eBUS Specification

Network Management

V 1.0.1

03/2007

Content

1 Change Log

Version	Date	Notes	
1.0	May 2000	Authors:	*M. Scheurer c&s group* *F. Fischer, c&s group* *R. Sanders,Karl Dungs GmbH & Co* *M. Bünnemeyer, Kromschröder* *T. Maier, Lamberti* *H. Falk, TEM* *B. Scheffold, Max Weishaupt GmbH* *A. Konzelmann, Wikon*
1.0	March 2001	translation to English version	by order of User Club eBUS e.V.
1.0.1	März 2007	Change of the footnote from © User Club eBUS e.V. www.eBUS.de to © eBUS Interest Group www.eBUS.de	Frank Hoffmann

2 Reference Documents

Document 'OSEK/VDX Network Management, Concept and Application Programming Interface' serves as reference for the development of the Network Management pertaining to the eBUS.

3 About this Document

The *Specifications Network Management for the eBUS* consists of the following:

- Description of the functionality of services between network management and application
- Description of the functionality of services between network management and eBUS Drivers
- Description of station-internal and station-spanning algorithms for warranting functionality of described services of the network management
- Boundary values and interpretation of the network management

The type of implementation is left wide open to enable application-specific optimisation.

4 Introduction

The duty of the network management is to enable safe operation and interaction of all nodes within the net. The network performs this task through:

- Determination of those bus members that are needed for proper function
- Configuration monitoring of those bus members that are needed for proper function
- Monitoring of its own node
- Provision of status information for application, in order to enable application-specific reaction

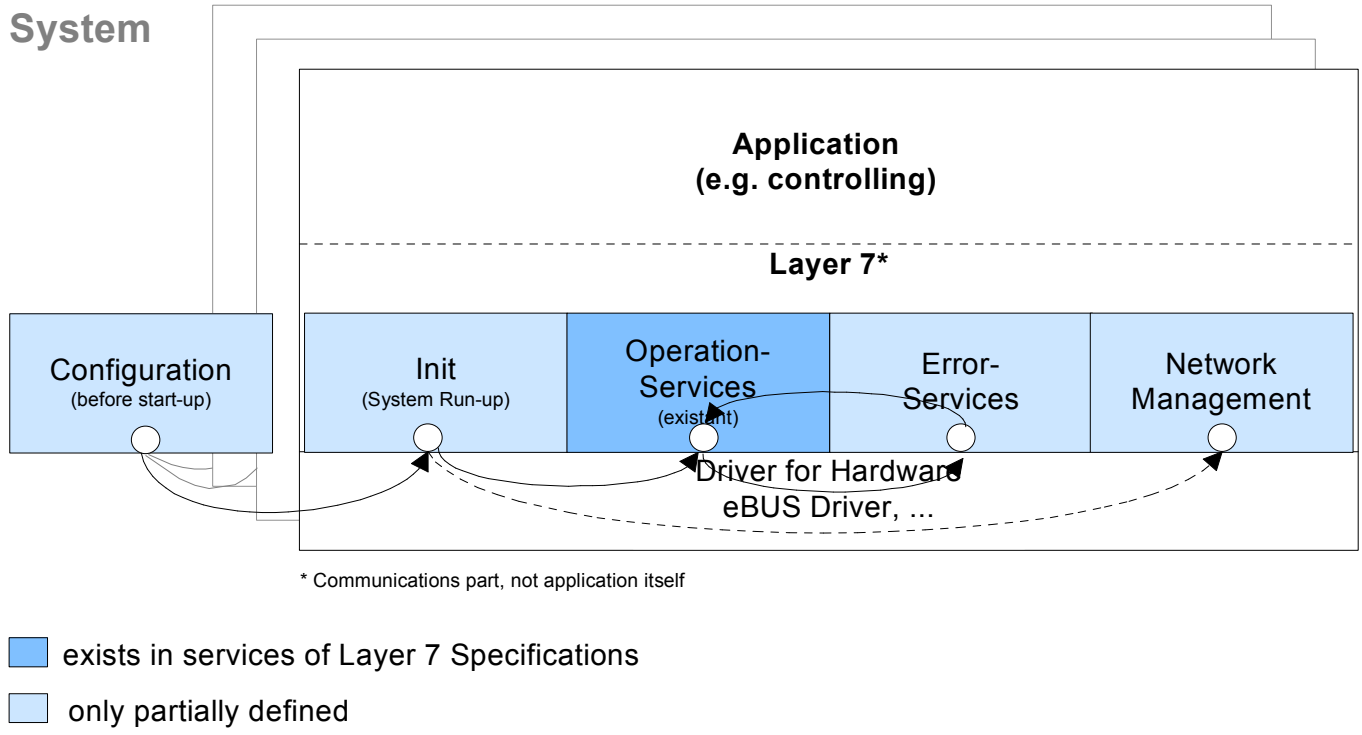
The network management for the eBUS follows the concept of indirect network management[1].
The indirect network management determines and monitors the configuration by means of cyclic application messages. Cyclic messages of individual nodes are, therefore, mandatory for the functionality of the network management.

The available specification of the network management cover the following required boundary conditions:

- Minimal requirement of resources
- Small implementing effort
- No additional bus load
- Definition of functionality of interfaces to the application and to the eBUS driver
- Data formats of interfaces not prescribed
- Standardised error messages
- Nodes on bus, that have no network implementation, will not influence the functionality of the network management in the other nodes on the net
- $\Rightarrow$ implementing of the network management is optional
- Slaves do not have any network management, each slave is monitored by all masters that needs it for their functionality
- Application in large as well as small systems
- excellent testability

[1] OSEK/VDX Networkmanagement, Concept and Application Programming Interface, Version 2.5

