_CHANNEL	0x02
API service called with wrong PDU ID.	UDPNM_E_INVALID_PDUID	0x03
UdpNm initialization has failed, e.g. selected configuration set doesn't exist	UDPNM_E_INIT_FAILED	0x04
Null pointer has been passed as an argument	UDPNM_E_PARAM_POINTER	0x12

⌋()

[SWS_UdpNm_00189]⌈ Development errors shall not be returned by API functions; in case of a development error, the respective API function will return `E_NOT_OK`, if applicable. ⌋()

7.12.2 Run Time Errors

[SWS_UdpNm_00465]⌈

Type of error	***Related error code***	***Error value***
NM-Timeout timer has expired outside Ready Sleep State (either in Repeat Message state or in Normal Operation state)	UDPNM_E_NETWORK_TIMEOUT	0x11
A NM message with PN Shutdown Request Bit was received on a channel that is actively coordinated by the ComM PNC Gateway. **Tags:** atp.Status=draft	UDPNM_E_INVALID_PN_SYNC_SHUTDOWN_REQUEST	0x20
Retransmission timer for a PN shutdown message has expired, because a PN shutdown message could not be transmitted on the network within the configured duration of re-transmission.	UDPNM_E_TRANSMISSION_OF_PN_SHUTDOWN_MESSAGE_FAILED	0x21

⌋()

[SWS_UdpNm_00466]⌈ When the NM-Timeout Timer expires in the Repeat Message State, the UdpNm module shall report the runtime error `UDPNM_E_NETWORK_TIMEOUT` to the Default Error Tracer. ⌋(RS_Nm_00137)

[SWS_UdpNm_00467] ⌈ When the NM-Timeout Timer expires in the Normal Operation State, the UdpNm module shall report runtime error `UDPNM_E_NETWORK_TIMEOUT` to the Default Error Tracer. ⌋(RS_Nm_00137)

7.12.3 Transient Faults

There are no transient faults.

7.12.4 Production Errors

There are no production errors.

7.12.5 Extended Production Errors

There are no extended production errors.

7.13 Scheduling of the main function

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

7.14 Application notes

7.14.1 Wakeup notification

Wakeup notification is defined in detail in the ECU State Manager specification [11].

7.14.2 Coordination of coupled networks

[SWS_UdpNm_00185] ⌈ Support of bus synchronization on demand shall be statically configurable with use of the `UdpNmBusSynchronizationEnabled` switch (configuration parameter). ⌋()

Note: Since the shutdown of UdpNm can be done at any time, the call of the API `Nm_SynchronizationPoint` is not supported.

7.15 Version check

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

7.16 Parameter check

[SWS_UdpNm_00196]⌈ If detection of development errors is enabled by `UDPNM_DEV_ERROR_DETECT` (configuration parameter), validity checks for all input parameters shall be performed for each UDP NM API service call.⌋()

[SWS_UdpNm_00197]⌈ Parameter type checking shall be performed at compile time; if types do not match, the compilation process shall be stopped and respective compilation warnings or errors shall be returned as far as supported by the compiler. ⌋()

[SWS_UdpNm_00198]⌈ Parameter value check (for parameters of the constant value) shall be performed at configuration time; if the value is invalid, the configuration process shall be stopped and the respective configuration error shall be reported. ⌋()

[SWS_UdpNm_00199]⌈ Parameter value check (for parameters of the variable value) shall be performed at execution time; if the value is invalid, execution of a service shall be denied and the respective development error shall be reported. ⌋()

8 API specification

[SWS_UdpNm_00244] ⌈ The UdpNm module shall reject the execution of a service called with an invalid parameter and shall inform the DET. ⌋()

AUTOSAR UdpNm API consists of services, which are UDP specific and can be called whenever they are required; each service apart from `UdpNm_Init` refers to one NM channel only.

[SWS_UdpNm_00190] ⌈ Production errors shall not be returned by API functions; in case of a production error, the respective API function will return `E_NOT_OK`, if applicable. ⌋()

[SWS_UdpNm_00192] ⌈ When NM API service with an invalid network handle is called, the called function shall not be executed, but instead of that it shall report `UDPNM_E_INVALID_CHANNEL` to the Default Error Tracer (if development error detection is enabled) otherwise it shall return `E_NOT_OK` to the calling function⌋()

Note: The network handle is invalid if it is different from allowed configured values.

[SWS_UdpNm_00492] {DRAFT} ⌈ When a Null pointer has been passed to a UdpNm service, the called function shall not be executed and it shall return E_NOT_OK to the calling function if applicable. If development error detection is enabled (UdpNmDevErrorDetect is set to TRUE) the corresponding error UDPNM_E_PARAM_POINTER shall be reported to DET. ⌋()

[SWS_UdpNm_00463] ⌈ When UdpNm Callback Notifications with an invalid Pdu ID are called, the called function shall not be executed and `E_NOT_OK` shall be returned if possible. If Development Error Detection is enabled then additionally UdpNm shall report `UDPNM_E_INVALID_PDUID` to the Default Error Tracer. ⌋()

[SWS_UdpNm_00314] ⌈ If `UdpNmComUserDataSupport` is enabled and the UdpNm User Data length does not match with the length of the referenced I-PDU an error shall be reported at generation time. ⌋()

Note: NULL Pointer checking is specified within BSW General [21].

8.1 Imported Types

The following types of `Std_Types.h` are imported:

```
boolean
uint8
uint16
```

uint32

[]⌈

Module	*Header File*	*Imported Type*
ComStack_Types	ComStack_Types.h	NetworkHandleType
	ComStack_Types.h	PNCHandleType
	ComStack_Types.h	PduIdType
	ComStack_Types.h	PduInfoType
	ComStack_Types.h	PduLengthType
Nm	NmStack_types.h	Nm_ModeType
	NmStack_types.h	Nm_StateType
Std	Std_Types.h	Std_ReturnType
	Std_Types.h	Std_VersionInfoType

⌋()

8.2 Type Definitions

8.2.1 UdpNm_ConfigType

This type shall contain the parameters of the container `UdpNm_GlobalConfig` and its sub containers.
[SWS_UdpNm_00308]⌈

Name	UdpNm_ConfigType	
Kind	Structure	
Elements	implementation specific	
	Type	--
	Comment	This type shall contain the parameters of the container UdpNm_Global Config and its sub containers.
Description	--	
Available via	UdpNm.h	

⌋()

8.2.2 UdpNm_PduPositionType

[SWS_UdpNm_00304]⌈

Name	UdpNm_PduPositionType		
Kind	Enumeration		
Range	UDPNM_PDU_BYTE_0	0x00	Byte 0 is used
	UDPNM_PDU_BYTE_1	0x01	Byte 1 is used
	UDPNM_PDU_OFF	0xFF	Node Identification is not used
Description	Used to define the position of the control bit vector within the NM PACKET.		
Available via	UdpNm.h		

⌋()

8.3 Function definitions

8.3.1 UdpNm_Init

[SWS_UdpNm_00208]⌈

Service Name	UdpNm_Init	

Syntax	`void UdpNm_Init (`
	`  const UdpNm_ConfigType* UdpNmConfigPtr`
	`)`	

Service ID [hex]	0x01	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	UdpNmConfigPtr	Pointer to a selected configuration structure
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Initialize the complete UdpNm module, i.e. all channels which are activated at configuration time are initialized. A UDP socket shall be set up with the TCP/IP stack. Caveats: This function has to be called after initialization of the TCP/IP stack. Configuration: Mandatory	
Available via	UdpNm.h	

⌋()

[SWS_UdpNm_00210] ⌈ If an error has to be indicated to the DET the value `0x00` shall be used as the instance id. ⌋()

Rationale: the value 0 x 00 is not error value but instance ID

8.3.2 UdpNm_PassiveStartUp

[SWS_UdpNm_00211]⌈

Service Name	UdpNm_PassiveStartUp	
Syntax	`Std_ReturnType UdpNm_PassiveStartUp ( NetworkHandleType nmChannelHandle )`	
Service ID [hex]	0x0e	
Sync/Async	Asynchronous	
Reentrancy	Reentrant (but not for the same NM-Channel)	
Parameters (in)	nmChannelHandle	Identification of the NM-channel
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: No error E_NOT_OK: Passive startup of network management has failed
Description	Passive startup of the AUTOSAR UdpNm. It triggers the transition from Bus-Sleep Mode or Prepare Bus Sleep Mode to the Network Mode in Repeat Message State. Caveats: UdpNm is initialized correctly.	
Available via	UdpNm.h	

⌋()

[SWS_UdpNm_00147] ⌈If `UdpNm_PassiveStartUp` is called in the Network Mode, the UdpNm module shall not execute this service and shall return `E_NOT_OK`.⌋()

8.3.3 UdpNm_NetworkRequest

[SWS_UdpNm_00213]⌈

Service Name	UdpNm_NetworkRequest
Syntax	`Std_ReturnType UdpNm_NetworkRequest (`

	`NetworkHandleType nmChannelHandle`
	`)`	

Service ID [hex]	0x02	
Sync/Async	Asynchronous	
Reentrancy	Reentrant (but not for the same NM-Channel)	
Parameters (in)	nmChannelHandle	Identification of the NM-channel
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: No error E_NOT_OK: Requesting of network has failed
Description	Request the network, since ECU needs to communicate on the bus. Network state shall be changed to 'requested' Caveats: UdpNm is initialized correctly. Configuration: Optional (Only available if UdpNmPassiveModeEnabled == false)	
Available via	UdpNm.h	

⌋()

8.3.4 UdpNm_NetworkRelease

[SWS_UdpNm_00214]⌈

Service Name	UdpNm_NetworkRelease	

Syntax	`Std_ReturnType UdpNm_NetworkRelease (`
	`  NetworkHandleType nmChannelHandle`
	`)`	

Service ID [hex]	0x03	
Sync/Async	Asynchronous	
Reentrancy	Reentrant (but not for the same NM-Channel)	
Parameters (in)	nmChannelHandle	Identification of the NM-channel
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: No error E_NOT_OK: Releasing of network has failed
Description	Release the network, since ECU doesn't have to communicate on the bus.	

	Network state shall be changed to 'released'. Caveats: UdpNm is initialized correctly. Configuration: Optional (Only available if UdpNmPassiveModeEnabled == false)
Available via	UdpNm.h

⌋()

8.3.5 UdpNm_DisableCommunication

[SWS_UdpNm_00215]⌈

Service Name	UdpNm_DisableCommunication	

Syntax	`Std_ReturnType UdpNm_DisableCommunication (`
	`  NetworkHandleType nmChannelHandle`
	`)`	

Service ID [hex]	0x0c	
Sync/Async	Asynchronous	
Reentrancy	Reentrant (but not for the same NM-Channel)	
Parameters (in)	nmChannel Handle	Identification of the NM-channel
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: No error E_NOT_OK: Disabling of NM PDU transmission ability has failed
Description	Disable the NM PDU transmission ability due to a ISO14229 Communication Control (0x28) service Caveats: UdpNm is initialized correctly. Configuration: Optional (Only available if UdpNmComControlEnabled == true)	
Available via	UdpNm.h	

⌋(RS_Nm_02512)

[SWS_UdpNm_00307]⌈ If the module operates in passive mode (`UdpNmPassiveModeEnabled`) the service `UdpNm_DisableCommunication` shall have no effects and shall directly return `E_NOT_OK.` ⌋()

8.3.6 UdpNm_EnableCommunication

[SWS_UdpNm_00216]⌈

Service Name	UdpNm_EnableCommunication

Syntax	`Std_ReturnType UdpNm_EnableCommunication ( NetworkHandleType nmChannelHandle )`	
Service ID [hex]	0x0d	
Sync/Async	Asynchronous	
Reentrancy	Reentrant (but not for the same NM-Channel)	
Parameters (in)	nmChannelHandle	Identification of the NM-channel
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: No error E_NOT_OK: Enabling of NM PDU transmission ability has failed
Description	Enable the NM PDU transmission ability due to a ISO14229 Communication Control (0x28) service Caveats: UdpNm is initialized correctly. Configuration: Optional (Only available if UdpNmComControlEnabled == true).	
Available via	UdpNm.h	

⌋(RS_Nm_02512)

[SWS_UdpNm_00176] ⌈ The optional service `UdpNm_EnableCommunication` shall enable the NM PDU transmission ability if the NM PDU transmission ability is disabled. ⌋()

