Document Title	Specification of Network Management
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	898

Document Status	published
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	R22-11

Document Change History

Date	Release	Changed by	Description
2022-11-24	R22-11	AUTOSAR Release Management	• Added clarifications regarding Operational Modes • Several quality improvements
2021-11-25	R21-11	AUTOSAR Release Management	• Several quality improvements • Removed chapter 10
2020-11-30	R20-11	AUTOSAR Release Management	• Several quality improvments • Changed NetworkState DataType from bool to NetworkStateType
2019-11-28	R19-11	AUTOSAR Release Management	• Added Functional Cluster Lifecycle Chapter • Several quality improvments • Improved linking to Manifest • Changed Document Status from Final to published
2019-03-29	19-03	AUTOSAR Release Management	• Introduced Service Interface for interaction via SM • Introduced possibility to group PNCs/Channels/VLANs
2018-10-30	18-10	AUTOSAR Release Management	• Updated interaction with State Management • Removed APIs and Services (interaction is done via SM) • Temporary removed user data access to applications

2018-03-29	18-03	AUTOSAR Release Management	• Initial relase

Disclaimer

Contents

1 Introduction and functional overview

This specification describes the functionality, API and the configuration of the `Network Management` for the AUTOSAR `Adaptive Platform`.

Adaptive Network Management (NM) is intended to work independent of the communication stack used. Its main purpose is to coordinate the transition between normal operation and bus-sleep mode of the underlying networks (physical and partial networks).

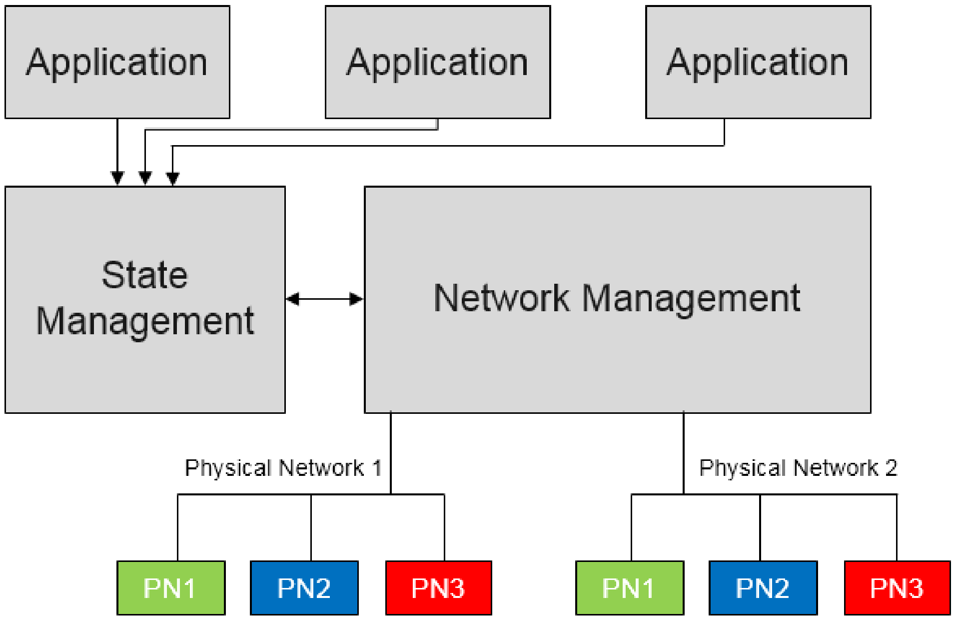

Figure 1.1: Architecture overview with example applications

2 Acronyms and Abbreviations

The glossary below includes acronyms and abbreviations relevant to the Adaptive Network Management that are not included in the AUTOSAR glossary [1].

Abbreviation / Acronym:	Description:
API	Application Programming Interface
CBV	Control Bit Vector
CM	Communication Management
CWU	Car Wakeup
EM	Execution Management
IP	Internet Protocol
MTU	Maximum Transmission Unit
NM	Network Management
NM Node	A node that supports network management. Please note that network node, node and NM node are used with the same meaning througout the document.
PN	Partial Network
PNI	Partial Network Information
PNL	Partial Network Learning
UDP	User Datagram Protocol

Terms:	Description:
Bus communication	Communication on the physical medium
Logical Network	A network in which devices can be addressed independent from the actual network technology.
NM cluster	Set of NM nodes coordinated with the use of the NM algorithm.
NM message	Refers to the payload transmitted in a packet. It contains the NM User Data, Partial Network Information 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 an UDP header in addition to a NM message. Please note that adaptive network management is currently only supported for Ethernet.
PN communication	Communication during partial network operation
Physical channel	A channel enabling communication using physical devices, such as I/O ports and cables.
Repeat Message Request Bit Indication	Repeat Message Bit set in the Control Bit Vector of a received NM message.
Internally Requested	At least one field NetworkRequestedState associated to that channel/network/PNC/VLAN is set to kFullCom.
Exernally Requested	A Network Management Message associated to that channel/network/PNC/VLAN has been received. In case of PNC associated means the bit corresponding to this PNC had the value 1.
FULL_COM	Communication over the network is possible/allowed, the network is up.
NO_COM	Communication over the network is impossible/disabled, the network is down.

3 Related documentation

3.1 Input documents & related standards and norms

[1] Glossary
AUTOSAR_TR_Glossary

[2] General Requirements specific to Adaptive Platform
AUTOSAR_RS_General

[3] Specification of the AUTOSAR Network Management Protocol
AUTOSAR_PRS_NetworkManagementProtocol

[4] Requirements on AUTOSAR Network Management
AUTOSAR_RS_NetworkManagement

[5] Specification of State Management
AUTOSAR_SWS_StateManagement

3.2 Further applicable specification

AUTOSAR provides a General Specification on Basic Software modules [2, RS General], which is also valid for the NM.

4 Constraints and assumptions

4.1 Known Limitations

The Adaptive Network Management is actually only supporting UdpNM.

The Adaptive Network Management does not allow node detection (Repeat Message State) but only handles incoming requests.

The Adaptive Network Management cannot be configured as the master network co-ordinator.

The Adaptive Network Management does not support coordinated shutdown using the information in CBV.

The Adaptive Network Management does not support passive mode and passive start-up. Passive start-up would mean that a node has started (i.e. goes to Normal mode), but the network has been woken up by another node.

Modeling part for mapping the logical networks to the BitVector positions as defined in chapter 7.3 is not available in the manifest.