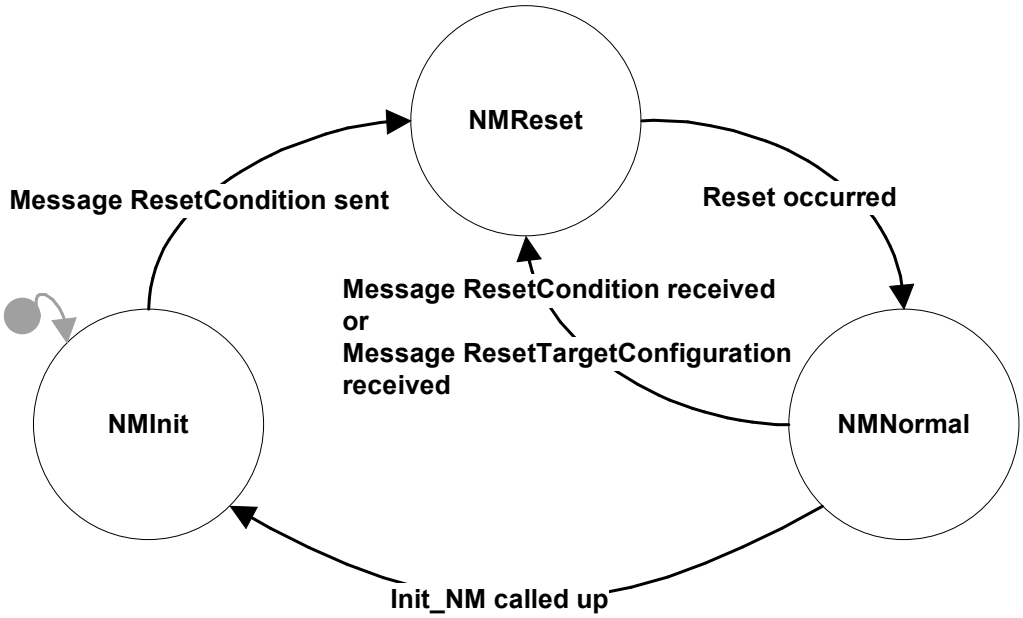
5 Classification the Network Management in eBUS Nodes



Picture 1: Classification of Network Management in eBUS Node

6 States of the Network Management

The network management features three different states, namely:
- Initialising phase (NMInit)
- Reset phase (NMReset)
- Monitoring phase (NMNormal)



Picture 2: Diagram showing Conditions of the Network Management

Once a member has been connected to the eBUS, network management of the newly added node is in condition NMInit. In this state, a message called ResetCondition is issued. This will cause a change of state with the already existing bus members to NMReset. The newly joined member also switches over to NMReset state. In this state, status chart, net status and internal variables of the implementation are all set to their default values (Reset). The start flag is set.
If a dynamic target configuration[2] is to be used, but has not been generated yet, the target configuration is initialised.
When a member has received the message ResetTargetConfiguration, an already generated dynamic target configuration will be erased and the target configuration is re-initialised.
Then, an immediate switch to condition NMNormal takes place. In this state, if required, the target configuration is generated dynamically and the dynamic or static target configuration is monitored.
All nodes that are to be monitored are initially judged as 'in good order' (OK), therefore, the net status is OK.
The start flag, having been set in status NMReset, the network management of the application enables judgement as to whether a node is listed as OK because a reset has occurred (cycle time not yet expired, therefore not yet checked) or if it is considered OK while in monitoring mode.
The start flag is reset, when the maximal (longest) cycle time has expired during which messages are expected.
All monitored nodes that are registered as OK following setting of the start flag, are regarded as existent.
In case new errors are recognised (own node is not capable of sending or a monitored node has failed), a one time only error message[3] (failure message) is issued. As an option, this error message may be repeated with a minimum of 15 minutes interval, as long as at least one node is considered as having failed.
If during monitoring service Init_NM is called up, the network management of the node changes to the condition NMInit.

[2] dynamic target configuration, see chapter 'Target Configuration'
[3] error messages are standardised in eBUS Specification Application Layer - OSI 7 V 1.4

7 Interface Specifications: Data Content

The interface specifications ref. data content of the network management not only determines the data that must be made available to the network management, but also the data that are made available by the network management.
The respective data content is listed below. The data types are not pre-determined; the way of implementation is largely left to the designer.

Target Configuration	The target configuration is the listing of all to be monitored nodes, including (implicitly) of the proper node. The eBUS addresses serve as the identification of the respective nodes
Condition Chart	The condition chart contains a message for all to be monitored nodes (including the proper node). The message reflects the state of each node that is being monitored (node exists / Node does not exist)
Cycle Times	The time frame, during which a message is expected from the monitored nodes, or respectively, during which a message is to be issued by the proper node. There are two options: - the default cycle time applies to all nodes that are being monitored - specific cycle time for each individual node
Start Flag	The start flag is set following a reset and is reset following expiration of the longest to be monitored cycle time.
optional: Net Status	The net status represents a sum information. Net status = OK, if all to monitored nodes exist and the proper node is OK. Otherwise, net status = NOK

7.1 Cycle Times

A network management node must have cycle time information for each of the monitored nodes, during which it expects a message from the respective node.
The network management must further have knowledge of the time frame during which it must issue a message.

The cycle time is application-specific and must, therefore, be adapted to the respective application. Two variations of time framing are available:

- Using a default cycle time, applicable to all nodes
- Using node-specific cycle times for each node

Specific cycle times must be dedicated to each of the monitored nodes including the proper node, as well as made available to the network management.

7.2 Conditions Chart

In the conditions chart, an entry must exist for each of the monitored nodes as well as the proper node. The entries must be dedicated to the respective nodes.
If a node has failed to report during the allocated cycle time or if the proper node has been unable to send a message during the time allocated to it, the respective entry in the condition chart will be set to NOK. As soon as the network management has received a message from one of the monitored nodes or if the proper node has issued a message, the respective entry in the condition chart will be set to OK.
The condition chart is updated by the network management at every message issue and receipt or on request (service GetStatus), then made available to the application.