[SWS_UdpNm_00177] ⌈ The optional service `UdpNm_EnableCommunication` shall return `E_NOT_OK` if the NM PDU transmission ability is already enabled when the service is called. ⌋()

[SWS_UdpNm_00305] ⌈ The service `UdpNm_EnableCommunication` shall return `E_NOT_OK`, if the current mode is not Network Mode. ⌋()

[SWS_UdpNm_00306] ⌈ If the module operates in passive mode (`UdpNmPassiveModeEnabled` is `TRUE`) the service `UdpNm_EnableCommunication` shall have no effects and shall directly return `E_NOT_OK`. ⌋()

8.3.7 UdpNm_SetUserData

[SWS_UdpNm_00217]⌈

Service Name	UdpNm_SetUserData

Syntax	`Std_ReturnType UdpNm_SetUserData ( NetworkHandleType nmChannelHandle, const uint8* nmUserDataPtr )`	
Service ID [hex]	0x04	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	nmChannel Handle	Identification of the NM-channel
	nmUserDataPtr	Pointer where the user data for the next transmitted NM message shall be copied from.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_Return-Type	E_OK: No error E_NOT_OK: Setting of user data has failed
Description	Set user data for all NM messages transmitted on the bus after this function has returned without error. Caveats: UdpNm is initialized correctly. Configuration: Optional (Only available if UdpNmUserDataEnabled==true and UdpNmPassiveModeEnabled==false).	
Available via	UdpNm.h	

⌋()

8.3.8 UdpNm_GetUserData

[SWS_UdpNm_00218]⌈

Service Name	UdpNm_GetUserData	
Syntax	`Std_ReturnType UdpNm_GetUserData ( NetworkHandleType nmChannelHandle, uint8* nmUserDataPtr )`	
Service ID [hex]	0x05	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	nmChannel Handle	Identification of the NM-channel
Parameters (inout)	None	

Parameters (out)	nmUserData Ptr	Pointer where user data out of the most recently received NM message shall be copied to.
Return value	Std_Return- Type	E_OK: No error E_NOT_OK: Getting of user data has failed
Description	Get user data from the most recently received NM message. Caveats: UdpNm is initialized correctly. Configuration: Optional (Only available if UdpNmUserDataEnabled == true).	
Available via	UdpNm.h	

⌋()

8.3.9 UdpNm_GetNodeIdentifier

[SWS_UdpNm_00219]⌈

Service Name	UdpNm_GetNodeIdentifier	

Syntax	`Std_ReturnType UdpNm_GetNodeIdentifier (`
	`  NetworkHandleType nmChannelHandle,`
	`  uint8* nmNodeIdPtr`
	`)`	

Service ID [hex]	0x06	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	nmChannel Handle	Identification of the NM-channel
Parameters (inout)	None	
Parameters (out)	nmNodeId Ptr	Pointer where the source node identifier from the most recently received NM PDU shall be copied to.
Return value	Std_Return- Type	E_OK: No error E_NOT_OK: Getting of the node identifier out of the most recently received NM PDU has failed or is not configured for this network handle.
Description	Get node identifier from the most recently received NM PDU. Caveats: UdpNm is initialized correctly.	
Available via	UdpNm.h	

⌋()

[SWS_UdpNm_00132]⌈ The service call `UdpNm_GetNodeIdentifier` shall provide the node identifier out of the most recently received Network Management PDU if `UdpNmNodeIdEnabled` is set to `TRUE`. ⌋()

8.3.10 UdpNm_GetLocalNodeIdentifier

[SWS_UdpNm_00220]⌈

Service Name	UdpNm_GetLocalNodeIdentifier	

Syntax	`Std_ReturnType UdpNm_GetLocalNodeIdentifier (`
	`  NetworkHandleType nmChannelHandle,`
	`  uint8* nmNodeIdPtr`
	`)`	

Service ID [hex]	0x07	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	nmChannel Handle	Identification of the NM-channel
Parameters (inout)	None	
Parameters (out)	nmNodeIdPtr	Pointer where node identifier of the local node shall be copied to.
Return value	Std_Return-Type	E_OK: No error E_NOT_OK: Getting of the node identifier of the local node has failed or is not configured for this network handle.
Description	Get node identifier configured for the local node. Caveats: UdpNm is initialized correctly.	
Available via	UdpNm.h	

⌋()

[SWS_UdpNm_00133] ⌈ The service call `UdpNm_GetLocalNodeIdentifier` shall provide the node identifier configured for the local host node if `UdpNmNodeIdEnabled` is set to `TRUE`. ⌋()

8.3.11 UdpNm_RepeatMessageRequest

[SWS_UdpNm_00221]⌈

Service Name	UdpNm_RepeatMessageRequest

Syntax	`Std_ReturnType UdpNm_RepeatMessageRequest (`
	`  NetworkHandleType nmChannelHandle`
	`)`

Service ID [hex]	0x08
Sync/Async	Asynchronous
Reentrancy	Reentrant (but not for the same NM-Channel)

Parameters (in)	nmChannel Handle	Identification of the NM-channel
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_Return-Type	E_OK: No error E_NOT_OK: Setting of Repeat Message Request Bit has failed or is not configured for this network handle.
Description	Set Repeat Message Request Bit for all NM messages transmitted on the bus after this function has returned without error.	
Available via	UdpNm.h	

⌋()

[SWS_UdpNm_00137] ⌈ If the service `UdpNm_RepeatMessageRequest` is called in Repeat Message State, Prepare Bus-Sleep Mode or Bus-Sleep Mode, the UdpNm module shall not execute the service and return `E_NOT_OK`. ⌋()

8.3.12 UdpNm_GetPduData

[SWS_UdpNm_00309]⌈

Service Name	UdpNm_GetPduData	

Syntax	`Std_ReturnType UdpNm_GetPduData (`
	`  NetworkHandleType nmChannelHandle,`
	`  uint8* nmPduDataPtr`
	`)`	

Service ID [hex]	0x0a	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	nmChannel Handle	Identification of the NM-channel
Parameters (inout)	None	
Parameters (out)	nmPduDataPtr	Pointer where NM PDU shall be copied to.
Return value	Std_Return-Type	E_OK: No error E_NOT_OK: Getting of NM PDU Data has failed or is not configured for this network handle.
Description	Get the whole PDU data out of the most recently received NM message. Caveats: UdpNm is initialized correctly.	

Available via	UdpNm.h

⌋()

[SWS_UdpNm_00138] ⌈ The service call `UdpNm_GetPduData` shall provide whole payload (Source Node ID, Control Bit Vector and User Data) of the most recently received Network Management PDU if `UdpNmNodeDetectionEnabled` or `UdpNmUserDataEnabled` or `UdpNmNodeIdEnabled` is set to `TRUE`. ⌋()

8.3.13 UdpNm_GetState

[SWS_UdpNm_00310]⌈

Service Name	UdpNm_GetState	
Syntax	`Std_ReturnType UdpNm_GetState ( NetworkHandleType nmChannelHandle, Nm_StateType* nmStatePtr, Nm_ModeType* nmModePtr )`	
Service ID [hex]	0x0b	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	nmChannel Handle	Identification of the NM-channel
Parameters (inout)	None	
Parameters (out)	nmStatePtr	Pointer where state of the network management shall be copied to.
	nmModePtr	Pointer where the mode of the network management shall be copied to.
Return value	Std_ReturnType	E_OK: No error E_NOT_OK: Getting of NM state has failed
Description	Returns the state and the mode of the network management. Caveats: UdpNm is initialized correctly. Configuration: Mandatory	
Available via	UdpNm.h	

⌋()

8.3.14 UdpNm_GetVersionInfo

[SWS_UdpNm_00224]⌈

Service Name	UdpNm_GetVersionInfo	

Syntax	`void UdpNm_GetVersionInfo (`
	`  Std_VersionInfoType* versioninfo`
	`)`	

Service ID [hex]	0x09	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	versioninfo	Pointer to where to store the version information of this module.
Return value	None	
Description	This service returns the version information of this module.	
Available via	UdpNm.h	

⌋()

[SWS_UdpNm_00318] ⌈ If DET is enabled for the UdpNm module, the function `UdpNm_GetVersionInfo` shall raise `UDPNM_E_PARAM_POINTER`, if the argument versioninfo is a NULL pointer and return without any action. ⌋()

8.3.15 UdpNm_RequestBusSynchronization

[SWS_UdpNm_00226]⌈

Service Name	UdpNm_RequestBusSynchronization	

Syntax	`Std_ReturnType UdpNm_RequestBusSynchronization (`
	`  NetworkHandleType nmChannelHandle`
	`)`	

Service ID [hex]	0x14	
Sync/Async	Asynchronous	
Reentrancy	Non Reentrant	
Parameters (in)	nmChannelHandle	Identification of the NM-channel
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: No error E_NOT_OK: Requesting of bus synchronization has failed

Description	Request bus synchronization. Caveats: UdpNm is initialized correctly. Configuration: Optional (only available if UdpNmBusSynchronization Enabled==true and UdpNmPassiveModeEnabled==false).
Available via	UdpNm.h

⌋()

[SWS_UdpNm_00130] ⌈ The service call `UdpNm_RequestBusSynchronization` shall trigger transmission of a single Network Management PDU if `UdpNmPassiveModeEnabled` (configuration parameter) is `FALSE`. ⌋()

Rationale: This service is typically used for supporting the NM gateway extensions.

[SWS_UdpNm_00187] ⌈ If `UdpNm_RequestBusSynchronization` is called in Bus-Sleep Mode and Prepare Bus-Sleep Mode the UdpNm module shall not execute the service and shall return `E_NOT_OK`. ⌋()

8.3.16 UdpNm_CheckRemoteSleepIndication

[SWS_UdpNm_00227]⌈

Service Name	UdpNm_CheckRemoteSleepIndication	

Syntax	`Std_ReturnType UdpNm_CheckRemoteSleepIndication (`
	`  NetworkHandleType nmChannelHandle,`
	`  boolean* NmRemoteSleepIndPtr`
	`)`	

Service ID [hex]	0x11	
Sync/Async	Synchronous	
Reentrancy	Reentrant (but not for the same NM-Channel)	
Parameters (in)	nmChannelHandle	Identification of the NM-channel
Parameters (inout)	None	
Parameters (out)	NmRemoteSleep IndPtr	Pointer where check result of remote sleep indication shall be copied to.
Return value	Std_ReturnType	E_OK: No error E_NOT_OK: Checking of remote sleep indication bits has failed
Description	Check if remote sleep indication takes place or not. Caveats: UdpNm is initialized correctly. Configuration: Optional (only available if UdpNmRemoteSleepIndEnabled == true)	
Available via	UdpNm.h	

⌋()

[SWS_UdpNm_00153] ⌈ The service call `UdpNm_CheckRemoteSleepIndication` shall provide the information about current status of Remote Sleep Indication (i.e. already detected or not). ⌋()

8.3.17 UdpNm_SetSleepReadyBit

[SWS_UdpNm_00324]⌈

Service Name	UdpNm_SetSleepReadyBit	

Syntax	`Std_ReturnType UdpNm_SetSleepReadyBit (`
	`  NetworkHandleType nmChannelHandle,`
	`  boolean nmSleepReadyBit`
	`)`	

Service ID [hex]	0x16	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	nmChannel Handle	Identification of the NM-channel
	nmSleepReadyBit	Value written to ReadySleep Bit in CBV
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: No error E_NOT_OK: Writing of remote sleep indication bit has failed
Description	Set the NM Coordinator Sleep Ready bit in the Control Bit Vector	
Available via	UdpNm.h	

⌋()

8.3.18 UdpNm_Transmit

[SWS_UdpNm_00313]⌈

Service Name	UdpNm_Transmit

Syntax	`Std_ReturnType UdpNm_Transmit (`
	`  PduIdType TxPduId,`
	`  const PduInfoType* PduInfoPtr`
	`)`

Service ID [hex]	0x49

Sync/Async	Synchronous	
Reentrancy	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in)	TxPduId	Identifier of the PDU to be transmitted
	PduInfoPtr	Length of and pointer to the PDU data and pointer to Meta Data.
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_Return-Type	E_OK: Transmit request has been accepted. E_NOT_OK: Transmit request has not been accepted.
Description	Requests transmission of a PDU.	
Available via	UdpNm.h	

⌋()

[SWS_UdpNm_00315] ⌈ If `UdpNmComUserDataSupport` or `UdpNmPnEnabled` is enabled the UdpNm implementation shall provide an API `UdpNm_Transmit`. ⌋ (RS_Nm_02503)