Update and access of User Data was removed as the service interface to Applications has been removed. State Management will control the network request/release and it must be clarified if user data changes/indications shall be done via State Management or directly by applications.

4.2 Applicability to car domains

AUTOSAR Adaptive Network Management can be used for all car domains.

5 Dependencies to other Functional Clusters

There are no dependencies to other functional clusters.

5.1 Platform dependencies

This specification is part of the AUTOSAR Adaptive Platform and therefore depends on it.

5.2 Protocol layer dependencies

The Adaptive Network Management is based on the protocol mentioned in PRS NetworkManagementProtocol [3].

Adaptive Network Management uses functionality of the underlying communication stack in order to send or receive NM messages on the physical networks.

6 Requirements Tracing

The following table references the requirements specified in RS Adaptive Network Management [4] and links to the fulfillment of these. Please note that if column "Satisfied by" is empty for a specific requirement, or a requirement is not reported in the table, it means that this requirement is not fulfilled by this document.

Requirement	Description	Satisfied by
[RS_Nm_00043]	Nm shall not prohibit bus traffic with Nm not being initialized	[SWS_ANM_00090]
[RS_Nm_00044]	The Nm shall be applicable to different types of communication systems which are in the scope of Autosar and support a bus sleep mode.	[SWS_ANM_00005] [SWS_ANM_00006] [SWS_ANM_00007] [SWS_ANM_00008] [SWS_ANM_00009] [SWS_ANM_00012] [SWS_ANM_00013] [SWS_ANM_00016] [SWS_ANM_00017] [SWS_ANM_00021] [SWS_ANM_00044] [SWS_ANM_00046] [SWS_ANM_00047] [SWS_ANM_00062] [SWS_ANM_00070] [SWS_ANM_00092]
[RS_Nm_00047]	Nm shall provide a service to request to keep the bus awake and a service to cancel this request.	[SWS_ANM_00011] [SWS_ANM_00014] [SWS_ANM_00015] [SWS_ANM_00016] [SWS_ANM_00018] [SWS_ANM_00019] [SWS_ANM_00020] [SWS_ANM_00022] [SWS_ANM_00023] [SWS_ANM_00025] [SWS_ANM_00066] [SWS_ANM_00086] [SWS_ANM_00087] [SWS_ANM_00088]
[RS_Nm_00048]	Nm shall put the communication controller into sleep mode if there is no bus communication	[SWS_ANM_00024]
[RS_Nm_00050]	The Nm shall provide the current state of Nm	[SWS_ANM_00063] [SWS_ANM_00083]
[RS_Nm_00051]	Nm shall inform application when Nm state changes occur.	[SWS_ANM_00093] [SWS_ANM_91000] [SWS_ANM_91001]
[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_ANM_00024]

Requirement	Description	Satisfied by
[RS_Nm_00150]	Specific features of the Network Management shall be configurable	[SWS_ANM_00007] [SWS_ANM_00013] [SWS_ANM_00029] [SWS_ANM_00033] [SWS_ANM_00035] [SWS_ANM_00040] [SWS_ANM_00044] [SWS_ANM_00047] [SWS_ANM_00051] [SWS_ANM_00081] [SWS_ANM_00084] [SWS_ANM_00085] [SWS_ANM_00089]
[RS_Nm_00151]	The Network Management algorithm shall allow any node to integrate into an already running Nm cluster	[SWS_ANM_00037] [SWS_ANM_00038] [SWS_ANM_00071] [SWS_ANM_00091]
[RS_Nm_02505]	The Nm shall optionally set the local node identifier to the Nm-message	[SWS_ANM_00033] [SWS_ANM_00034]
[RS_Nm_02508]	Every node shall have a node identifier associated with it that is unique in the Nm-cluster.	[SWS_ANM_00034]
[RS_Nm_02517]	CanNm shall support Partial Networking on CAN	[SWS_ANM_00067] [SWS_ANM_00092]
[RS_Nm_02520]	No description	[SWS_ANM_00051] [SWS_ANM_00055]
[RS_Nm_02527]	Nm shall implement a filter algorithm dropping all Nm messages that are not relevant for the ECU	[SWS_ANM_00081]
[RS_Nm_02528]	Nm shall provide a service which allows for instantaneous sending of Nm messages.	[SWS_ANM_00094]

7 Functional specification

The Adaptive Network Management offers services that allows to request and query the network states for logical network handles that can be mapped to physical or partial networks.

To do so, the following functionalities are provided:

1) Field for requesting and releasing logical network handles
2) Support for partial networking

7.1 Architectural Overview

Figure 7.1 gives an overview of the Adaptive NM service.

The following figure shows an overview on the interaction between [5] and `Network Management` as well as an example mapping between logical networks, partial networks and physical networks.

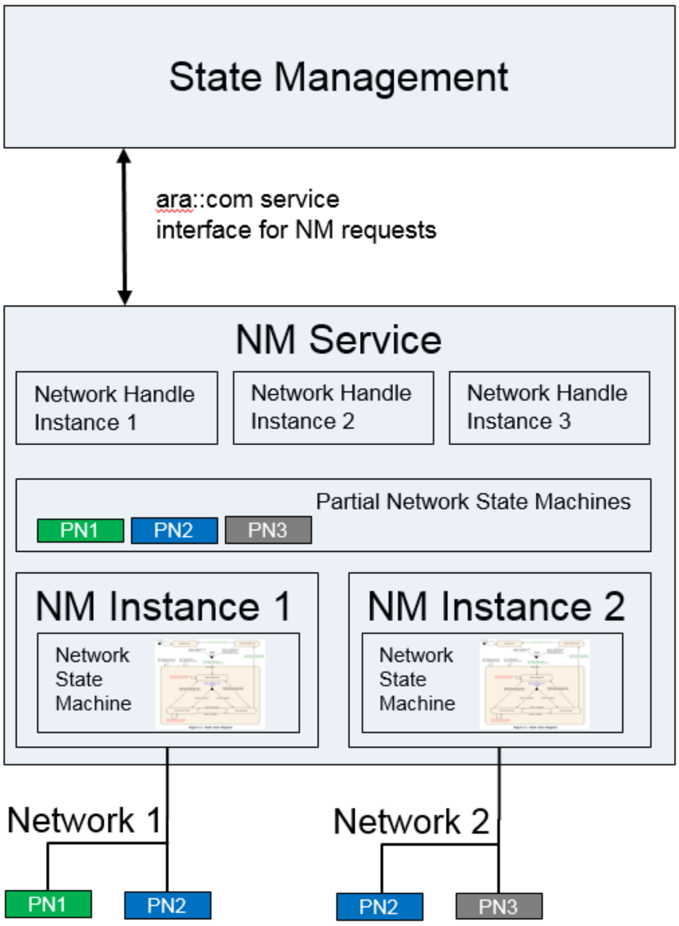

Figure 7.1: Overview Of Network Management

7.2 Network Management Algorithm

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

The AUTOSAR Adaptive NM algorithm is based on periodic NM packets, which are received by all nodes in the cluster via multicast. Reception of NM packets indicates that sending nodes want to keep the NM-cluster awake. If any node is ready to go to sleep mode, it stops sending NM packets, but as long as NM packets from other nodes are received, it postpones transition to sleep mode. Finally, if a dedicated timer elapses because no NM packets are received anymore, every node initiates transition to the sleep mode.