Following a reset, the values in the condition chart at covered by the default value OK. An OK in the condition chart makes no judgement on the true state of the monitored node as long as the start flag is set. The start flag is reset, when the longest of all monitored cycle times has expired once. Only then is it safe to say that all monitored nodes have been checked for potential issuing of a message within the respective allocated cycle time.
In case a message from a node with a shorter cycle time is not received, the condition of this node is set to NOK.
Note: NOK represent a valid statement regarding the condition of a node, even when the start flag is set.

7.3 Net Status

The net status represents an information summary regarding the conditions of all monitored nodes, including ones' proper node.
The net status is assumed as 'in good order' (OK), if the condition of all monitored nodes is indicated as OK and if the monitoring node is capable of sending. It is considered as 'not in good order', if at least one node in the status chart is marked as NOK or if the monitoring node is incapable of sending.
Following the reset, the net status is engaged with its default value OK. An OK of the net status makes no real statement about the true status of all members of the bus, as long as the start flag is set.
Only after the start flag is reset, it is safe to say that all monitored nodes have been checked for sending a message within their respective preset cycle time. A NOK of the net status is an indicator for an error condition of at least one node, even if the start flag is set.
The provision of the net status by the network management is optional. It is, however, recommended.
If the network status as sum information is waived, the application has to pass through the entire status chart, in order to receive a statement about the monitored target configuration. If a sum information is furnished, a complete pass through the status chart is required only, if a NOK is displayed and the error nodes are to be specified.

7.4 Start Flag

All monitored nodes are initially carried as 'in order' (OK) in the status chart; the net status is, therefore, OK.
With the start flag set under NMReset condition, the network management of the application enables the distinction if a node in the status chart is entered as OK, because a reset has been carried out (cycle time for the node potentially not yet expired and node, therefore, not yet checked) or if it is seen as OK by the running monitoring service.
The start flag is reset, as soon as the maximal (longest) cycle time for expected messages has expired. All monitored nodes that are listed as OK after resetting of the start flag, are considered as definitely existing.

7.5 Target Configuration

The network management of a node monitors those members that the node requires for its functionality.
The addresses of these members must be made available as target configuration. For this, there are two possibilities:

- the target configuration may be furnished statically, as far as known on start-up and unless changed during operation
- the target configuration may be generated dynamically during start-up.

In the target configuration, the eBUS addresses serve as identification for the individual nodes that are to be monitored. In the target configuration, slaves as well as masters are listed. In the target configuration, the eBUS address of the proper node does not explicitly need to be listed, because its address is known.

Example:

Node 1:	needs the exterior temperature, the interior temperature and the clock time.
Node 2(Slave):	may furnish exterior temperature on request.
Node 3(Master):	cyclically furnishes a telegram that contains the interior temperature
Node 5(Master):	furnishes a telegram that contains the clock time

⇒ In the target configuration of node 1, eBUS addresses of nodes 2, 3 and 5 are explicitly included. Also included is the eBUS address of the proper node.

7.5.1 Static Target Configuration

The target configuration may be advanced to the network management in static form, as far as the to be monitored addresses are known at the time of start-up and provided that they will not be altered during the operation.
The static target configuration must remain available to the network management for the entire period of operation.
In case cycle times other than the default cycle times are to be considered for the monitoring of the nodes, the respective cycle time must be assigned in the target configuration of each node.

7.5.2 Dynamic Generation of Target Configuration

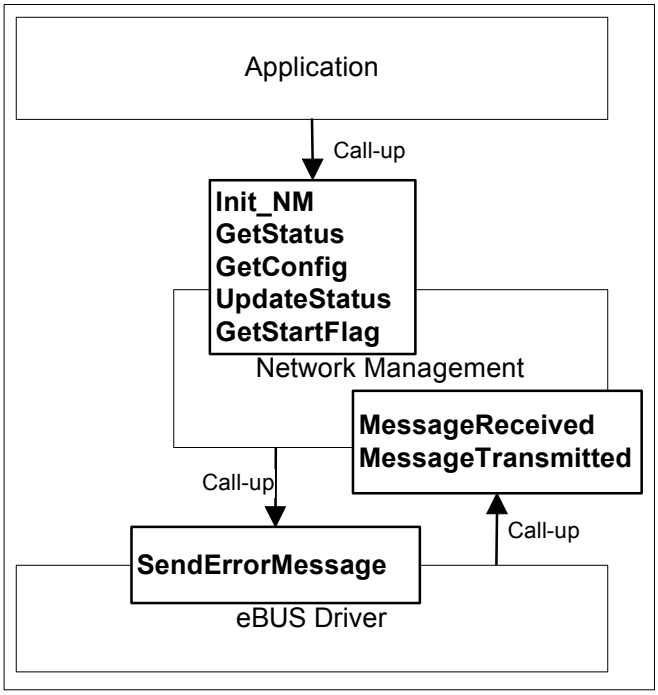
If the top be monitored addresses are not yet known on start-up or if the to be monitored addresses may be changed during operation, the target configuration must be generated dynamically during operation.
In this case, the target configuration is complemented with one value each 0xAA for each requested service. This is accomplished in status NMReset and in addition to the eBUS address of the proper node.
For each additional -optional- service, which is not absolutely needed by the node for its proper functionality, that it however uses in case it receives this service, an ‘address’ 0xA9 is written into the target configuration.
The appropriate cycle times of the services are assigned to these ‘addresses’ 0xAA and 0xA9.
In status NMNormal, the target configuration is filled under application-specific rules.
This initialisation of the target configuration is carried out only if a target configuration is empty or of message ResetTarget Configuration has been received.
When a required service is received, the corresponding entry 0xAA is overwritten or reset by the eBUS address of the sender, provided the sender is already present in the target configuration.
The existence of 0xAA in the target configuration leads to issuing of the error message, if the associated monitoring cycle time has expired. In the data segment, this message marks the address 0xAA as ‘failed’.
With the receipt of an optional service, the associated entry 0xA9 by the eBUS address of the sender is overwritten or reset respectively, in case the sender already exists in target configuration.
The existence of 0xA9 in the target configuration does not lead to the release of the failed-message, when the associated monitoring cycle time has expired.
A proposal for a network management implementation with application-specific rules for the dynamic generation of the target configuration is found in chapter ‘Flow charts of the network management’.