8.3.19 UdpNm_PnLearningRequest

[SWS_UdpNm_91004]{DRAFT} ⌈

Service Name	UdpNm_PnLearningRequest (draft)	

Syntax	`Std_ReturnType UdpNm_PnLearningRequest (`
	`  NetworkHandleType nmChannelHandle`
	`)`	

Service ID [hex]	0x4a	
Sync/Async	Asynchronous	
Reentrancy	Reentrant (but not for the same NM-channel)	
Parameters (in)	nmChannel Handle	Identification of the NM-channel
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_ReturnType	E_OK: No error E_NOT_OK: PN Learning Requesthas failed or is not

		configured for this network handle.
Description	Set Repeat Message Request Bit and Partial Network Learning Bit for NM messages transmitted next on the bus. This will force all nodes on the bus to enter the PNC Learning Phase. This is needed for the optional Dynamic PNC-to-channel-mapping feature. **Tags:** atp.Status=draft	
Available via	UdpNm.h	

⌋()

[SWS_UdpNm_00471]⌈ If the function UdpNm_PnLearningRequest is called in “Prepare Bus-Sleep Mode” or “Bus Sleep Mode” no functionality shall be executed and E_NOT_OK shall be returned.⌋()

8.3.20 UdpNm_RequestSynchronizedPncShutdown

[SWS_UdpNm_91002]{DRAFT} ⌈

Service Name	UdpNm_RequestSynchronizedPncShutdown (draft)	

Syntax	`Std_ReturnType UdpNm_RequestSynchronizedPncShutdown (`
	`  NetworkHandleType nmChannelHandle,`
	`  PNCHandleType pncId`
	`)`	

Service ID [hex]	0x4b	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different nmChannelHandle. Non reentrant for the same nmChannel Handle.	
Parameters (in)	nmChannel Handle	Identifier of the NM-Channel where the given PNC (pncId) is assigned to.
	pncId	Identifier of the PNC which is requested for a synchronized shutdown across the PN topology
Parameters (inout)	None	
Parameters (out)	None	
Return value	Std_Return-Type	E_OK: Request has been accepted. E_NOT_OK: Request has not been accepted.
Description	Requests transmission of a NM-PDU with PNSR bit set to 1 (PN shutdown message) **Tags:** atp.Status=draft	
Available via	UdpNm.h	

⌋(RS_Nm_02545)

[SWS_UdpNm_00479] {DRAFT} ⌈ If `UdpNmSynchronizedPncShutdownEnabled` is set to `TRUE` the UdpNm implementation shall provide an API `UdpNm_RequestSynchronizedPncShutdown`. ⌋(RS_Nm_02503)

8.4 Call-back notifications

8.4.1 UdpNm_SoAdIfTxConfirmation

[SWS_UdpNm_00228]⌈

Service Name	UdpNm_SoAdIfTxConfirmation	
Syntax	`void UdpNm_SoAdIfTxConfirmation ( 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.	
Available via	UdpNm.h	

⌋()

Note: The callback function `UdpNm_SoAdIfTxConfirmation` is called by the SoAd and is implemented by the UdpNm module.

Note: The callback function `UdpNm_SoAdIfTxConfirmation` is either called on interrupt level (interrupt mode) or on task level (Polling Mode) with respect to the context.

The value passed to UdpNm via the API parameter `TxPduId` shall refer to the NM channel handle, i.e. a mapping from PduId to NM channel handle is not necessary.

[SWS_UdpNm_00316] ⌈ If `UdpNmComUserDataSupport` is enabled the UdpNm shall call `PduR_UdpNmTxConfirmation` within the message transmission confirmation function `UdpNm_SoAdIfTxConfirmation` called by the SoAd and with result passed by SoAd ⌋()

8.4.2 UdpNm_SoAdIfRxIndication

[SWS_UdpNm_00231]⌈

Service Name	UdpNm_SoAdIfRxIndication	

Syntax	`void UdpNm_SoAdIfRxIndication (`
	`  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)	RxPdu Id	ID of the received PDU.
	Pdu InfoPtr	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.	
Available via	UdpNm.h	

⌋()

The callback function `UdpNm_SoAdIfRxIndication` called by the SoAd and implemented by the UdpNm module. It is called in case of a receive indication event of the SoAd.

The value passed to UdpNm via the API parameter udpNmRxPduId shall refer to the UdpNm channel handle, i.e. a mapping from PduId to UdpNm channel handle is not necessary.

8.4.3 UdpNm_SoAdIfTriggerTransmit

[SWS_UdpNm_91001]⌈

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

⌋()

[SWS_UdpNm_00377] { OBSOLETE replaced by SWS_UdpNm_00495 } ⌈ If `UdpNmComUserDataSupport` is enabled the UdpNm shall collect the NM User Data from the referenced NM I-PDU by calling `PduR_UdpNmTriggerTransmit` and combine the user data with the further NM bytes within the call of `UdpNm_SoAdIfTriggerTransmit`. ⌋(RS_Nm_02503)

[SWS_UdpNm_00493] {DRAFT} ⌈ If `UdpNmPnEnabled` is `TRUE`, and either `UdpNmSynchronizedPncShutdownEnabled` is set to `TRUE` and no requests for synchronized PNC shutdown are pending or `UdpNmSynchronizedPncShutdownEnabled` is set to `FALSE`, then the UdpNm module shall perform the following actions in the given order within the call of `UdpNm_TriggerTransmit`:

- Call Nm_PncBitVectorTxIndication(<NM-channel>, <buffer to store the unfiltered PNC bit vector of aggregated internal PNC requests>) to indicate the transmission request and to retrieve internal PNC requests
- Copy the received PNC bit vector for internal PNC requests to the NM-PDU by considering NmPncBitVectorOffset and NmPncBitVectorLength of the corresponding NM-channel
- If user data is enabled, fetch the available data (either from Com if UdpNmComUserDataSupport is enabled or from internal storage) and copy the data in the user data range of the NM-PDU.

⌋(RS_Nm_02546, RS_Nm_02519, RS_Nm_02547)

[SWS_UdpNm_00480] { OBSOLETE replaced by SWS_UdpNm_00494 } ⌈ If UdpNmSynchronizedPncShutdownEnabled is set to TRUE and requests for synchronized PNC shutdown are pending, then the UdpNm module shall set for this message additionally the following data beneath the normal data within the call of UdpNm_TriggerTransmit:

- Set the PNSR bit in the CBV to 1
- Overwrite the PN information in the user data (after NM User Data has been fetched, if UdpNmComUserDataSupport is enabled) by setting bits that correspond to PNC IDs stored as pending request for a synchronized PNC shutdown to 1 and all other bits to 0.⌋(RS_Nm_02545)

[SWS_UdpNm_00494] {DRAFT} ⌈ If UdpNmSynchronizedPncShutdownEnabled is set to TRUE and requests for synchronized PNC shutdown are pending, then the UdpNm module shall set for this message additionally the following data beneath the normal data within the call of UdpNm_TriggerTransmit:

- Set the PNSR bit in the CBV to 1
- If user data is enabled, fetch the available data (either from Com if UdpNmComUserDataSupport is enabled or from internal storage) and copy the data in the user data range of the NM-PDU
- Write the PNC bit vector with respect to NmPncBitVectorOffset and NmPncBitVectorLength of the corresponding NM-channel by setting bits that corresponds to PNC IDs stored as pending request for a synchronized PNC shutdown to 1 and all other bits to 0 ⌋(RS_Nm_02545)

Note: The UdpNm modul has to aggregate all PNCs which were indicated for a synchronized PNC shutdown and transfer the pncId's to a byte array (PN C vit vector). Each bit (PNC bit) of the PN C vit vector represent a particular PNC. The byteIndex and bitindex within the PN Info range of PNC bit shall be determined as follows:

- byteIndex = (PncId div 8) - NmPncBitVectorOffset
- bitIndex = (PncId mod 8)

[SWS_UdpNm_00495] ⌈ If UdpNm_SoAdIfTriggerTransmit is called and UdpNmComUserDataSupport is enabled, UdpNm shall collect the NM User Data from the referenced NM I-PDU by calling PduR_UdpNmTriggerTransmit and copy the data to the user data range of the NM-PDU. ⌋(RS_Nm_02503)

[SWS_UdpNm_00378] ⌈ The function `UdpNm_SoAdIfTriggerTransmit` shall copy the NM PDU data of the according NM PDU requested by TxPduId. ⌋()

Note: The function `UdpNm_SoAdIfTriggerTransmit` might be called by the SoAd in an interrupt context.

8.5 Scheduled Functions

8.5.1 UdpNm_MainFunction_<Instance Id>

[SWS_UdpNm_00234]⌈

Service Name	UdpNm_MainFunction<Instance_Id>

Syntax	`void UdpNm_MainFunction<Instance_Id> (`
	`  void`
	`)`

Service ID [hex]	0x13
Description	Main function of the UdpNm which processes the algorithm describes in that document. E.g.: UdpNm_MainFunction_0() represents the UdpNm instance for the UDP channel 0 UdpNm_MainFunction_1() represents the UdpNm instance for the UDP channel 1
Available via	SchM_UdpNm.h

⌋()

8.6 Expected Interfaces

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

8.6.1 Mandatory Interfaces

This chapter defines all interfaces which are required to fulfill the core functionality of the module.
[SWS_UdpNm_91007]⌈

API Function	***Header File***	***Description***
Det_Report-RuntimeError	Det.h	Service to report runtime errors. If a callout has been configured then this callout shall be called.
Nm_BusSleep-Mode	Nm.h	Notification that the network management has entered Bus-Sleep Mode.
Nm_Network-	Nm.h	Notification that the network management has entered Network Mode.

Mode		
Nm_Network-StartIndication	Nm.h	Notification that a NM-message has been received in the Bus-Sleep Mode, what indicates that some nodes in the network have already entered the Network Mode.
Nm_Prepare-BusSleepMode	Nm.h	Notification that the network management has entered Prepare Bus-Sleep Mode.
SoAd_If-Transmit	SoAd.h	Requests transmission of a PDU.

⌋()

8.6.2 Optional Interfaces

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

[SWS_UdpNm_91006]⌈

API Function	***Header File***	***Description***
Det_ReportError	Det.h	Service to report development errors.
Nm_CarWakeUp-Indication	Nm.h	This function is called by a <Bus>Nm to indicate reception of a CWU request.
Nm_CoordReady-ToSleep-Cancellation	Nm.h	Cancels an indication, when the NM Coordinator Sleep Ready bit in the Control Bit Vector is set back to 0.
Nm_CoordReady-ToSleepIndication	Nm.h	Sets an indication, when the NM Coordinator Sleep Ready bit in the Control Bit Vector is set
Nm_Forward-SynchronizedPnc-Shutdown	Nm.h	Notification that the network management has received a PN shutdown message on a particular NM-channel. This is used to grant a nearly synchronized PNC shutdown across the entire PN topology.
Nm_PduRx-Indication	Nm.h	Notification that a NM message has been received.
Nm_PncBit-VectorRx-Indication (draft)	Nm.h	Indication that a bus specific network management has received a NM message on a particular NM-channel that contain a PNC bit vector. This is used to aggregate the external PNC requests. The function evaluate if a relevant PNC request (PNC bit set to '1') is available in the given PNC bit vector. If a relevant PNC request is available (PNC bit passes the PNC bit vector filter), then the RelevantPncRequestDetectedPtr refers to a boolean with value set to TRUE. Otherwise refer to booelan with value set to FALSE. RelevantPncRequestDetectedPtr is evaluated by the callee <Bus>Nm module to qualify the further processing of the received NM-PDU. **Tags:** atp.Status=draft
Nm_PncBit-VectorTx-Indication (draft)	Nm.h	Function called by <Bus>Nms to request the aggregated internal PNC requests for transmission within the Nm message. **Tags:** atp.Status=draft
Nm_Remote-SleepCancellation	Nm.h	Notification that the network management has detected that not all other nodes on the network are longer ready to enter Bus-Sleep Mode.
Nm_Remote-SleepIndication	Nm.h	Notification that the network management has detected that all other nodes on the network are ready to enter Bus-Sleep Mode.
Nm_Repeat-Message-Indication	Nm.h	Service to indicate that an NM message with set Repeat Message Re- quest Bit has been received. This is needed for node detection and the Dynamic PNC-to-channel-mapping feature.
Nm_State-	Nm.h	Notification that the state of the lower layer <BusNm> has changed.