If any node in the NM-cluster requires bus-communication, it can keep the NM-cluster awake by transmitting NM packets.

The main concept of the AUTOSAR Adaptive NM coordination algorithm as described in [3] can be summarized by the following key-behavior:

Every network node transmits periodic NM messages as long as it requires bus-communication; otherwise it does not transmit NM messages.

7.3 NetworkControl

Logical network handles are a mapped to one or more partial or physical networks, while a partial network itself can be mapped to one or multiple physical networks. By using the logical network hanlde all assigned partial networks, VLANs and underlying physical channel(s) are controlled together.

With the introduction of the `State Management` functional cluster, `Network Management` no longer receives logical network requests from applications, instead they are controlled by the `State Management`. `State Management` can split the one or more applications in multiple functions that might require network communication. Applications (or part of) would then request different functions to be activated/deactivated from `State Management` and then `State Management` would in turn, depending on configuration, request/release different logical networks. NM checks then the requested state over all logical networks handles and will activate or deactivate the according physical networks.

[SWS_ANM_00063] ⌈Each port offered by `NM` shall enable control of one logical `NmNetworkHandle` which in turn can be mapped to partial networks or VLANs.⌋*(RS_Nm_00050)*

Note: In the Manifest the untagged VLAN represents the physical ethernet channel.

[SWS_ANM_00066] ⌈Each logical `NmNetworkHandle` shall be mapped to partial networks(via `PncMappingIdent`) and/or VLANs (via `EthernetCommunicationConnector`). Configurations where a logical `NmNetworkHandle` maps the same VLAN directly and via partial network(s) shall not be possible.⌋*(RS_Nm_00047)*

[SWS_ANM_00067] ⌈If partial networking is used a mapping between partial network(s) and `EthernetCommunicationConnector` shall be configured in `PncMapping`.⌋*(RS_Nm_02517)* Note: One Partial Network can be mapped to several VLAN(s)

[SWS_ANM_00083] ⌈The value of the field `NetworkCurrentState` shall be kFullCom if all PNCs, VLANs and/or physical channels associated to this instance of the `NetworkState` service are in `kFullCom`. Otherwise the value shall be `kNoCom`.⌋*(RS_Nm_00050)*

Note: The consequence of [SWS_ANM_00083] is, that a lowest wins strategy is applied.

[SWS_ANM_00084] ⌈Network Management shall consider each PNC, VLAN and/or physical channel as internally requested that is associated to an instance of the service `NetworkState` and the value of the field `NetworkRequestedState` has the value kFullCom.⌋*(RS_Nm_00150)*

Note:The consequence of [SWS_ANM_00084] is, that a highest wins strategy is applied, that means that if in any instance of the `NetworkState` Service the field `NetworkRequestedState` is set to kFullCom the target state of the associated PNC/VLAN/channel(s) is `Network Mode`.

[SWS_ANM_00085] ⌈Network Management shall bring (or keep) all networks/physical channels to kFullCom that are either internally requested or externally requested.⌋ *(RS_Nm_00150)*

[SWS_ANM_00086] ⌈A PNC shall be considered in kFullCom, if all physical channels, to which it is mapped, are in kFullCom and the PNC is internally or externally requested. This includes the change notification to the corresponding NetworkState.-NetworkCurrentState(s).⌋ *(RS_Nm_00047)*

[SWS_ANM_00087] Handling of external wake-up ⌈Upon detection of of an external wake-up, kFullCom shall be the target state for the corresponding channel(s). If Network Management is configured for that channel, the target state shall be Network Mode, with the default initial sub state Repeat Message.⌋ *(RS_Nm_00047)*

Note: Its up to the Platfrom Implementation how an external wakeup event is detected.

[SWS_ANM_00088] Default target state after start-up ⌈The default target state after start up for channels for which no external wake-up has been detected shall be kNoCom.⌋ *(RS_Nm_00047)*

7.4 Operational Modes

This chapter describes the operational modes of the AUTOSAR Adaptive NM.

[SWS_ANM_00062]{DRAFT} ⌈NM shall realize the state machine mentioned in the figure described below for every physical channel separately.⌋*(RS_Nm_00044)*

Note: The state machine in Figure 7.2 is applied to physical channels. In case of partial networking, the Nm module should additionally take care of relevant PNs.

The Network Management contains three operational modes:

- Network Mode, see 7.4.1
- Prepare Bus-Sleep Mode, see 7.4.2
- Bus-Sleep Mode, see 7.4.3

These modes will not be visible to the Adaptive Application as it is.

When the NM is in Network mode it implies that the network is requested or active. And the logical network information bit will be set to 1.

When the NM is in Prepare Bus-Sleep or Bus-Sleep Mode, It implies that the network is released or inactive. And the logical network information bit will be set to 0.

The following figure shows the state diagram. Mode change related transitions are denoted in green and error handling related transmissions in red.