7.5.3 Combination static Target Configuration and dynamic Generation of Target Configuration

In case only some of the to be monitored addresses are known at time of start-up, it will be possible to apply a combination of static target configuration and dynamic generation of target configuration.
In this case, the target configuration is initialised with the known eBUS addresses as well as with 'addresses' 0xAA, i.e. for the required services and 'addresses' 0xA9 for optional services.
Allocation of respective cycle times is done as already described.

8 Interface Specification: Services

The interface between network management and application, network management and eBUS driver respectively is described below through service call-ups[4].
The data (see also chapter 'Interface Specifications: Data Content') may be exchanged through direct referencing on the occasion of service call-ups or through joint access to global variables. Processing of the specified tasks of the services is incumbent on the implementer.



Picture 3: Architectural Model

8.1 Interface Services between Network Management and Application

Init_NM	Starts network management.
UpdateStatus	Updates status chart and, if applicable, net status.
GetConfig	Delivers status chart with status of monitored nodes.
GetStartFlag	Obtains start flag.
optional: GetStatus	Obtains a sum information about the status of the to be monitored configuration and about its proper node (net status)

8.2 Interface Services between Network Management and eBUS Driver

MessageReceived	Triggers updating of status charts and, if applicable, net status (UpdateStatus).
MessageTransmitted	Triggers updating of status charts and, if applicable, net status (UpdateStatus).
SendErrorMessage	Sends an error telegram on the eBUS, as soon as a monitored node has failed.

[4] An Implementation is not mandatory necessary. The description is only used for the further understandig of the funtional definitions of the Network Managment.

8.3 Network Management Messages

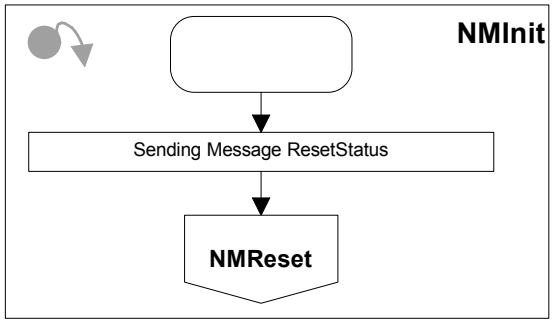
In the eBUS specification application layer – OSI 7, respective services are defined. When implementing a network management, the support of at least the following services is required:

Service FFh 00h	Reset status NM
Service FFh 02h	Failure message

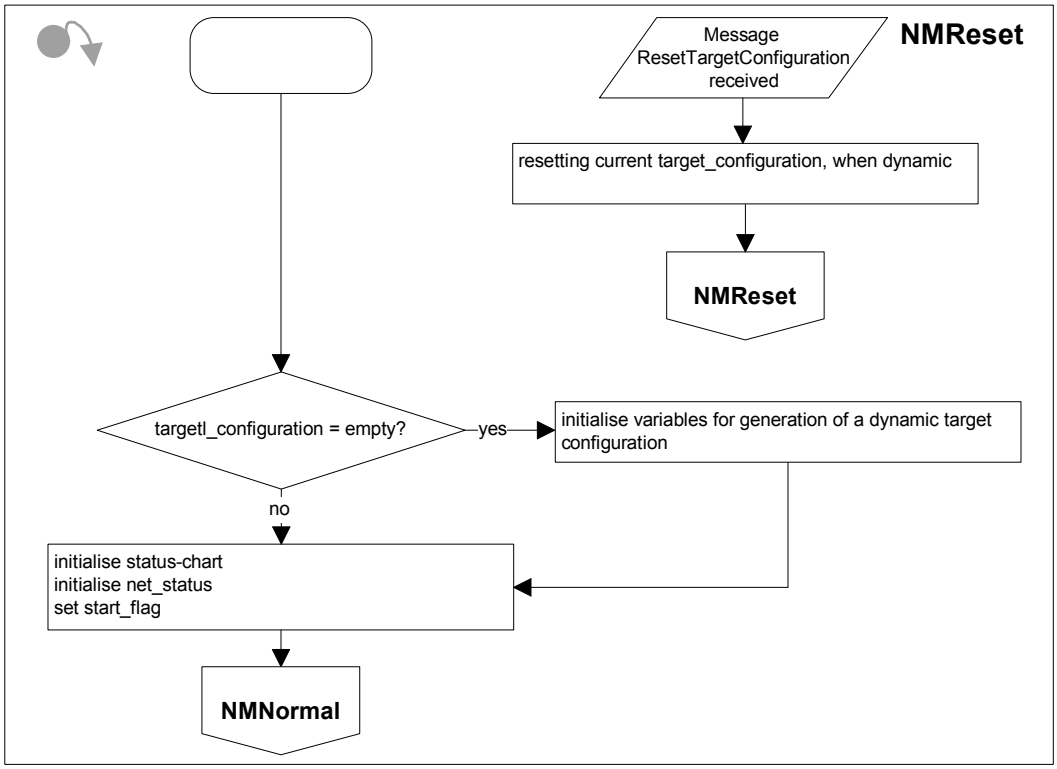
Support of additional services to the network management is optional.