Change-Notification		
Nm_TxTimeout-Exception	Nm.h	Service to indicate that an attempt to send an NM message failed.
PduR_UdpNmRx-Indication	PduR_Udp Nm.h	Indication of a received PDU from a lower layer communication interface module.
PduR_UdpNm-TriggerTransmit	PduR_Udp Nm.h	Within this API, the upper layer module (called module) shall check whether the available data fits into the buffer size reported by Pdu InfoPtr->SduLength. If it fits, it shall copy its data into the buffer provided by PduInfoPtr->SduDataPtr and update the length of the actual copied data in PduInfoPtr->SduLength. If not, it returns E_ NOT_OK without changing PduInfoPtr.
PduR_UdpNmTx-Confirmation	PduR_Udp Nm.h	The lower layer communication interface module confirms the transmission of a PDU, or the failure to transmit a PDU.

⌋()

8.6.3 Configurable interfaces

Not applicable

8.7 Service Interfaces

Not applicable

8.8 UML State chart diagram

The following figure shows an UML state diagram with respect to the API specification. Mode change related transitions are denoted in green, error handling related transitions in red and optional node detection related transitions in blue.

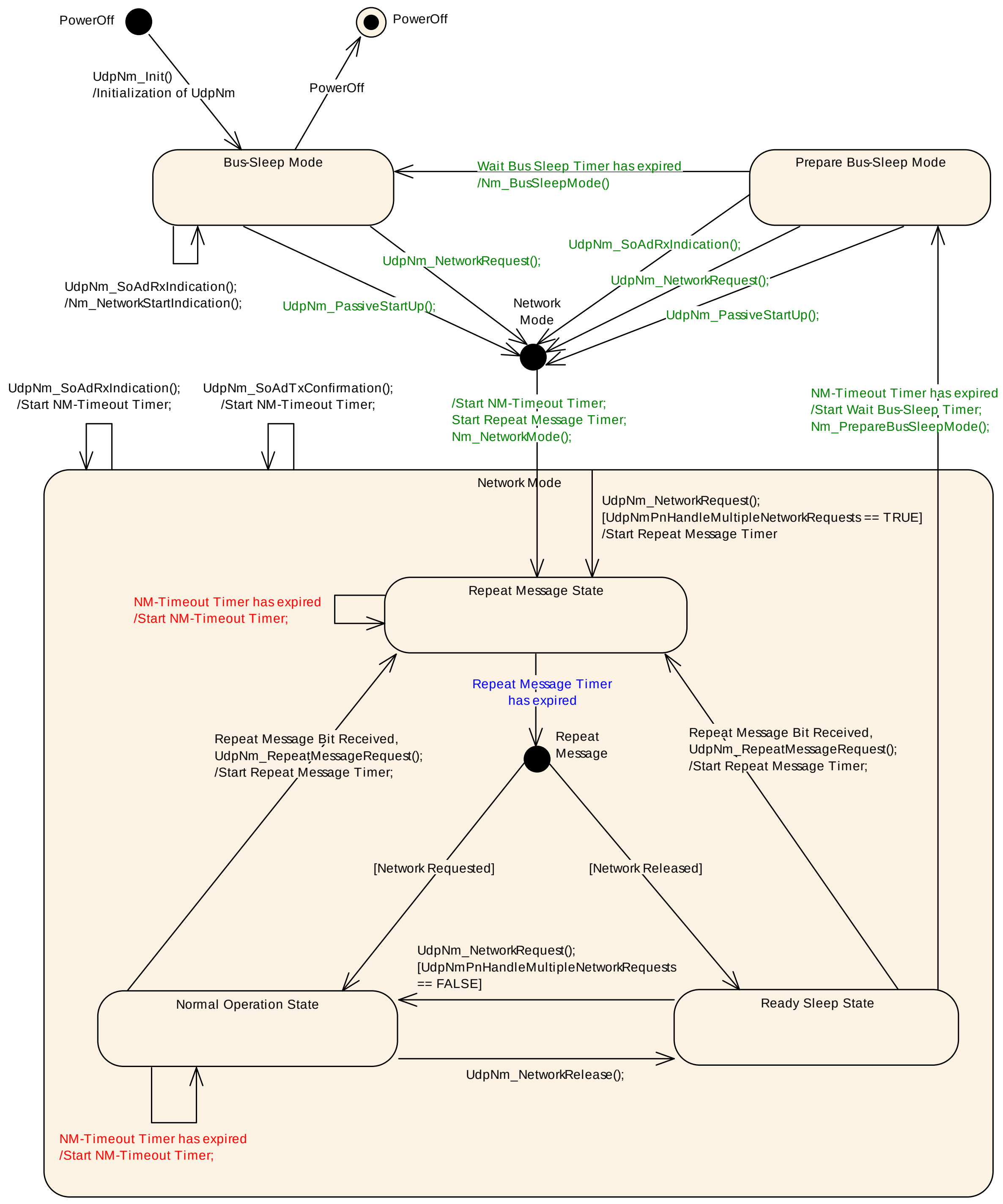
PowerOff
PowerOff
UdpNm_Init()
/Initialization of UdpNm
PowerOff
Bus-Sleep Mode
Wait Bus Sleep Timer has expired
/Nm_BusSleepMode()
Prepare Bus-Sleep Mode
UdpNm_SoAdRxIndication();
/Nm_NetworkStartIndication();
UdpNm_NetworkRequest();
UdpNm_SoAdRxIndication();
UdpNm_PassiveStartUp();
UdpNm_NetworkRequest();
Network
Mode
UdpNm_PassiveStartUp();
UdpNm_SoAdRxIndication();
/Start NM-Timeout Timer;
UdpNm_SoAdTxConfirmation();
/Start NM-Timeout Timer;
/Start NM-Timeout Timer;
Start Repeat Message Timer;
Nm_NetworkMode();
NM-Timeout Timer has expired
/Start Wait Bus-Sleep Timer;
Nm_PrepareBusSleepMode();
Network Mode
UdpNm_NetworkRequest();
[UdpNmPnHandleMultipleNetworkRequests == TRUE]
/Start Repeat Message Timer
NM-Timeout Timer has expired
/Start NM-Timeout Timer;
Repeat Message State
Repeat Message Timer
has expired
Repeat Message Bit Received,
UdpNm_RepeatMessageRequest();
/Start Repeat Message Timer;
Repeat
Message
Repeat Message Bit Received,
UdpNm_RepeatMessageRequest();
/Start Repeat Message Timer;
[Network Requested]
[Network Released]
UdpNm_NetworkRequest();
[UdpNmPnHandleMultipleNetworkRequests
== FALSE]
Normal Operation State
Ready Sleep State
UdpNm_NetworkRelease();
NM-Timeout Timer has expired
/Start NM-Timeout Timer;

Figure 4: State chart diagram.

9 Sequence diagrams and Transition Tables

9.1 UdpNmTransmission

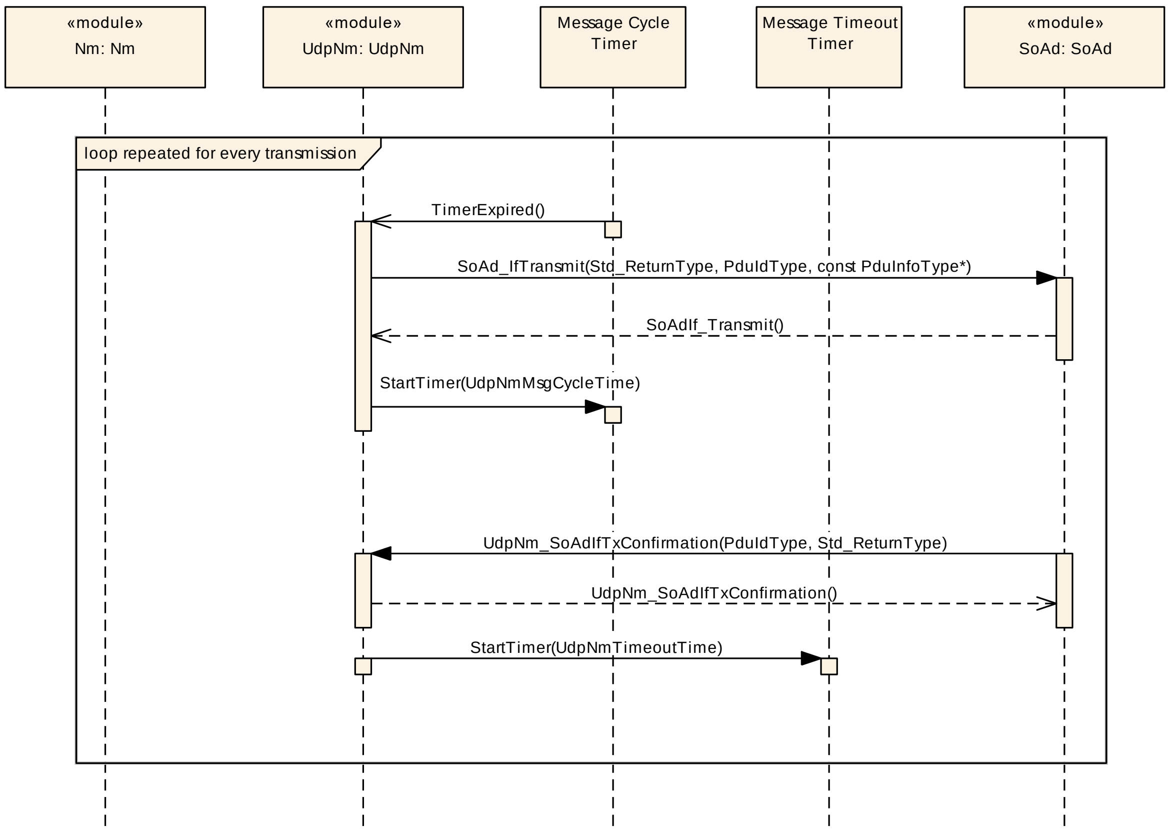
«module»
Nm: Nm
«module»
UdpNm: UdpNm
Message Cycle Timer
Message Timeout Timer
«module»
SoAd: SoAd
loop repeated for every transmission
TimerExpired()
SoAd_IfTransmit(Std_ReturnType, PduIdType, const PduInfoType*)
SoAdIf_Transmit()
StartTimer(UdpNmMsgCycleTime)
UdpNm_SoAdIfTxConfirmation(PduIdType, Std_ReturnType)
UdpNm_SoAdIfTxConfirmation()
StartTimer(UdpNmTimeoutTime)

Figure 5: Sequence diagram – PDU transmission.

9.2 UdpNm Reception

Call direction	*Action/Decision*	*Description*
`SoAd->UdpNm`	`UdpNm_SoAdIfRxIndication()`	

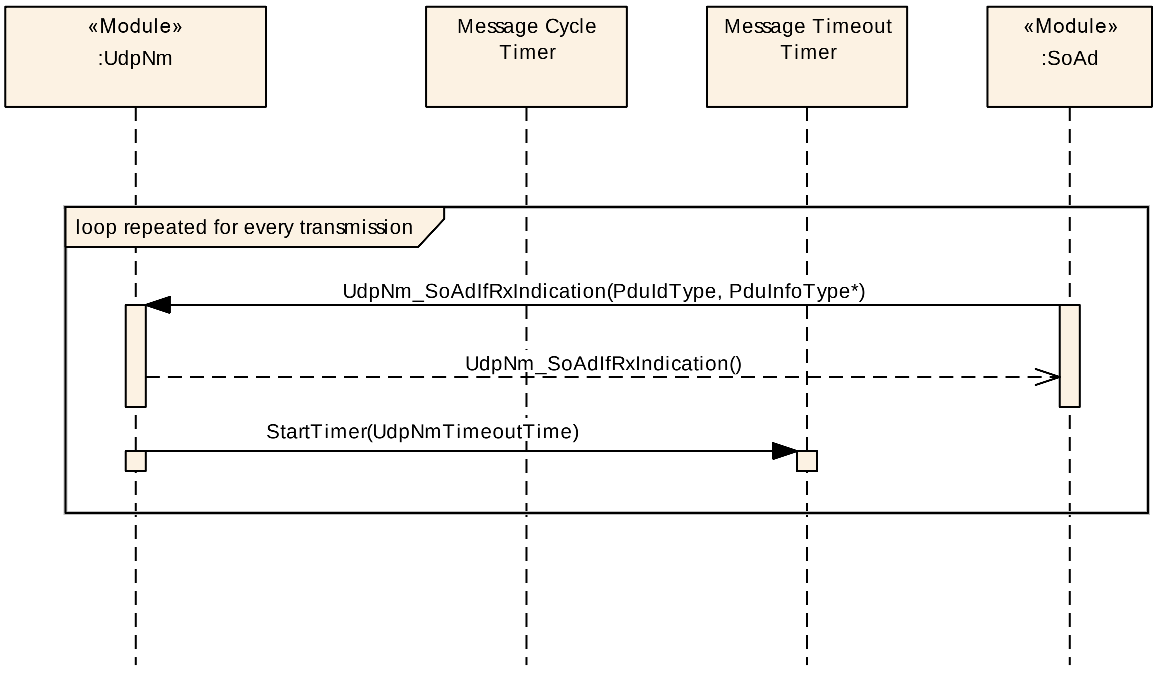
«Module»
:UdpNm
Message Cycle Timer
Message Timeout Timer
«Module»
:SoAd
loop repeated for every transmission
UdpNm_SoAdIfRxIndication(PduIdType, PduInfoType*)
UdpNm_SoAdIfRxIndication()
StartTimer(UdpNmTimeoutTime)

Figure 6: Sequence diagram – PDU reception.