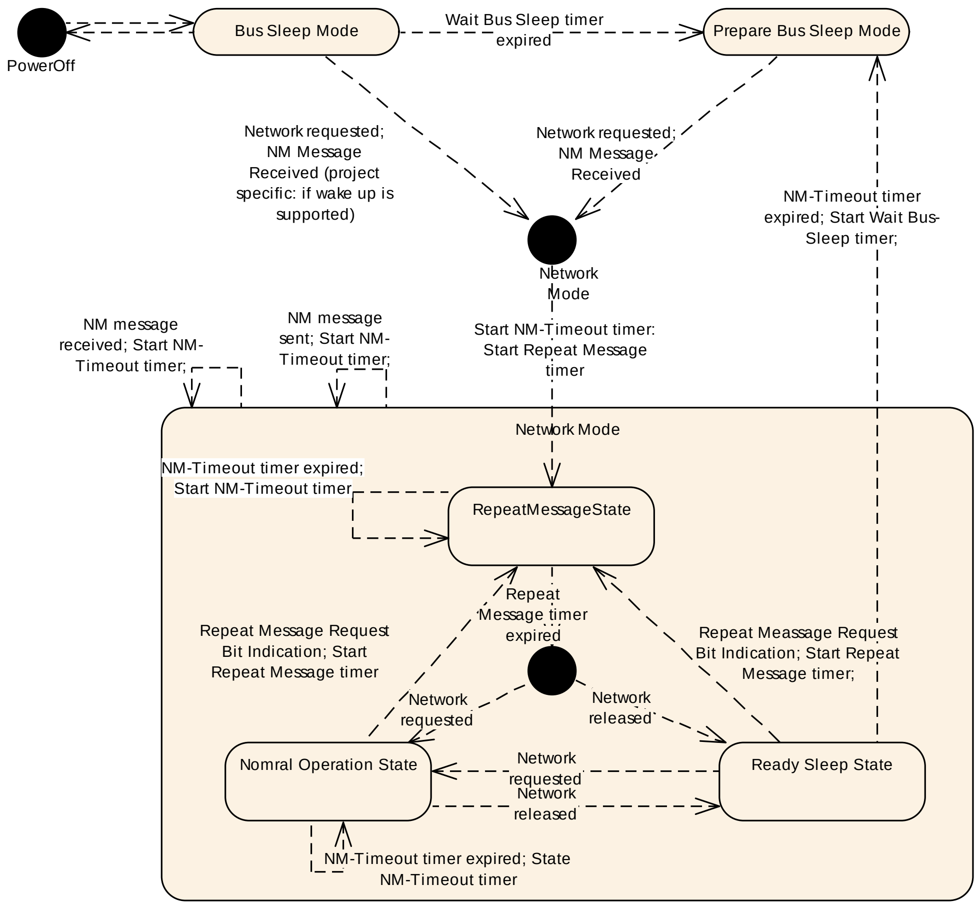

Figure 7.2: State Chart Diagram

7.4.1 Network Mode

[SWS_ANM_00005] ⌈The Network Mode shall consist of three internal states:

- Repeat Message State
- Normal Operation State
- Ready Sleep State

⌋*(RS_Nm_00044)*

For more information check the following chapters:

- Repeat Message State, see 7.4.1.1
- Normal Operation State, see 7.4.1.2
- Ready Sleep State, see 7.4.1.3

[SWS_ANM_00006] ⌈When the Network Mode is entered from Bus-Sleep Mode or Prepare Bus-Sleep Mode, by default, the Repeat Message State shall be entered.⌋ *(RS_Nm_00044)*

[SWS_ANM_00007] ⌈When the Network Mode is entered, the NM-Timeout Timer shall be started with the value nmNetworkTimeout.⌋*(RS_Nm_00044, RS_Nm_00150)*

[SWS_ANM_00008] ⌈Upon successful reception of a NM message in Network Mode, the NM-Timeout Timer shall be restarted with the value nmNetworkTimeout.⌋*(RS_Nm_00044)*

[SWS_ANM_00009] ⌈Upon successful transmission of a NM message in Network Mode, the NM-Timeout Timer shall be restarted with the value nmNetworkTimeout.⌋*(RS_Nm_00044)*

7.4.1.1 Repeat Message State

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.

[SWS_ANM_00011] ⌈When the Repeat Message State of Network Mode is entered from Bus-Sleep Mode, Prepare-Bus-Sleep Mode or from within Network Mode (Normal Operation State or Ready Sleep State) transmission of NM messages shall be (re-) started.⌋*(RS_Nm_00047)*

[SWS_ANM_00012] ⌈When the NM-Timeout Timer expires in the Repeat Message State, the NM-Timeout Timer shall be restarted.⌋*(RS_Nm_00044)*

[SWS_ANM_00013] ⌈The NM shall stay in the Repeat Message State for a configurable amount of time determined by the nmRepeatMessageTime; after that time the Repeat Message State shall be left.⌋*(RS_Nm_00044, RS_Nm_00150)*

[SWS_ANM_00014] ⌈When Repeat Message State is left, the Normal Operation State shall be entered, if the network has been requested.⌋*(RS_Nm_00047)*

[SWS_ANM_00015] ⌈When Repeat Message State is left, the Ready Sleep State shall be entered, if the network has been released.⌋*(RS_Nm_00047)*

[SWS_ANM_00070] ⌈The Repeat Message State of Network Mode is entered from Bus-Sleep Mode or Prepare-Bus-Sleep Mode by default, when a network (channel/PNC/VLAN) is requested by setting the value of NetworkRequestedState to kFullCom and the NM module shall transmit a NM message immediately.⌋*(RS_Nm_00044)*

[SWS_ANM_00092]{DRAFT} **nmPnHandleMultipleNetworkRequests** ⌈If in Ready Sleep State, Normal Operation State or Repeat Message State and nmPnHandleMultipleNetworkRequests is set to true and the requested state of the channel/VLAN or an associated PNC changes, Repeat Message State shall be (re-)entered.⌋*(RS_Nm_00044, RS_Nm_02517)*

7.4.1.2 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_ANM_00016] ⌈When the Normal Operation State is entered from Ready Sleep State, transmission of NM messages shall be started immediately.⌋*(RS_Nm_00047, RS_Nm_00044)*

[SWS_ANM_00017] ⌈When the NM-Timeout Timer expires in the Normal Operation State, the NM-Timeout Timer shall be restarted.⌋*(RS_Nm_00044)*

[SWS_ANM_00018] ⌈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.⌋*(RS_Nm_00047)*

[SWS_ANM_00019] ⌈If Repeat Message Request Bit (set in the CBV of the received NM message) is received in the Normal Operation State, the Normal Operation State shall be left and the Repeat Message State shall be entered.⌋*(RS_Nm_00047)*

7.4.1.3 Ready Sleep State

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

[SWS_ANM_00020] ⌈When the Ready Sleep State is entered from Repeat Message State or Normal Operation State, transmission of NM messages shall be stopped.⌋ *(RS_Nm_00047)*

[SWS_ANM_00021] ⌈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.⌋ *(RS_Nm_00044)*

[SWS_ANM_00022] ⌈When the network is requested (by setting the value of `NetworkRequestedState` to kFullCom) and the current state is the Ready Sleep State, the Ready Sleep State shall be left and the Normal Operation State shall be entered.⌋ *(RS_Nm_00047)*

[SWS_ANM_00023] ⌈If Repeat Message Request Bit (set in the CBV of the received NM message) is received in the Ready Sleep State, the Ready Sleep State shall be left and the Repeat Message State shall be entered.⌋*(RS_Nm_00047)*

Note: Handling of multiple transition conditions which might arise at the same time (e.g. NM-Timeout timer expires vs. network is requested) is considered to be implementation-specific.