9 Flow Charts of the Network Management

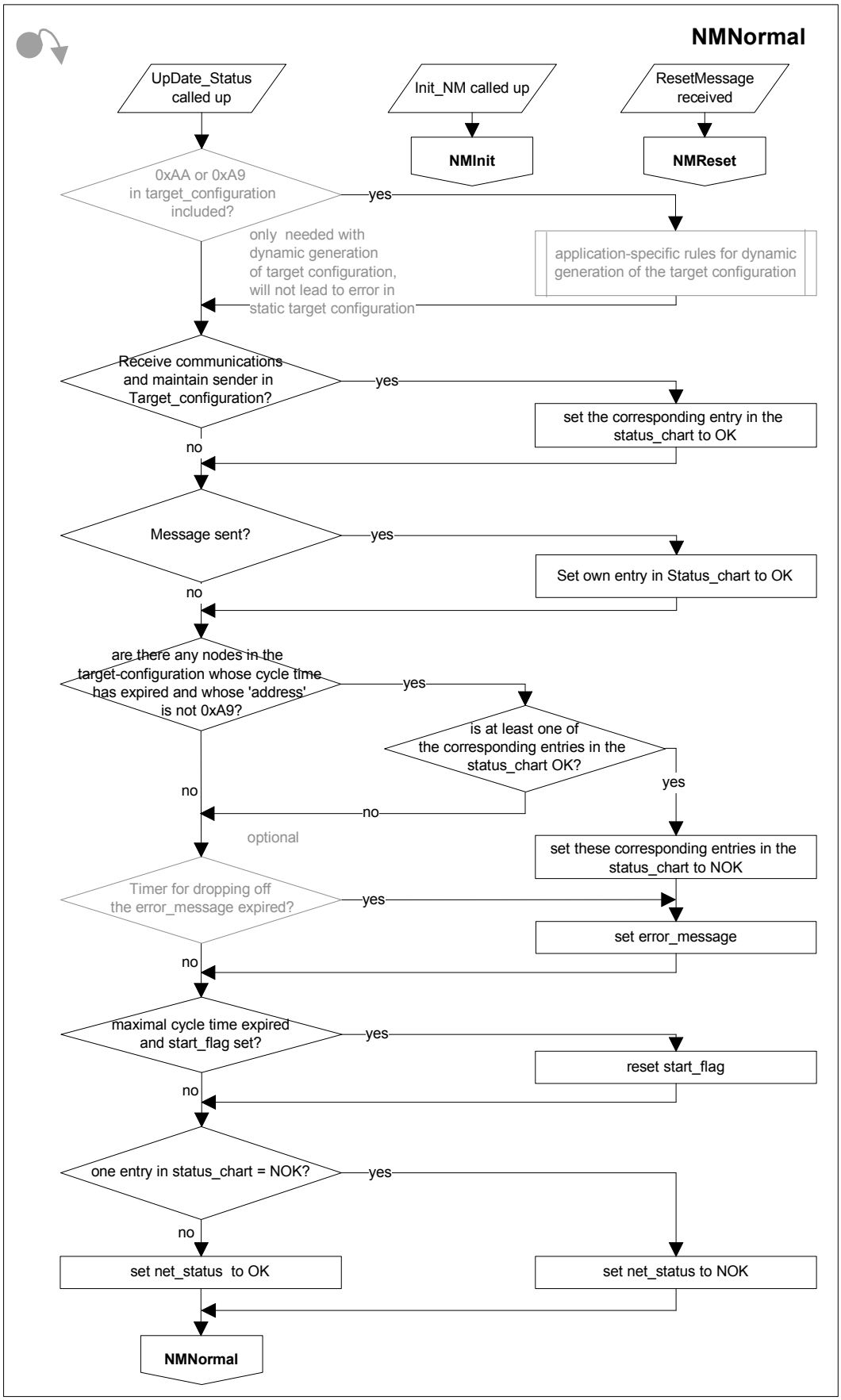
9.1 Flow Charts NMInit, NMReset and NMNormal



Picture 4: Flow Chart NMInit



Picture 5: Flow Chart NMReset



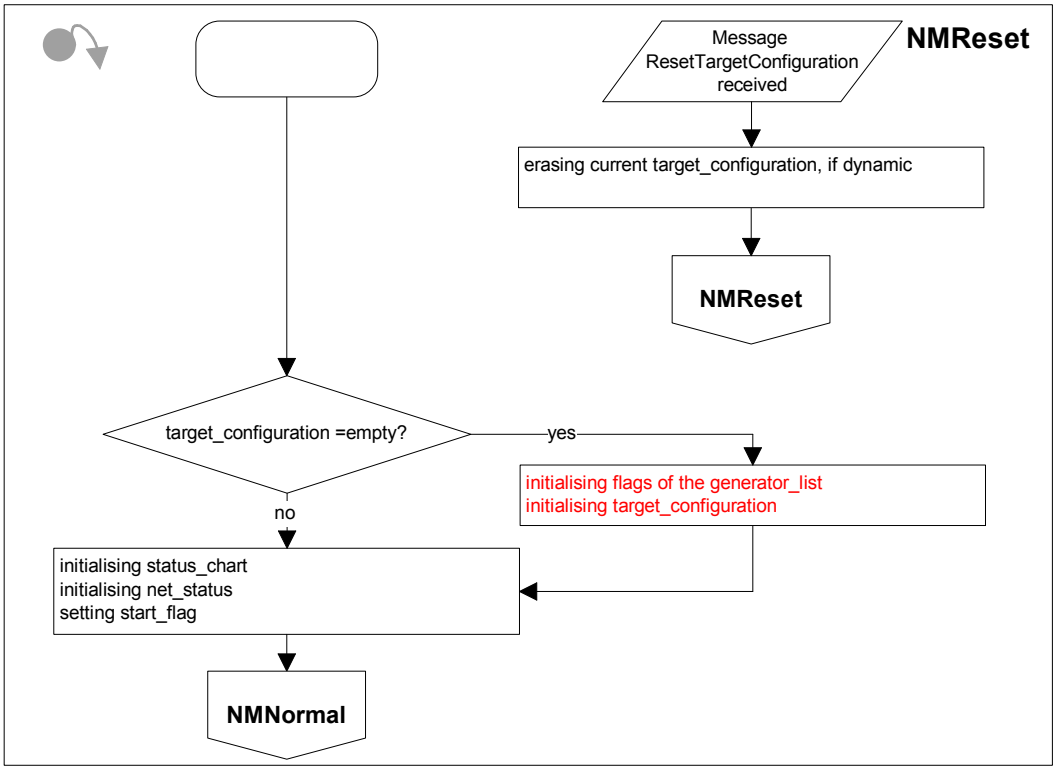
Picture 6: Flow Chart NMNormal

9.2 Proposal for Network Management Implementation with application-specific Rules for dynamic Generation of Target Configuration

A node requires certain data that are made available by other members by means of the Service PB SB. The address of this member is unknown at time of start-up. Therefore, the target configuration must be generated during operation.
This is why the node maintains a generator_list. This contains:

- the services that are required by the node for its functionality and whose sender it shall monitor
- for each service a flag received/not received
- for each service its corresponding cycle time, provided that a given cycle time is applicable for the services

From the generator_list, the target configuration is generated during service. Below, please find a conceivable sequence of the target configuration finding is shown (behaviour in NMInit as with static target configuration).