10 Configuration specification

In general, this chapter defines configuration parameters and their clustering into containers. In order to support the specification chapter 10.1 describes fundamentals. It also specifies a template (table) to be use for the parameter specification. Chapter 10.1 is intended to remain in the specification document to ensure comprehensiveness.

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

Chapter 10.3 specifies published information of module UdpNm.

10.1 How to read this chapter

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

10.2 Containers and configuration parameters

The configuration parameters as defined in this chapter are used to create a data model for an AUTOSAR tool chain. The realization in the code is implementation specific.

The configuration parameters as defined in this chapter are used to create a data model for an AUTOSAR tool chain. The realization in the code is implementation specific.

The configuration parameters are divided into parameters used to enable features, parameters affecting all instances of the UdpNm and parameters affecting the respective instances of the UdpNm.

[SWS_UdpNm_00026] ⌈ All configuration items shall be located outside the kernel of the module. ⌋()

[SWS_UdpNm_00202] ⌈ The container `UdpNm_ChannelConfig` specifies configuration parameter that shall be located in a data structure of type `UdpNm_ConfigType`. ⌋()

[SWS_UdpNm_00203] ⌈ Runtime configurable parameters listed in container `UdpNm_ChannelConfig` shall be configurable for each NM-cluster separately. ⌋()

10.2.1 UdpNm

SWS Item	**ECUC_UdpNm_00088 :**
Module Name	*UdpNm*
Module Description	--
Post-Build Variant Support	true
Supported Config Variants	VARIANT-LINK-TIME, VARIANT-PRE-COMPILE

Included Containers		
Container Name	***Multiplicity***	***Scope / Dependency***
UdpNmGlobalConfig	1	This container contains all global configuration parameters of UDP NM configured from the CanTrcv Module perspective.

10.2.2 UdpNmGlobalConfig

SWS Item	**ECUC_UdpNm_00001 :**
Container Name	UdpNmGlobalConfig
Parent Container	UdpNm
Description	This container contains all global configuration parameters of UDP NM configured from the CanTrcv Module perspective.
Configuration Parameters	

SWS Item	**ECUC_UdpNm_00006 :**		
Name	UdpNmBusSynchronizationEnabled		
Parent Container	UdpNmGlobalConfig		
Description	Pre-processor switch for enabling bus synchronization support. This feature is required for gateway nodes only. It must not be defined if UdpNmPassiveModeEnabled==true. This parameter shall be derived from NmBusSynchronizationEnabled.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00013 :**		
Name	UdpNmComControlEnabled		
Parent Container	UdpNmGlobalConfig		
Description	Pre-processor switch for enabling the Communication Control support.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	

Scope / Dependency	scope: local dependency: calculationFormula = If (UdpNmPassiveModeEnabled == False) then Equal(NmComControlEnabled) else Equal(False)		

SWS Item	**ECUC_UdpNm_00055 :**		
Name	UdpNmComUserDataSupport		
Parent Container	UdpNmGlobalConfig		
Description	Enable/disable the user data support.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local dependency: If UdpNmPassiveModeEnabled == True OR if all bytes of the NM PDU are used for NM System Bytes and for the PNC bit vector and no space is left for user data, then UdpNmComUserDataSupport shall be set to False .		

SWS Item	**ECUC_UdpNm_00059 :**		
Name	UdpNmCoordinatorSyncSupport		
Parent Container	UdpNmGlobalConfig		
Description	Enables/disables the coordinator synchronization support.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local dependency: UdpNmCoordinatorSyncSupport has to be set to FALSE if UdpNmPassiveModeEnabled is set to TRUE.		

SWS Item	**ECUC_UdpNm_00002 :**		
Name	UdpNmDevErrorDetect		
Parent Container	UdpNmGlobalConfig		
Description	Switches the development error detection and notification on or off. • true: detection and notification is enabled. • false: detection and notification is disabled.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00094 :**
Name	UdpNmDynamicPncToChannelMappingSupport
Parent Container	UdpNmGlobalConfig

Description	Precompile time switch to enable the dynamic PNC-to-channel-mapping handling. False: Dynamic PNC-to-channel-mapping is disabled True: Dynamic PNC-to-channel-mapping is enabled **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_UdpNm_00009 :**		
Name	UdpNmImmediateRestartEnabled		
Parent Container	UdpNmGlobalConfig		
Description	Pre-processor switch for enabling the immediate transmission of a NM PACKET upon bus-communication request in Prepare-Bus-Sleep mode. Must not be defined if UdpNmPassiveModeEnabled== true.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00014 :**		
Name	UdpNmNumberOfChannels		
Parent Container	UdpNmGlobalConfig		
Description	Number of NM channels allowed within one ECU.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 255		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00010 :**		
Name	UdpNmPassiveModeEnabled		
Parent Container	UdpNmGlobalConfig		
Description	Pre-processor switch for enabling support of the Passive Mode.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	

Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00011 :**		
Name	UdpNmPduRxIndicationEnabled		
Parent Container	UdpNmGlobalConfig		
Description	Pre-processor switch for enabling the PDU Rx Indication. This parameter shall be derived from NmPduRxIndicationEnabled.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00066 : (Obsolete)**		
Name	UdpNmPnEiraCalcEnabled		
Parent Container	UdpNmGlobalConfig		
Description	Specifies if UdpNm calculates the PN request information for internal and external requests. (EIRA) true: PN request are calculated false: PN request are not calculated **Tags:** atp.Status=obsolete		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: only available if UdpNmPnEnabled == true for at least one UdpNm Channel		

SWS Item	**ECUC_UdpNm_00065 : (Obsolete)**		
Name	UdpNmPnResetTime		
Parent Container	UdpNmGlobalConfig		
Description	Specifies the runtime of the reset timer in seconds. This reset time is valid for the reset of PN requests in the EIRA and in the ERA. The value shall be the same for every channel. Thus it is a global config parameter. **Tags:** atp.Status=obsolete		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0.001 .. 65.535]		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration	***Pre-compile time***	X	VARIANT-PRE-COMPILE

Class	***Link time***	X	VARIANT-LINK-TIME
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: only available if UdpNmPnEnabled == true for at least one UdpNm Channel.		

SWS Item	**ECUC_UdpNm_00098 :**		
Name	UdpNmPnShutdownMessageRetransmissionDuration		
Parent Container	UdpNmGlobalConfig		
Description	Specifies the duration in seconds of retransmission phase of a PN shutdown message. A retransmission shall be performed per affected NM channel, as long as the PN shutdown message could not be successfully sent and the retransmission timer is running. The value shall be a multiple integral of UdpNmMainFunctionPeriod.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0.001 .. 65.535]		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: * Only valid if UdpNmSynchronizedPncShutdownEnabled == TRUE * UdpNmPnShutdownMessageRetransmissionDuration ≤ UdpNmPnResetTime		

SWS Item	**ECUC_UdpNm_00096 :**		
Name	UdpNmPnSyncShutdownErrorReactionEnabled		
Parent Container	UdpNmGlobalConfig		
Description	Pre-processor switch for enabling reaction, if a top-level PNC coordinator received a PN shutdown message on a NM-channel which refer to a ComM channel that is actively coordinated by a PNC gateway. **Tags:** atp.Status=draft		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local dependency: This parameter shall only be set to TRUE if UdpNmSynchronizedPncShutdownEnabled is set to TRUE on at least one channel.		

SWS Item	**ECUC_UdpNm_00005 :**

Name	UdpNmRemoteSleepIndEnabled		
Parent Container	UdpNmGlobalConfig		
Description	Pre-processor switch for enabling remote sleep indication support. This feature is required for gateway nodes only. It must not be defined if UdpNmPassiveModeEnabled==true. This parameter shall be derived from NmRemoteSleepIndEnabled.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00012 :**		
Name	UdpNmStateChangeIndEnabled		
Parent Container	UdpNmGlobalConfig		
Description	Pre-processor switch for enabling the UDP NM state change notification. This parameter shall be derived from NmStateChangeIndEnabled.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00004 :**		
Name	UdpNmUserDataEnabled		
Parent Container	UdpNmGlobalConfig		
Description	Pre-processor switch for enabling user data support. This parameter shall be derived from NmUserDataEnabled.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local dependency: UdpNmUserDataEnabled shall be set to FALSE, if all bytes of the NM PDU are used for NM System Bytes and for the PNC bit vector and no space is left for user data. Otherwise the parameter shall be set according the following formular: calculationFormula =Equal(NmUserDataEnabled).		

SWS Item	**ECUC_UdpNm_00003 :**
Name	UdpNmVersionInfoApi
Parent Container	UdpNmGlobalConfig
Description	Pre-processor switch for enabling version info API support.
Multiplicity	1
Type	EcucBooleanParamDef
Default value	false
Post-Build Variant Value	false

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

SWS Item	**ECUC_UdpNm_00062 : (Obsolete)**		
Name	UdpNmPnEiraRxNSduRef		
Parent Container	UdpNmGlobalConfig		
Description	Reference to a Pdu in the COM-Stack. Only one SduRef is required for UdpNm because the EIRA is the aggregation over all Ethernet Channels. **Tags:** atp.Status=obsolete		
Multiplicity	0..1		
Type	Reference to [Pdu]		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	true		
Multiplicity Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	

Scope / Dependency	scope: local dependency: only available if UdpNmPnEnabled == true for at least one UdpNm Channel

Included Containers		
Container Name	***Multiplicity***	***Scope / Dependency***
UdpNmChannelConfig	1..*	This container contains the channel-specific configuration parameters of the UdpNm.
UdpNmPnInfo	0..1	PN information configuration **Tags:** atp.Status=obsolete