7.4.2 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_ANM_00024] ⌈The NM shall stay in the Prepare Bus-Sleep Mode for an amount of time determined by the `nmWaitBusSleepTime`; after that time, the Prepare Bus-Sleep Mode shall be left and the Bus-Sleep Mode shall be entered.⌋*(RS_Nm_00048, RS_Nm_00054)*

[SWS_ANM_00025] ⌈Upon successful reception of a NM message 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.⌋*(RS_Nm_00047)*

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.

7.4.3 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. Transmission and reception capabilities can be switched off if supported by hardware.

If a configurable amount of time determined by `nmNetworkTimeout` + `nmWaitBusSleepTime` 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.

[SWS_ANM_00029]{DRAFT} ⌈In Bus-Sleep Mode the value of the corresponding `NetworkState.NetworkCurrentState` field(s) are kNoCom (see also [SWS_ANM_00083]).⌋*(RS_Nm_00150)*

Note: Reception of a message during bus sleep (if receiving capability is not switched off) is not explicitly handled in this specification as for example wake-up is considered project specific.

Note: In Bus-Sleep Mode, it is assumed that all nodes in a cluster are in this state. Typically, all nodes request the communication approximately at the same time by a common trigger, for instance a wake-up line.

7.5 Message Format

Message Layout is shown in [3], chapter 5.1.

Note: As mentioned in [3], the length of an NM packet shall not exceed the MTU (Maximum Transmission Unit) of the underlying physical transport layer.

7.5.1 Source Node Identifier

[SWS_ANM_00033] ⌈The location of the source node identifier shall be taken from `nmNidPosition`. If `nmNidPosition` is not set, NID shall not be contained in the Nm messages (see [PRS_Nm_00074]).⌋*(RS_Nm_00150, RS_Nm_02505)*

[SWS_ANM_00034] ⌈The source node identifier shall be set with configurable NodeId value `nmNodeId` unless the location of the source node identifier is set to Off (see [PRS_Nm_00013]).⌋*(RS_Nm_02508, RS_Nm_02505)*

7.5.2 Control Bit Vector

The format (bit-layout) and definition of the CBV is shown in [3], chapter 5.1.2 Control Bit Vector.

[SWS_ANM_00035] ⌈The location of the Control Bit Vector shall be configurable using `nmCbvPosition`. If `nmCbvPosition` is not set, CBV shall not be contained in the Nm messages (see [PRS_Nm_00075]).⌋*(RS_Nm_00150)*

[SWS_ANM_00037] ⌈Repeat Message Request Bit shall always be set to 0 in the transmitted NM message.⌋*(RS_Nm_00151)*

[SWS_ANM_00038] ⌈Active Wakeup Bit shall always be set to 0 in the transmitted NM message.⌋*(RS_Nm_00151)*

[SWS_ANM_00071] ⌈NM Coordinator Sleep Ready Bit shall always be set to 0 in the transmitted NM message.⌋*(RS_Nm_00151)*

[SWS_ANM_00091] ⌈Partial Network Learning Bit (PNL) shall always be set to 0 in the transmitted NM message.⌋*(RS_Nm_00151)*

7.5.3 User Data

[SWS_ANM_00040] ⌈If NM user data is configured (i.e. `nmUserDataLength` is existing with a value greater than 0) it shall be always included in the NM message (see [PRS_Nm_00158]).⌋*(RS_Nm_00150)*

Note: the range (in bytes) that contains the user data in the received NM message is defined by `nmUserDataLength`.

Note: UserData does not include the PNI in case the Partial Networking is active. Received and Transmitted UserData does not overlap with the PNI.

7.6 Nm Transmission

7.6.1 Transmission Scheduling

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

[SWS_ANM_00044] ⌈If the Repeat Message State is entered ([SWS_ANM_00070]), the transmission of NM message shall be delayed by `nmMsgCycleOffset` after entering the Repeat Message State.⌋*(RS_Nm_00044, RS_Nm_00150)*

[SWS_ANM_00046] ⌈If transmission of NM messages has been started and the NM Message Cycle Timer expires, a NM message transmission shall be initiated.⌋*(RS_Nm_00044)*

[SWS_ANM_00047] ⌈If the NM Message Cycle Timer expires it shall be restarted with `nmMsgCycleTime`.⌋*(RS_Nm_00044, RS_Nm_00150)*

[SWS_ANM_00094] Immediate Nm Transmissions ⌈Upon an active network request (corresponding NetworkRequestedState has been set to true) immediate Nm messages shall be sent according to PRS_Nm_00334 using `nmImmediateNmCycleTime` as NmImmediateNmCycleTime and `nmImmediateNmTransmissions` as NmImmediateNmTransmissions.⌋*(RS_Nm_02528)*

7.7 Nm User Data Handling

Note: Although contained in the underlying Protocol Specification [3] currently no use case is seen for user data.

7.8 Partial Networking

7.8.1 Partial Network State Machine

The partial network state machine mentioned in Figure 7.1 is supposed to be implementation specific. Note: Although being implementation specific, the implemented behaviour shall conform to the Partial Networking requirements described in [3].

7.8.2 Rx Handling of NM messages

Note : Reception Handling of PNC bit vector as described in [3] is switched on/off by NmCluster.nmPncParticipation

[SWS_ANM_00051] ⌈If nmPncParticipation is TRUE and the PNI bit in the received NM message is 1, the NM shall update the value of the associated NetworkCurrentState, every time a BitVector has been received in the NM message that leads to a change in the actual request state of a relevant Partial Network or a channel to kFullCom.⌋*(RS_Nm_00150, RS_Nm_02520)*

Note: If nmPncParticipation is TRUE and the PNI bit in the received NM message is 0, NM module still will process the user data information.

Note: PnResetTime specified in [3] shall be configured by pnResetTimer

7.8.3 Tx Handling of NM messages

Note: nmPncParticipation enable/disable the PNI transmission as specified in chapter "Handling of Tx NM messages" of [3].

Note: The usage of the CBV is mandatory in case Partial Networking is used. This has to be ensured by configuration in the respective platform.