Picture 7: Proposal for dynamic Generation of the Target Configuration – Flow Chart for Status NMReset

At change to status NMNormal, generator_list, target_configuration, status_chart, start_flag and net_status are initialised as follows:

PB 1	SB 1	Time 1	Not received	...	PB n	SB n	Time n	Not received	Generator_list

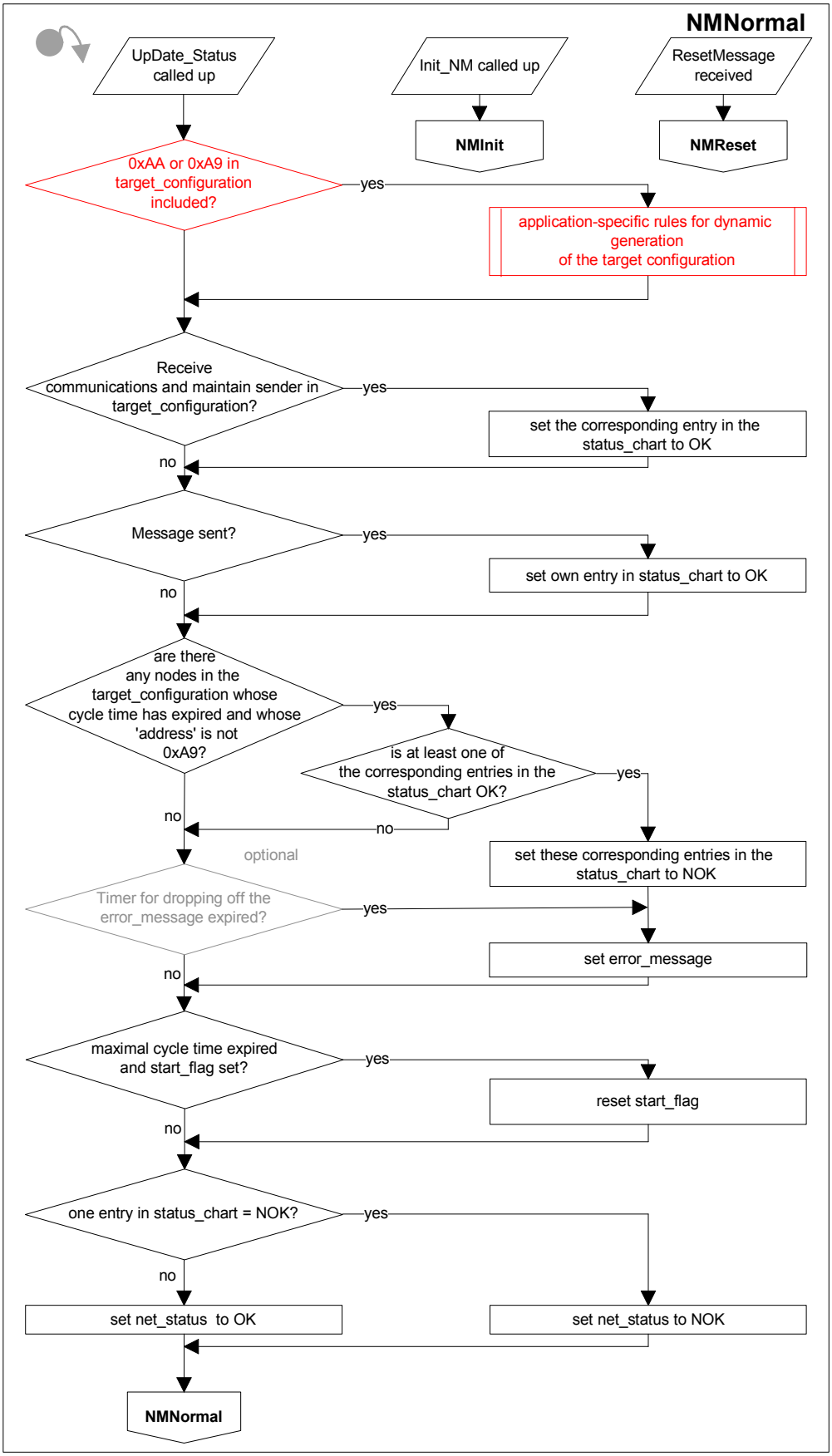
0xAA	...	0xAA	Own address	Target_configuration