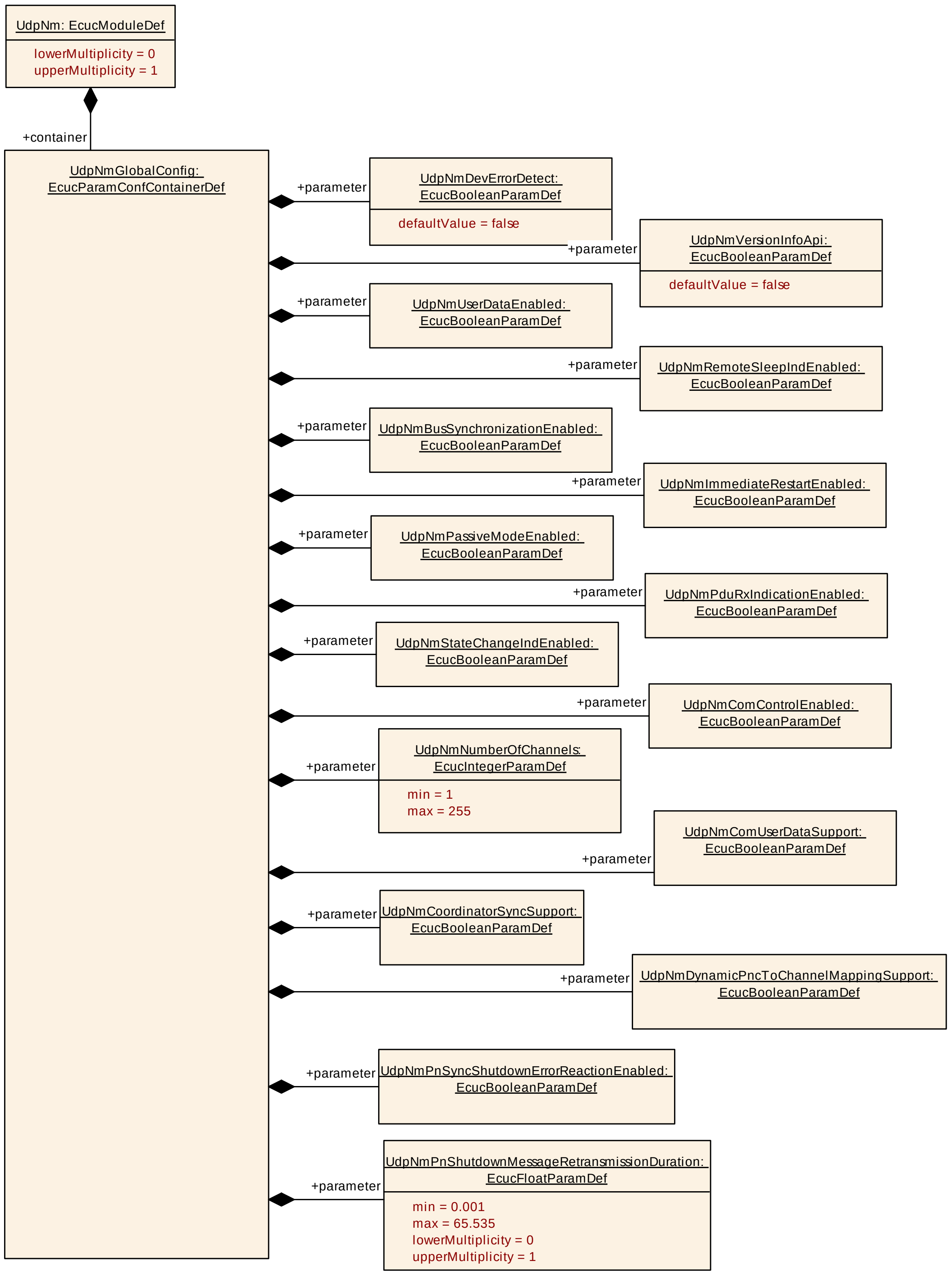
UdpNm: EcucModuleDef
lowerMultiplicity = 0
upperMultiplicity = 1
+container
UdpNmGlobalConfig: EcucParamConfContainerDef
+parameter
UdpNmDevErrorDetect: EcucBooleanParamDef
defaultValue = false
UdpNmVersionInfoApi: EcucBooleanParamDef
defaultValue = false
UdpNmUserDataEnabled: EcucBooleanParamDef
UdpNmRemoteSleepIndEnabled: EcucBooleanParamDef
UdpNmBusSynchronizationEnabled: EcucBooleanParamDef
UdpNmImmediateRestartEnabled: EcucBooleanParamDef
UdpNmPassiveModeEnabled: EcucBooleanParamDef
UdpNmPduRxIndicationEnabled: EcucBooleanParamDef
UdpNmStateChangeIndEnabled: EcucBooleanParamDef
UdpNmComControlEnabled: EcucBooleanParamDef
UdpNmNumberOfChannels: EcucIntegerParamDef
min = 1
max = 255
UdpNmComUserDataSupport: EcucBooleanParamDef
UdpNmCoordinatorSyncSupport: EcucBooleanParamDef
UdpNmDynamicPncToChannelMappingSupport: EcucBooleanParamDef
UdpNmPnSyncShutdownErrorReactionEnabled: EcucBooleanParamDef
UdpNmPnShutdownMessageRetransmissionDuration: EcucFloatParamDef
min = 0.001
max = 65.535
lowerMultiplicity = 0
upperMultiplicity = 1

Figure 9: Diagram: UdpNmGlobalConfig

10.2.3 UdpNmChannelConfig

SWS Item	**ECUC_UdpNm_00017 :**
Container Name	UdpNmChannelConfig
Parent Container	UdpNmGlobalConfig
Description	This container contains the channel-specific configuration parameters of the UdpNm.
Configuration Parameters	

SWS Item	**ECUC_UdpNm_00074 :**		
Name	UdpNmActiveWakeupBitEnabled		
Parent Container	UdpNmChannelConfig		
Description	Enables/Disables the handling of the Active Wakeup Bit in the UdpNm module.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: This parameter is only valid if UdpNmPassiveModeEnabled is False.		

SWS Item	**ECUC_UdpNm_00089 :**		
Name	UdpNmAllNmMessagesKeepAwake		
Parent Container	UdpNmChannelConfig		
Description	Specifies if UdpNm drops irrelevant NM PDUs. false: Only NM PDUs with a PNI bit = true and containing a PN request for this ECU triggers the standard RX indication handling true: Every NM PDU triggers the standard RX indication handling		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	

Scope / Dependency	scope: local dependency: (DRAFT) only valid if NmPnEiraCalcEnabled == true or NmPnEraCalcEnabled == true (OBSOLETE): only available if UdpNmPnEnabled == true		

SWS Item	**ECUC_UdpNm_00087 :**		
Name	UdpNmCarWakeUpBitPosition		
Parent Container	UdpNmChannelConfig		
Description	Specifies the Bit position of the CWU within the NM PDU.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 7		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: only available if UdpNmCarWakeUpRxEnabled == TRUE		

SWS Item	**ECUC_UdpNm_00086 :**		
Name	UdpNmCarWakeUpBytePosition		
Parent Container	UdpNmChannelConfig		
Description	Specifies the Byte position of the CWU within the NM PDU.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 7		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: only available if UdpNmCarWakeUpRxEnabled == TRUE UdpNmCarWakeupBytePosition ≥ number of enabled system bytes (CBV, NID)		

SWS Item	**ECUC_UdpNm_00077 :**
Name	UdpNmCarWakeUpFilterEnabled
Parent Container	UdpNmChannelConfig
Description	If CWU filtering is supported, only the CWU bit within the NM PDU with source node identifier UdpNmCarWakeUpFilterNodeId is considered as CWU request. FALSE - CWU filtering is not supported TRUE - CWU filtering is supported.

Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: only available if UdpNmCarWakeUpRxEnabled == TRUE		

SWS Item	**ECUC_UdpNm_00078 :**		
Name	UdpNmCarWakeUpFilterNodeId		
Parent Container	UdpNmChannelConfig		
Description	Source node identifier for CWU filtering. If CWU filtering is supported, only the CWU bit within the NM PDU with source node identifier UdpNmCarWakeUpFilterNodeId is considered as CWU request.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: only available if UdpNmCarWakeUpFilterEnabled == TRUE		

SWS Item	**ECUC_UdpNm_00076 :**		
Name	UdpNmCarWakeUpRxEnabled		
Parent Container	UdpNmChannelConfig		
Description	Enables or disables support of CarWakeUp bit evaluation in received NM PDUs. FALSE - CarWakeUp not supported. TRUE - CarWakeUp supported.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_UdpNm_00095 :**
Name	UdpNmDynamicPncToChannelMappingEnabled
Parent Container	UdpNmChannelConfig

Description	Channel-specific parameter to enable the dynamic PNC-to-channel-mapping feature. False: Dynamic PNC-to-channel-mapping is disabled True: Dynamic PNC-to-channel-mapping is enabled **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: Shall only be TRUE if UdpNmDynamicPncToChannelMappingSupport is TRUE		

SWS Item	**ECUC_UdpNm_00079 :**		
Name	UdpNmImmediateNmCycleTime		
Parent Container	UdpNmChannelConfig		
Description	Defines the immediate NM PDU cycle time in seconds which is used for UdpNmImmediateNmTransmissions NM PDU transmissions.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0.001 .. 65.535]		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: This parameter is only valid if UdpNmImmediateNmTransmissions is greater one.		

SWS Item	**ECUC_UdpNm_00075 :**		
Name	UdpNmImmediateNmTransmissions		
Parent Container	UdpNmChannelConfig		
Description	Defines the number of immediate NM PDUs which shall be transmitted. If the value is zero no immediate NM PDUs are transmitted. The cycle time of immeditate NM PDUs is defined by UdpNmImmediateNmCycleTime.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	

Scope / Dependency	scope: local dependency: If UdpNmImmediateRestartEnabled = true then UdpNmImmediateNmTransmissions = 0 If UdpNmPnHandleMultipleNetworkRequests == True then UdpNmImmediateNmTransmissions > 0		

SWS Item	**ECUC_UdpNm_00032 :**		
Name	UdpNmMainFunctionPeriod		
Parent Container	UdpNmChannelConfig		
Description	Call cycle of UdpNm_MainFunction_x for the respective instance in [s].		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	]0 .. INF[		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00029 :**		
Name	UdpNmMsgCycleOffset		
Parent Container	UdpNmChannelConfig		
Description	Time offset in the periodic transmission node. It determines the start delay of the transmission. < UdpNmMsgCycleTime This parameter is only valid if UdpNmPassiveModeEnabled is disabled.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. 65.535]		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00028 :**		
Name	UdpNmMsgCycleTime		
Parent Container	UdpNmChannelConfig		
Description	Period of a NM-message. It determines the periodic rate and is the basis for transmit scheduling. NmTimeoutTime = n * UdpNmMsgCycleTime This parameter is only valid if UdpNmPassiveModeEnabled is disabled.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0.001 .. 65.535]		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_UdpNm_00090 :**

Name	UdpNmNodeDetectionEnabled		
Parent Container	UdpNmChannelConfig		
Description	Pre-processor switch for enabling the node detection support. This parameter shall be derived from NmNodeDetectionEnabled. This parameter shall only be enabled if UdpNmNodeIdEnabled == true. If(UdpNmPduCbvPosition != UDPNM_PDU_OFF) then Equal(NmNodeDetectionEnabled) else Equal(False).		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: Not available if UdpNmPassiveModeEnabled		

SWS Item	**ECUC_UdpNm_00031 :**		
Name	UdpNmNodeId		
Parent Container	UdpNmChannelConfig		
Description	Node identifier of local node.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: This parameter is only relevant if UdpNmNodeIdEnabled == True.		

SWS Item	**ECUC_UdpNm_00091 :**		
Name	UdpNmNodeIdEnabled		
Parent Container	UdpNmChannelConfig		
Description	Pre-processor switch for enabling the source node identifier. This parameter shall be derived from NmNodeIdEnabled.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00026 :**
Name	UdpNmPduCbvPosition
Parent Container	UdpNmChannelConfig
Description	Defines the position of the control bit vector within the NM PACKET. The value of the parameter represents the location of the control bit vector in the NM PACKET (UDPNM_PDU_BYTE_0 means byte 0, UDPNM_PDU_BYTE_1 means byte 1, UDPNM_PDU_OFF means the control bit vector is not part of the NM PACKET)

	See also UdpNmPduNidPosition
	
	if (UdpNmPduCbvPosition != UDPNM_PDU_OFF && UdpNmPduNidPosition != UDPNM_PDU_OFF) then UdpNmPduCbvPosition != UdpNmPduNidPosition
	
	if (UdpNmPduCbvPosition != UDPNM_PDU_OFF && UdpNmPduNidPosition == UDPNM_PDU_OFF) then UdpNmPduCbvPosition = UDPNM_PDU_BYTE0		

Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	UDPNM_PDU_BYTE_0	--	
	UDPNM_PDU_BYTE_1	--	
	UDPNM_PDU_OFF	--	
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00025 :**		
Name	UdpNmPduNidPosition		
Parent Container	UdpNmChannelConfig		
Description	Defines the position of the source node identifier within the NM PACKET. ImplementationType: UdpNm_PduPositionType The value of the parameter represents the location of the source node identifier in the NM PACKET (UDPNM_PDU_BYTE_0 means byte 0, UDPNM_PDU_BYTE_1 means byte 1, UDPNM_PDU_OFF means source node identifier is not part of the NM PACKET) See also UdpNmPduCbvPosition if (UDPNM_PDU_NID_POSITION != UDPNM_PDU_OFF && UDPNM_PDU_CBV_POSITION != UDPNM_PDU_OFF) then UDPNM_PDU_NID_POSITION != UDPNM_PDU_CBV_POSITION if (UDPNM_PDU_NID_POSITION != UDPNM_PDU_OFF && UDPNM_PDU_CBV_POSITION == UDPNM_PDU_OFF) then UDPNM_PDU_IND_POSITION = UDPNM_PDU_BYTE0		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	UDPNM_PDU_BYTE_0	Byte 0 is used.	
	UDPNM_PDU_BYTE_1	Byte 1 is used.	
	UDPNM_PDU_OFF	Node Identification is not used.	
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00061 :**
Name	UdpNmPnEnabled
Parent Container	UdpNmChannelConfig
Description	Enables or disables support of partial networking. false: Partial networking Range not supported

	true: Partial networking supported		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00060 : (Obsolete)**		
Name	UdpNmPnEraCalcEnabled		
Parent Container	UdpNmChannelConfig		
Description	Specifies if UdpNm calculates the PN request information for external requests. (ERA) false: PN request are not calculated true: PN request are calculated. **Tags:** atp.Status=obsolete		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: only available if UdpNmPnEnabled == true		