7.8.4 NM message Filter Algorithm

[SWS_ANM_00055] ⌈The range (in bytes) that contains the Partial Network request information (PNC bit vector) in the received NM message shall be defined by PNC bit vector offset (pncVectorOffset) starting from byte 0 and PNC bit vector length (pncVectorLength).⌋*(RS_Nm_02520)*

Example:

- PNC bit vector Offset = 3
- PNC bit vector Length = 2

In the above example only Byte 3 and Byte 4 of the NM message contain Partial Network request information.

Note: Every bit of the PNC bit vector 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 Partial Network.

[SWS_ANM_00081] ⌈The Nm shall filter out messages containing Partial Network request information if they do not contain at least one bit set to 1 that corresponds to a Partial Network that is configured in a NmNetworkHandle and allNmMessagesKeepAwake is FALSE.⌋*(RS_Nm_00150, RS_Nm_02527)*

Note: When activated the Nm Message Filter Algorithm will filter out any Nm Message not containing at least one relevant Partial Network being requested (its Bit in the PN bit vector set to 1).

[SWS_ANM_00089] allNmMessagesKeepAwake ⌈If no relevant Partial Network is requested in the received NM Message and `allNmMessagesKeepAwake` is TRUE the Message shall not be filtered out from further Rx Indication handling.⌋*(RS_Nm_00150)*

Note: This is required to enable the ECU to stay awake on any kind of NM Message.

7.9 Functional Cluster Lifecycle

7.9.1 Startup

No special startup handling needed for Network Management. The environment is expected to take care that Network Management is running and able to serve communication requests as soon as network communication is needed. If and how the Nm is actually start up in advance depends on platform constraints like e.g. fast (re-)start etc.

7.9.2 Shutdown

[SWS_ANM_00090] Communication Shutdown ⌈When a SIGTERM is received by NM, any active Network Requests shall be withdrawn and NetworkCurrentState shall be set to kNoCom. Afterwards all provided service offers shall be stopped.⌋*(RS_-Nm_00043)*

Note: The NetworkHardware might be shutdown afterwards. It is assumed that State-Management takes care that no shutdown is initiated while Network Communication is still needed and that active Network Requests during shutdown are an exceptional situation.

8 API specification

The `Network Management` does not contain any APIs. All the functionality is provided via Services.

9 Service Interfaces

The services provided by `Network Management` are not intended to be used by applications. Requests for networks shall be received from `State Management`.

[SWS_ANM_00093] ⌈

Name	NetworkStateType	
Namespace	ara::nm	
Kind	TYPE_REFERENCE	
Derived from	`uint8_t`	
Description	NetworkStateType	
Range / Symbol	***Limit***	***Description***
kNoCom		kNoCom
kFullCom		kFullCom

⌋*(RS_Nm_00051)*

[SWS_ANM_91001] ⌈

Name	NetworkState_{NetworkHandle}		
Kind	ProvidedPort	***Interface***	`NetworkState`
Description	Provides information about network status per NetworkHandle. Intended to be only used by State Management!		
Variation	`FOR NetworkHandle : MODEL.filterType("NmNetworkHandle");`		

⌋*(RS_Nm_00051)*

[SWS_ANM_91000]{DRAFT} ⌈

Name	NetworkState
NameSpace	ara::nm

Field	NetworkCurrentState
Description	PNC / VLAN / Physical Network is currently active or not
Type	`NetworkStateType`
HasGetter	true
HasNotifier	true
HasSetter	false

Field	NetworkRequestedState
Description	Request or Release PNC / VLAN / Physical Network to get active or to release
Type	`NetworkStateType`
HasGetter	true
HasNotifier	true
HasSetter	true

⌋*(RS_Nm_00051)*

A Mentioned Manifest Elements

For the sake of completeness, this chapter contains a set of class tables representing meta-classes mentioned in the context of this document but which are not contained directly in the scope of describing specific meta-model semantics.

Chapter is generated.

Class	**EthernetCommunicationConnector**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Ethernet specific attributes to the CommunicationConnector.			
Base	*ARObject*, *CommunicationConnector*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
apApplication Endpoint	ApApplicationEndpoint	*	aggr	Collection of Application Addresses that are used on the CommunicationConnector. **Tags:**atp.Status=draft
canXlProps	CanXlProps	*	ref	If the Ethernet frames handled by this Ethernet CommunicationConnector are tunneled through CAN XL, then this reference shall refer the CanXlProps which contains the specific configuration parameters of the CAN XL controller of the physical CAN XL connection to be used for tunneling. **Tags:**atp.Status=draft
maximum Transmission Unit	PositiveInteger	0..1	attr	This attribute specifies the maximum transmission unit in bytes.
neighborCache Size	PositiveInteger	0..1	attr	This attribute specifies the size of neighbor cache or ARP table in units of entries.
pathMtu Enabled	Boolean	0..1	attr	If enabled the IPv4/IPv6 processes incoming ICMP "Packet Too Big" messages and stores a MTU value for each destination address.
pathMtuTimeout	TimeValue	0..1	attr	If this value is >0 the IPv4/IPv6 will reset the MTU value stored for each destination after n seconds.
unicastNetwork Endpoint	NetworkEndpoint	*	ref	Network Endpoint that defines the IPAddress of the machine. **Tags:**atp.Status=draft

Table A.1: EthernetCommunicationConnector

Class	**MachineDesign**			
Package	M2::AUTOSARTemplates::AdaptivePlatform::SystemDesign			
Note	This meta-class represents the ability to define requirements on a Machine in the context of designing a system. **Tags:**atp.recommendedPackage=MachineDesigns			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element, *AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
accessControl	AccessControlEnum	0..1	attr	This attribute defines how the access restriction to the Service Instance is defined.

▽

△

Class	MachineDesign			
communication Connector	Communication Connector	*	aggr	This aggregation defines the network connection of the machine. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=communicationConnector.shortName
communication Controller	Communication Controller	*	aggr	CommunicationControllers of the Machine that are used for description of 10-Base-T1S topologies **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=communicationController.shortName
ethIpProps	EthIpProps	*	ref	Machine specific IP attributes.
pncPrepare SleepTimer	TimeValue	0..1	attr	Time in seconds the PNC state machine shall wait in PNC_PREPARE_SLEEP.
pnResetTimer	TimeValue	0..1	attr	Specifies the runtime of the reset timer in seconds. This reset time is valid for the reset of PN requests.
service Discovery Config	ServiceDiscovery Configuration	*	aggr	Set of service discovery configuration settings that are defined on the machine for individual Communication Connectors. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=serviceDiscoveryConfig
tcpIpIcmpProps	EthTcpIpIcmpProps	*	ref	Machine specific ICMP (Internet Control Message Protocol) attributes
tcpIpProps	EthTcpIpProps	*	ref	Machine specific TcpIp Stack attributes.