Time 1	...	Time n	Own time	Cycle times

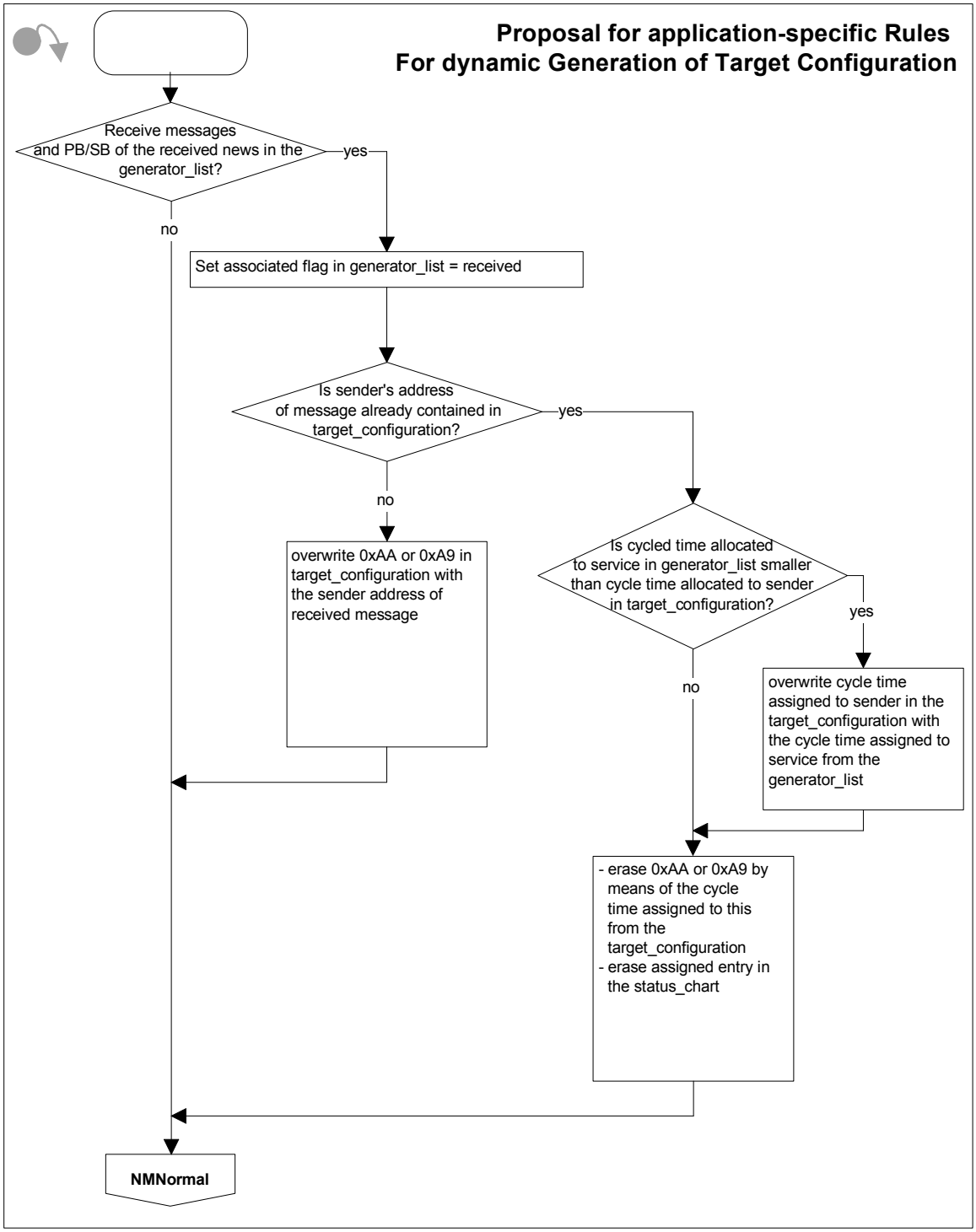
OK	...	OK	OK	Status_chart

OK	Net_status

Set	Start_flag



Picture 8: Proposal for dynamic Generation of Target Configuration – Flow Chart for Status NMNormal



Picture 9: Proposal for the dynamic generation of the target configuration – Flow Chart for application-specific rules

After transmitting an own message and receipt of a communication with PB = PB1 and SB = SB1 Address 0xXX, generator_list, target_configuration, status_chart and net_status are achieved as follows (Note: changes are highlighted with grey back ground):

PB 1	SB 1	Time 1	Received	...	PB n	SB N	Time n	Not received	Generator_list

0xXX	...	0xAA	Own address	Target_configuration

Time 1	...	Time N	Own time	Cycle times

OK	...	OK	OK	Status_chart

OK	Net_status

Set	Start_flag

Following receipt of additional communications that meet the required services, generator_list, target_configuration, status_chart and net_status result, after the last required service has been received (Note: changes after receipt of last message are highlighted using dark grey back ground):

PB 1	SB 1	Time 1	Received	...	PB n	SB N	Time n	Received	Generator_list

0xXX	...	0xZZ	Own address	Target_configuration

Time 1	...	Time n	Own time	Cycle times

OK	...	OK	OK	Status_chart

OK	Net_status

Not set	Start_flag

The network management is now conscious of target configuration it must monitor.
A change in the net configuration (e.g. exchange of a node that supports a required service for a node with another address) leads to a re-generation of the target configuration by the use of the function NMReset.

This method of resolution for the dynamic generation of a target configuration is structured such as to easily allow respective implementation also for a network management node with static target configuration.

10 Proposals for Implementation

The following implementation examples show the respective memory requirements per node to be monitored as well as the suggested available memory capacity.

Each of these implementation examples is illustrated by means of a graph. This graph refers to a net with three nodes, in which one node 0x0F monitors the nodes 0x03 and 0x07 as well as itself.

Note:
The illustration of cycle times is based on 8 Bits each. When selecting appropriate resolution, the cycle time may be represented with less than 8 Bits. With application of specific cycle times, the resource requirement would thus be drastically reduced!

10.1 Proposal of implementation with static target configuration and a default cycle time

Data Definition of Interfaces:

Target configuration	Per monitored node, one byte is required for the respective eBUS Address For the eBUS Address of its own node, zero bytes are required, since the own address is known
Status chart	Per monitored node one bit: The status of the respective node is represented by the bit (1 = OK / 0 = NOK) For representation of the own node, one bit is required
Cycle time	One Byte
Start flag	One Bit (1 = ON / 0 = OFF)
Net status	One Bit (1 = OK / 0 = NOK)

Definition of internal variable for network management implementation

Current time data (for monitoring cycle times)	One byte per monitored node (as for cycle time) One byte for own node

Thus, 17 bits are required for each monitored node, namely:

Target configuration	8	bits
Status chart	1	bit
Current time data	8	bits
	∑17	**bits**

11 additional bits are required for monitoring own node, for start flag and for net status, namely:

Status chart	1	bit
Current time data	8	bits
Start flag	1	bit
Net status	1	bit
	∑11	**bits**