SWS Item	**ECUC_UdpNm_00063 :**		
Name	UdpNmPnHandleMultipleNetworkRequests		
Parent Container	UdpNmChannelConfig		
Description	false: UdpNm_NetworkRequest is ignored in NO. true: UdpNm_NetworkRequest triggers a change from NO to RM.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	

Scope / Dependency	scope: local dependency: only available if UdpNmPnEnabled == true		

SWS Item	**ECUC_UdpNm_00023 :**		
Name	UdpNmRemoteSleepIndTime		
Parent Container	UdpNmChannelConfig		
Description	Timeout for Remote Sleep Indication. It defines the time in [s] how long it shall take to recognize that all other nodes are ready to sleep. Typically it should be equal to: n * UdpNmMsgCycleTime, where n denotes the number of NM packets that are normally sent before Remote Sleep Indication is detected. The value of n decremented by one determines the amount of lost NM packets that can be tolerated by the Remote Sleep Indication procedure.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0.001 .. 65.535]		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00022 :**		
Name	UdpNmRepeatMessageTime		
Parent Container	UdpNmChannelConfig		
Description	Timeout for Repeat Message State. It defines the time in seconds how long the NM shall stay in the Repeat Message State.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0 .. 65.535]		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: UdpNmRepeatMessageTime = n * UdpNmMsgCycleTime; UdpNmRepeatMessageTime > UdpNmImmediateNmTransmissions * UdpNmImmediateNmCycleTime Typically it should be equal to: n * UdpNmMsgCycleTime, where n denotes the number of NM PDUs that are normally sent in the Repeat Message State. The value of n decremented by one determines the amount of lost NM PDUs that can be tolerated by the node detection procedure. The value 0 denotes that no Repeat Message State is configured. It means that Repeat Message State is transient what implicates that it is left immediately after entrance and in result no start-up stability is guaranteed and no node detection procedure is possible.		

SWS Item	**ECUC_UdpNm_00092 :**
Name	UdpNmRepeatMsgIndEnabled
Parent Container	UdpNmChannelConfig

Description	Enable/disable the notification that a RepeatMessageRequest bit has been received.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: UdpNmRepeatMsgIndEnabled = FALSE if UdpNmPassiveModeEnabled == TRUE or (UdpNmNodeDetectionEnabled == FALSE && UdpNmDynamicPncToChannelMappingEnabled == FALSE). UdpNmRepeatMsgIndEnabled = TRUE if UdpNmDynamicPncToChannelMappingEnabled == TRUE.		

SWS Item	**ECUC_UdpNm_00085 :**		
Name	UdpNmRetryFirstMessageRequest		
Parent Container	UdpNmChannelConfig		
Description	Specifies if first message request in UdpNm is repeated until accepted by SoAd.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: UdpNmRetryFirstMessageRequest = false if UdpNmPassiveModeEnabled == true		

SWS Item	**ECUC_UdpNm_00093 :**		
Name	UdpNmStayInPbsEnabled		
Parent Container	UdpNmChannelConfig		
Description	If this parameter is disabled Prepare Bus-Sleep Mode is left after UdpNmWaitBusSleepTime. If this parameter is enabled Prepare Bus-Sleep Mode can only be left if ECU is powered off or any restart reason applies.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00097 :**
Name	UdpNmSynchronizedPncShutdownEnabled

Parent Container	UdpNmChannelConfig		
Description	Specifies if UdpNm handle PN shutdown messages to support a synchronized PNC shutdown across a PN topology. This is only used for ECUs in the role of a top-level PNC coordinator or intermediate PNC coordinator. Thus, the PNC gateway functionality is enabled and therefore ERA calculation is used. FALSE: synchronized PNC shutdown is disabled TRUE: synchronized PNC shutdown is enabled **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: (OBSOLETE: Only available if UdpNmPnEnabled == TRUE and UdpNmPnEraCalcEnabled == TRUE.) DRAFT: Only available if UdpNmPnEnabled == TRUE and NmPnEraCalcEnabled == TRUE.		

SWS Item	**ECUC_UdpNm_00020 :**		
Name	UdpNmTimeoutTime		
Parent Container	UdpNmChannelConfig		
Description	Network Timeout for NM packets. It denotes the time in [s] how long the NM shall stay in the Network Mode before transition into Prepare Bus-Sleep Mode shall take place. It shall be equal for all nodes in the cluster. It shall be greater than UdpNmMsgCycleTime. Typically, it should be equal to: x * UdpNmMsgCycleTime, where n denotes the number of NM PACKET cycle times in the Ready Sleep State before transition into the Bus-Sleep Mode is initiated. The value of n decremented by one determines the amount of lost NM packets that can be tolerated by the coordination algorithm.		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	[0.002 .. 65.535]		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_UdpNm_00021 :**
Name	UdpNmWaitBusSleepTime

Parent Container	UdpNmChannelConfig		
Description	Timeout for bus calm down phase. It denotes the time in [s] how long the NM shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place. It shall be equal for all nodes in the cluster. It shall be long enough to empty all Tx-buffer empty.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0.001 .. 65.535]		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: In case UdpNmStayInPbsEnabled is disabled this parameter shall be mandatory.		

SWS Item	**ECUC_UdpNm_00018 :**		
Name	UdpNmComMNetworkHandleRef		
Parent Container	UdpNmChannelConfig		
Description	This reference points to the unique channel defined by the ComMChannel and provides access to the unique channel index value in ComMChannelId.		
Multiplicity	1		
Type	Symbolic name reference to [ComMChannel]		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_UdpNm_00073 : (Obsolete)**		
Name	UdpNmPnEraRxNSduRef		
Parent Container	UdpNmChannelConfig		
Description	Reference to a Pdu in the COM-Stack. The SduRef is required for every UdpNm Channel, because ERA is reported per channel. **Tags:** atp.Status=obsolete		
Multiplicity	0..1		
Type	Reference to [Pdu]		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: only available if UdpNmPnEnabled == true		

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
UdpNmRxPdu	1..*	This container describes the UdpNm RX PDU's.

UdpNmTxPdu	0..1	This container describes the UdpNm TX PDU's.
UdpNmUserDataTxPdu	0..1	Preprocessor switch for enabling the Tx path of Com User Data. Use case: Setting of NMUserData via SWC.

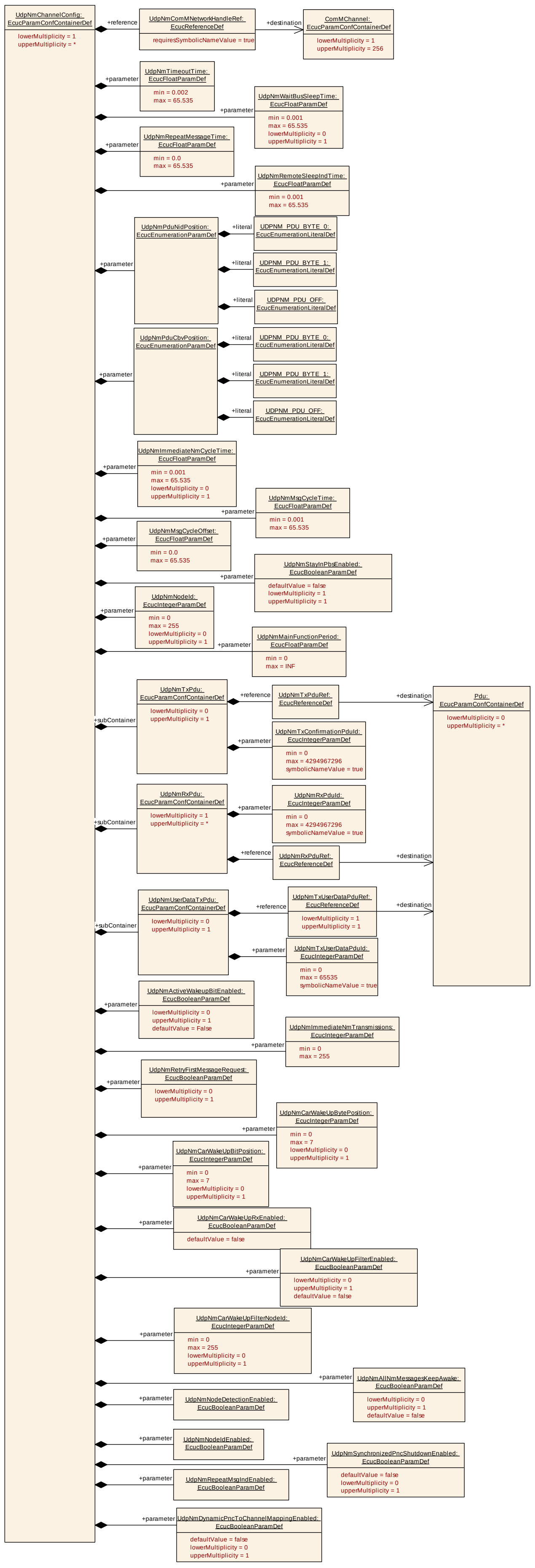
UdpNmChannelConfig: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+reference
UdpNmComMNetworkHandleRef: EcucReferenceDef
requiresSymbolicNameValue = true
+destination
ComMChannel: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = 256
+parameter
UdpNmTimeoutTime: EcucFloatParamDef
min = 0.002
max = 65.535
+parameter
UdpNmWaitBusSleepTime: EcucFloatParamDef
min = 0.001
max = 65.535
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
UdpNmRepeatMessageTime: EcucFloatParamDef
min = 0.0
max = 65.535
+parameter
UdpNmRemoteSleepIndTime: EcucFloatParamDef
min = 0.001
max = 65.535
+parameter
UdpNmPduNidPosition: EcucEnumerationParamDef
+literal
UDPNM_PDU_BYTE_0: EcucEnumerationLiteralDef
+literal
UDPNM_PDU_BYTE_1: EcucEnumerationLiteralDef
+literal
UDPNM_PDU_OFF: EcucEnumerationLiteralDef
+parameter
UdpNmPduCbvPosition: EcucEnumerationParamDef
+literal
UDPNM_PDU_BYTE_0: EcucEnumerationLiteralDef
+literal
UDPNM_PDU_BYTE_1: EcucEnumerationLiteralDef
+literal
UDPNM_PDU_OFF: EcucEnumerationLiteralDef
+parameter
UdpNmImmediateNmCycleTime: EcucFloatParamDef
min = 0.001
max = 65.535
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
UdpNmMsgCycleTime: EcucFloatParamDef
min = 0.001
max = 65.535
+parameter
UdpNmMsgCycleOffset: EcucFloatParamDef
min = 0.0
max = 65.535
+parameter
UdpNmStayInPbsEnabled: EcucBooleanParamDef
defaultValue = false
lowerMultiplicity = 1
upperMultiplicity = 1
+parameter
UdpNmNodeId: EcucIntegerParamDef
min = 0
max = 255
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
UdpNmMainFunctionPeriod: EcucFloatParamDef
min = 0
max = INF
+subContainer
UdpNmTxPdu: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 1
+reference
UdpNmTxPduRef: EcucReferenceDef
+destination
Pdu: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = *
+parameter
UdpNmTxConfirmationPduId: EcucIntegerParamDef
min = 0
max = 4294967296
symbolicNameValue = true
+subContainer
UdpNmRxPdu: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *
+parameter
UdpNmRxPduId: EcucIntegerParamDef
min = 0
max = 4294967296
symbolicNameValue = true
+reference
UdpNmRxPduRef: EcucReferenceDef
+destination
+subContainer
UdpNmUserDataTxPdu: EcucParamConfContainerDef
lowerMultiplicity = 0
upperMultiplicity = 1
+reference
UdpNmTxUserDataPduRef: EcucReferenceDef
lowerMultiplicity = 1
upperMultiplicity = 1
+destination
+parameter
UdpNmTxUserDataPduId: EcucIntegerParamDef
min = 0
max = 65535
symbolicNameValue = true
+parameter
UdpNmActiveWakeupBitEnabled: EcucBooleanParamDef
lowerMultiplicity = 0
upperMultiplicity = 1
defaultValue = False
+parameter
UdpNmImmediateNmTransmissions: EcucIntegerParamDef
min = 0
max = 255
+parameter
UdpNmRetryFirstMessageRequest: EcucBooleanParamDef
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
UdpNmCarWakeUpBytePosition: EcucIntegerParamDef
min = 0
max = 7
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
UdpNmCarWakeUpBitPosition: EcucIntegerParamDef
min = 0
max = 7
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
UdpNmCarWakeUpRxEnabled: EcucBooleanParamDef
defaultValue = false
+parameter
UdpNmCarWakeUpFilterEnabled: EcucBooleanParamDef
lowerMultiplicity = 0
upperMultiplicity = 1
defaultValue = false
+parameter
UdpNmCarWakeUpFilterNodeId: EcucIntegerParamDef
min = 0
max = 255
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
UdpNmAllNmMessagesKeepAwake: EcucBooleanParamDef
lowerMultiplicity = 0
upperMultiplicity = 1
defaultValue = false
+parameter
UdpNmNodeDetectionEnabled: EcucBooleanParamDef
+parameter
UdpNmNodeIdEnabled: EcucBooleanParamDef
+parameter
UdpNmSynchronizedPncShutdownEnabled: EcucBooleanParamDef
defaultValue = false
lowerMultiplicity = 0
upperMultiplicity = 1
+parameter
UdpNmRepeatMsgIndEnabled: EcucBooleanParamDef
+parameter
UdpNmDynamicPncToChannelMappingEnabled: EcucBooleanParamDef
defaultValue = false
lowerMultiplicity = 0
upperMultiplicity = 1