Table A.2: MachineDesign

Class	***NmCluster*** (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Set of NM nodes coordinated with use of the NM algorithm.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	CanNmCluster, FlexrayNmCluster, UdpNmCluster			
Aggregated by	NmConfig.nmCluster			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Cluster	CommunicationCluster	0..1	ref	Association to a CommunicationCluster in the topology description.
nmNode	NmNode	*	aggr	Collection of NmNodes of the NmCluster. atpVariation: Derived, because NmNode can be variable. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=nmNode.shortName, nmNode.variation Point.shortLabel vh.latestBindingTime=postBuild
nmPnc Participation	Boolean	0..1	attr	Defines whether this NmCluster contributes to the partial network mechanism.
pncCluster VectorLength	PositiveInteger	0..1	attr	Optionally defines the length of the PNC Vector per CommunicationCluster (and VLAN in case of UdpNm). If not defined then System.pncVectorLength applies. Should only make the PNC Vector shorter (or same length as defined in System.pncVectorLength). **Tags:**atp.Status=draft

Table A.3: NmCluster

Class	**NmNetworkHandle**			
Package	M2::AUTOSARTemplates::AdaptivePlatform::PlatformModuleDeployment::AdaptiveModuleImplementation			
Note	Group of partialNetworks and/or VLANs that can be controlled collectively.			
Base	*ARObject*, *Referrable*			
Aggregated by	NmInstantiation.networkHandle			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
partialNetwork	PncMappingIdent	*	ref	Reference to a Partial Network that is included in the NmNetworkHandle.
vlan	EthernetCommunicationConnector	*	ref	Reference to a VLAN that is included in the NmNetworkHandle.

Table A.4: NmNetworkHandle

Class	***NmNode*** (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	The linking of NmEcus to NmClusters is realized via the NmNodes.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	CanNmNode, FlexrayNmNode, UdpNmNode			
Aggregated by	*NmCluster*.nmNode			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
machine	MachineDesign	0..1	ref	Reference to the machine that contains the NmNode. **Tags:**atp.Status=draft
nmNodeId	Integer	0..1	attr	Node identifier of local NmNode. Shall be unique in the NmCluster.

Table A.5: NmNode

Class	**PncMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::PncMapping			
Note	Describes a mapping between one or several Virtual Function Clusters onto Partial Network Clusters. A Virtual Function Cluster is realized by a PortGroup. A Partial Network Cluster is realized by one or more ServiceInstances.			
Base	*ARObject*, *Describable*			
Aggregated by	SystemMapping.pncMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ident	PncMappingIdent	0..1	aggr	This adds the ability to become referrable to PncMapping.
physicalChannel	PhysicalChannel	*	ref	This reference maps the partial network to a communication channel.
pncConsumedProvidedServiceInstanceGroup	ConsumedProvidedServiceInstanceGroup	*	ref	ConsumedProvidedServiceInstanceGroup used in a Partial Network Cluster. This reference is optional, since this could be used for starting and stopping ConsumedProvidedServiceInstanceGroup according the requested partial network, but is not necessarily needed. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=pncConsumedProvidedServiceInstanceGroup.consumedProvidedServiceInstanceGroup, pncConsumedProvidedServiceInstanceGroup.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	PncMapping			
pncIdentifier	PositiveInteger	1	attr	Identifer of the Partial Network Cluster. This number represents the absolute bit position of this Partial Network Cluster in the NM Pdu.
pncWakeup Enable	Boolean	0..1	attr	If this parameter is available and set to true then this PNC will be woken up as soon as a channel wakeup occurs on a channel where this PNC is assigned to. This is ensured by adding this PNC to the corresponding channel wakeup sources during upstream mapping.
serviceInstance	AdaptivePlatform ServiceInstance	*	ref	Reference to ServiceInstances that are participating in a Partial Network Cluster. **Tags:**atp.Status=draft
shortLabel	Identifier	0..1	attr	This attribute specifies an identifying shortName for the PncMapping. It shall be unique in the System scope.
vfc	PortGroup	*	iref	Virtual Function Cluster to be mapped onto a Partial Network Cluster. This reference is optional in case that the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy systems. **InstanceRef implemented by:**PortGroupInSystem InstanceRef

Table A.6: PncMapping

Class	**PncMappingIdent**			
Package	M2::AUTOSARTemplates::SystemTemplate::PncMapping			
Note	This meta-class is created to add the ability to become the target of a reference to the non-Referrable PncMapping.			
Base	*ARObject*, *Referrable*			
Aggregated by	PncMapping.ident			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.7: PncMappingIdent

Class	**System**			
Package	M2::AUTOSARTemplates::SystemTemplate			
Note	The top level element of the System Description. **Tags:**atp.recommendedPackage=Systems			
Base	*ARElement*, *ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element, *AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**System**			
fibexElement	FibexElement	*	ref	Reference to ASAM FIBEX elements specifying Communication and Topology. All Fibex Elements used within a System Description shall be referenced from the System Element. atpVariation: In order to describe a product-line, all Fibex Elements can be optional. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=fibexElement.fibexElement, fibexElement.variationPoint.shortLabel vh.latestBindingTime=postBuild
interpolationRoutineMappingSet	InterpolationRoutineMappingSet	*	ref	This reference identifies the InterpolationRoutineMappingSets that are relevant in the context of the enclosing System.
mapping	SystemMapping	*	aggr	Aggregation of all mapping aspects relevant in the System Description. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=mapping.shortName, mapping.variationPoint.shortLabel vh.latestBindingTime=postBuild
pncVectorLength	PositiveInteger	0..1	attr	Length of the partial networking request release information vector (in bytes).
pncVectorOffset	PositiveInteger	0..1	attr	Absolute offset (with respect to the NM-PDU) of the partial networking request release information vector that is defined in bytes as an index starting with 0.
rootSoftwareComposition	RootSwCompositionPrototype	0..1	aggr	Aggregation of the root software composition, containing all software components in the System in a hierarchical structure. This element is not required when the System description is used for a network-only use-case. atpVariation: The RootSwCompositionPrototype can vary. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=rootSoftwareComposition.shortName, rootSoftwareComposition.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
systemVersion	RevisionLabelString	1	attr	Version number of the System Description.

Table A.8: System

Class	**UdpNmCluster**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Udp specific NmCluster attributes			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmCluster*, *Referrable*			
Aggregated by	NmConfig.nmCluster			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
networkConfiguration	UdpNmNetworkConfiguration	0..1	aggr	Configuration of a UDP port and UDP multicast IP address of the Nm communication on a VLAN. **Tags:**atp.Status=draft
nmCbvPosition	Integer	0..1	attr	Defines the position of the control bit vector within the Nm Pdu (Byte position). If this attribute is not configured, the Control Bit Vector is not used.

▽

△

Class	UdpNmCluster			
nmImmediate NmCycleTime	TimeValue	0..1	attr	Defines the immediate NmPdu cycle time in seconds which is used for nmImmediateNmTransmissions NmPdu transmissions. This attribute is only valid if nmImmediate NmTransmissions is greater one.
nmImmediate Nm Transmissions	PositiveInteger	0..1	attr	Defines the number of immediate NmPdus which shall be transmitted. If the value is zero no immediate NmPdus are transmitted. The cycle time of immediate NmPdus is defined by nmImmediateNmCycleTime.
nmMsgCycle Time	TimeValue	0..1	attr	Period of a NmPdu in seconds. It determines the periodic rate in the periodic transmission mode with bus load reduction and is the basis for transmit scheduling in the periodic transmission mode without bus load reduction.
nmNetwork Timeout	TimeValue	0..1	attr	Network Timeout for NmPdus in seconds. It denotes the time how long the UdpNm shall stay in the Network Mode before transition into Prepare Bus-Sleep Mode shall take place.
nmNidPosition	Integer	0..1	attr	Defines the byte position of the source node identifier within the NmPdu. If this attribute is not configured, the Node Identification is not used.
nmRepeat MessageTime	TimeValue	0..1	attr	Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.
nmUserData Length	Integer	0..1	attr	Defines the length in bytes of the user data contained in the Nm message. User data excludes the PNC bit vector.
nmUserData Offset	PositiveInteger	0..1	attr	Specifies the offset (in bytes) of the user data information in the NM message. User data excludes the PNC bit vector. **Tags:**atp.Status=draft
nmWaitBus SleepTime	TimeValue	0..1	attr	Timeout for bus calm down phase in seconds. It denotes the time how long the CanNm shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place.
vlan	EthernetPhysical Channel	0..1	ref	Reference to the vlan (represented by the Ethernet PhysicalChannel) this UdpNmCluster shall apply to.

Table A.9: UdpNmCluster

Class	**UdpNmNode**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Udp specific NM Node attributes.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmNode*, *Referrable*			
Aggregated by	*NmCluster*.nmNode			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
allNmMessages KeepAwake	Boolean	0..1	attr	Specifies if Nm drops irrelevant NM PDUs. false: Only NM PDUs with a Partial Network Information Bit (PNI) = true and containing a Partial Network request for this ECU trigger the standard RX indication handling and thus keep the ECU awake true: Every NM PDU triggers the standard RX indication handling and keeps the ECU awake

▽

△

Class	**UdpNmNode**			
communication Connector	EthernetCommunication Connector	0..1	ref	Reference to the CommunicationConnector that represents the UdpNmNode in the topology description. **Tags:**atp.Status=draft
nmMsgCycle Offset	TimeValue	0..1	attr	Node specific time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.
nmPnHandle MultipleNetwork Requests	Boolean	0..1	attr	Specifies if NM performs an additional transition from Network Mode to Repeat Message State (true) or not (false).

Table A.10: UdpNmNode

B History of Specification Items

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

B.1 Specification Item History of this document compared to AUTOSAR R19-11.

B.1.1 Added Traceables in R20-11

Number	Heading
[SWS_ANM_00091]	
[SWS_ANM_00092]	nmPnHandleMultipleNetworkRequests
[SWS_ANM_00093]	
[SWS_ANM_00094]	Immediate Nm Transmissions
[SWS_ANM_91000]	
[SWS_ANM_91001]	

Table B.1: Added Traceables in R20-11

B.1.2 Changed Traceables in R20-11

Number	Heading
[SWS_ANM_00022]	
[SWS_ANM_00033]	
[SWS_ANM_00035]	
[SWS_ANM_00051]	
[SWS_ANM_00066]	
[SWS_ANM_00070]	
[SWS_ANM_00083]	
[SWS_ANM_00084]	
[SWS_ANM_00085]	
[SWS_ANM_00086]	
[SWS_ANM_00087]	Handling of external wake-up
[SWS_ANM_00088]	Default target state after start-up
[SWS_ANM_00090]	Communication Shutdown

Table B.2: Changed Traceables in R20-11

B.1.3 Deleted Traceables in R20-11

Number	Heading
[SWS_ANM_00031]	
[SWS_ANM_00082]	

Table B.3: Deleted Traceables in R20-11

B.2 Specification Item History of this document compared to AUTOSAR R19-03.

B.2.1 Added Traceables in R19-11

Number	Heading
[SWS_ANM_00087]	Handling of external wake-up
[SWS_ANM_00088]	Default target state after start-up
[SWS_ANM_00089]	allNmMessagesKeepAwake
[SWS_ANM_00090]	Communication Shutdown

Table B.4: Added Traceables in R19-11

B.2.2 Changed Traceables in R19-11

Number	Heading
[SWS_ANM_00004]	
[SWS_ANM_00005]	
[SWS_ANM_00007]	
[SWS_ANM_00008]	
[SWS_ANM_00009]	
[SWS_ANM_00013]	
[SWS_ANM_00022]	
[SWS_ANM_00024]	
[SWS_ANM_00028]	
[SWS_ANM_00031]	
[SWS_ANM_00033]	
[SWS_ANM_00034]	
[SWS_ANM_00035]	
[SWS_ANM_00040]	
[SWS_ANM_00044]	
[SWS_ANM_00047]	
[SWS_ANM_00051]	
[SWS_ANM_00055]	
[SWS_ANM_00062]	
[SWS_ANM_00063]	
[SWS_ANM_00066]	
[SWS_ANM_00067]	
[SWS_ANM_00070]	
[SWS_ANM_00081]	

Table B.5: Changed Traceables in R19-11

B.2.3 Deleted Traceables in R19-11

Number	Heading
[SWS_ANM_00001]	
[SWS_ANM_00039]	
[SWS_ANM_00048]	
[SWS_ANM_00068]	

Table B.6: Deleted Traceables in R19-11