Figure 10: UdpNmChannelConfig

10.2.4 UdpNmRxPdu

SWS Item	**ECUC_UdpNm_00038 :**
Container Name	UdpNmRxPdu
Parent Container	UdpNmChannelConfig
Description	This container describes the UdpNm RX PDU's.
Configuration Parameters	

SWS Item	**ECUC_UdpNm_00043 :**		
Name	UdpNmRxPduId		
Parent Container	UdpNmRxPdu		
Description	ID of the RxPdu that will be used by a RxIndication of the lower layer.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 4294967296		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00039 :**		
Name	UdpNmRxPduRef		
Parent Container	UdpNmRxPdu		
Description	The reference to a PDU in the global PDU structure described in the AUTOSAR ECU Configuration Specification. This reference will be used by the UdpNm module to derive the PDU Id.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.5 UdpNmTxPdu

SWS Item	**ECUC_UdpNm_00036 :**
Container Name	UdpNmTxPdu
Parent Container	UdpNmChannelConfig
Description	This container describes the UdpNm TX PDU's.
Configuration Parameters	

SWS Item	**ECUC_UdpNm_00042 :**		
Name	UdpNmTxConfirmationPduId		
Parent Container	UdpNmTxPdu		
Description	Id of the TxPdu that will be used by a TxConfirmation from the lower layer.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 4294967296		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00037 :**		
Name	UdpNmTxPduRef		
Parent Container	UdpNmTxPdu		
Description	The reference to a PDU in the global PDU structure described in the AUTOSAR ECU Configuration Specification. This reference will be used by the UdpNm module to derive the PDU Id.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.6 UdpNmUserDataTxPdu

SWS Item	**ECUC_UdpNm_00056 :**
Container Name	UdpNmUserDataTxPdu
Parent Container	UdpNmChannelConfig
Description	Preprocessor switch for enabling the Tx path of Com User Data. Use case: Setting of NMUserData via SWC.
Configuration Parameters	

SWS Item	**ECUC_UdpNm_00058 :**		
Name	UdpNmTxUserDataPduId		
Parent Container	UdpNmUserDataTxPdu		
Description	This parameter defines the Handle ID of the NM User Data I-PDU.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_UdpNm_00057 :**		
Name	UdpNmTxUserDataPduRef		
Parent Container	UdpNmUserDataTxPdu		
Description	Reference to the NM User Data I-PDU in the global PDU collection.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.7 UdpNmPnInfo

SWS Item	**ECUC_UdpNm_00067 : (Obsolete)**
Container Name	UdpNmPnInfo
Parent Container	UdpNmGlobalConfig
Description	PN information configuration **Tags:** atp.Status=obsolete
Configuration Parameters	

SWS Item	**ECUC_UdpNm_00069 : (Obsolete)**		
Name	UdpNmPnInfoLength		
Parent Container	UdpNmPnInfo		
Description	Specifies the length of the PN request information in the NM message. **Tags:** atp.Status=obsolete		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 63		
Default value	1		
Post-Build Variant Value	false		
Value Configuration Class	*Pre-compile time*	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: only available if UdpNmPnEnabled == true for at least one UdpNm Channel.		

SWS Item	**ECUC_UdpNm_00068 : (Obsolete)**		
Name	UdpNmPnInfoOffset		
Parent Container	UdpNmPnInfo		
Description	Specifies the offset of the PN request information in the NM message. **Tags:** atp.Status=obsolete		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 63		
Default value	1		
Post-Build Variant Value	false		
Value Configuration Class	*Pre-compile time*	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: only available if UdpNmPnEnabled == true for at least one UdpNm Channel.		

Included Containers		
Container Name	*Multiplicity*	*Scope / Dependency*
UdpNmPnFilterMaskByte	1..63	PN information configuration **Tags:** atp.Status=obsolete

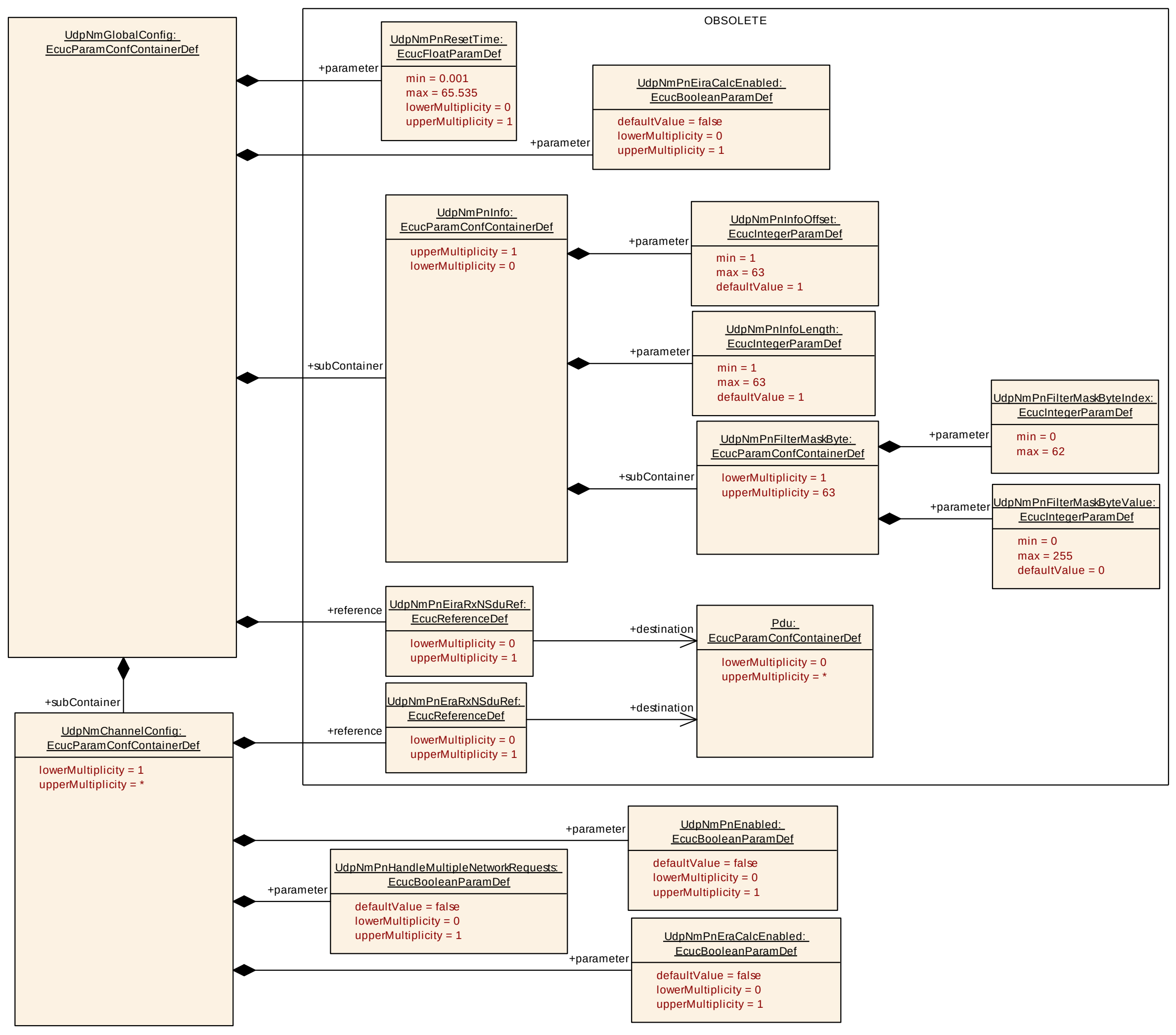

Figure 12: Diagram: UdpNmPNConfig

10.2.8 UdpNmPnFilterMaskByte

SWS Item	**ECUC_UdpNm_00070 : (Obsolete)**
Container Name	UdpNmPnFilterMaskByte
Parent Container	UdpNmPnInfo
Description	PN information configuration **Tags:** atp.Status=obsolete
Configuration Parameters	

SWS Item	**ECUC_UdpNm_00071 : (Obsolete)**		
Name	UdpNmPnFilterMaskByteIndex		
Parent Container	UdpNmPnFilterMaskByte		
Description	Index of the filter mask byte. Specifies the position within the filter mask byte array. **Tags:** atp.Status=obsolete		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 62		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: only available if UdpNmPnEnabled == true for at least one UdpNm Channel. UdpNmPnFilterMaskByteIndex < UdpNmPnInfoLength		

SWS Item	**ECUC_UdpNm_00072 : (Obsolete)**		
Name	UdpNmPnFilterMaskByteValue		
Parent Container	UdpNmPnFilterMaskByte		
Description	Parameter to configure the filter mask byte. **Tags:** atp.Status=obsolete		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	0		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	--	
Scope / Dependency	scope: local dependency: only available if UdpNmPnEnabled == true for at least one UdpNm Channel; UdpNmPnFilterMaskByteIndex < UdpNmPnInfoLength		

No Included Containers

10.3 Published parameters

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

11 Not applicable requirements

[SWS_UdpNm_NA_00999]⌈ This specification item references requirements that are not applicable to this specification. ⌋(SRS_BSW_00170, SRS_BSW_00375, SRS_BSW_00416, SRS_BSW_00168, SRS_BSW_00423, SRS_BSW_00424, SRS_BSW_00425, SRS_BSW_00426, SRS_BSW_00427, SRS_BSW_00429, SRS_BSW_00432, SRS_BSW_00336, SRS_BSW_00417, SRS_BSW_00161, SRS_BSW_00162, SRS_BSW_00005, SRS_BSW_00415, SRS_BSW_00164, SRS_BSW_00325, SRS_BSW_00160, SRS_BSW_00413, SRS_BSW_00347, SRS_BSW_00305, SRS_BSW_00307, SRS_BSW_00335, SRS_BSW_00410, SRS_BSW_00314, SRS_BSW_00328, SRS_BSW_00312, SRS_BSW_00006, SRS_BSW_00377, SRS_BSW_00306, SRS_BSW_00309, SRS_BSW_00330, SRS_BSW_00331, SRS_BSW_00172, SRS_BSW_00010, SRS_BSW_00333, SRS_BSW_00321, SRS_BSW_00341, SRS_BSW_00334, RS_Nm_00151, RS_Nm_00046, RS_Nm_00050, RS_Nm_00052, RS_Nm_02509, RS_Nm_00153, RS_Nm_00054, RS_Nm_00142, RS_Nm_00144, RS_Nm_00154)