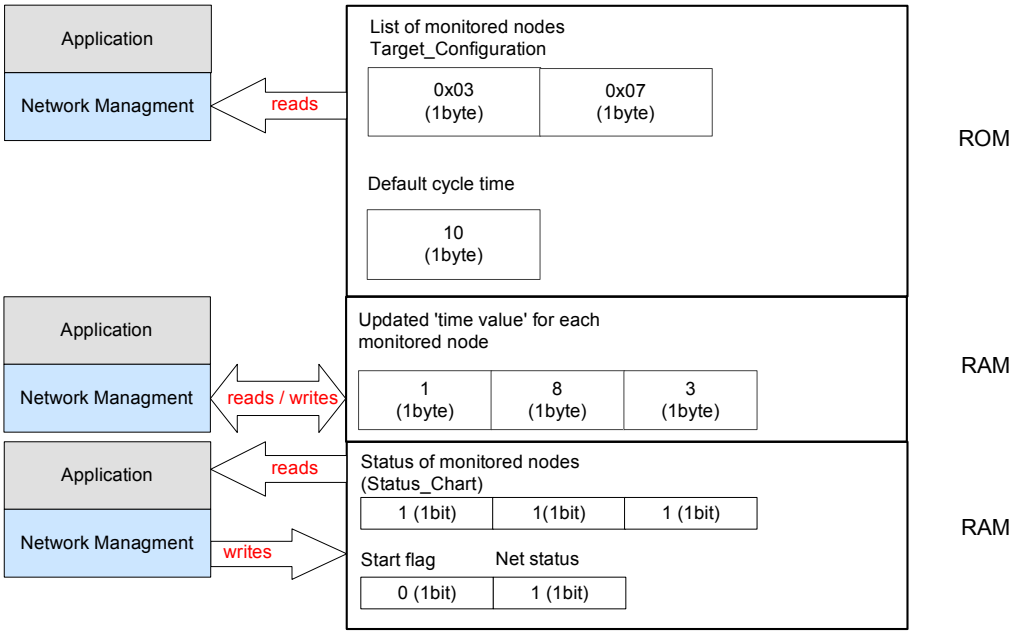
Default cycle time is represented by 8 bits.

Sum: 19 Bit + (number of monitored nodes · 17 bits)

The following graph shows one possible realisation of the implementation proposal for a master 0x0F.
Master 0x0F monitors master 0x03 and master 0x07 as well as itself.
The default cycle time is 10 minutes.

Resource requirement without source code is 53 Bit

Note: the cycle time in this graph is represented by 4 bits. It is based on the resource requirement for this example of 37 bits



Picture 10: Static Target_Configuration with default cycle time – Realisation of an implementation proposal

10.2 Proposal of an Implementation with static Target Configuration and specific Cycle Times

Definition of Interface Data:

Target configuration	One byte per monitored node is required for the respective eBUS address Zero bytes are required for the eBUS address of the own node, as the own address is known
Status chart	One bit per monitored node: The status of the respective node is represented by the bit (1 = OK / 0 = NOK) One bit is required for the status of the own node
Cycle time	One byte per monitored node One byte for own node
Start flag	One bit (1 = ON / 0 = OFF)
Net status	One bit (1 = OK / 0 = NOK)

Current time data (for monitoring of cycle times)	One byte per monitored node (as cycle time) One byte for own node

Therefore, 25 bits are required for each monitored node, namely:

Target configuration	8	bits
Status chart	1	bit
Cycle time	8	bits
Current time data	8	bits
	∑ 25	**bits**

Additional 19 bits are required for monitoring own node, for start flag and net status, as follows:

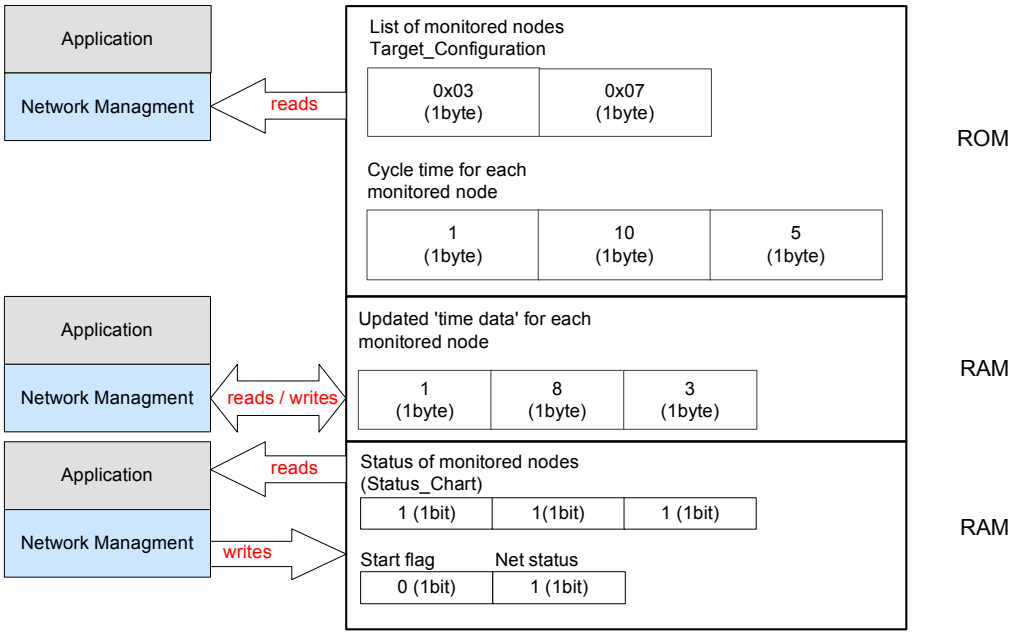
Status chart	1	bit
Cycle time	8	bits
Current time data	8	bits
Start flag	1	bit
Net status	1	bit
	∑19	**bits**

Sum: 19 bits + (Number of monitored nodes · 25 bits)

The following illustration shows the feasible realisation of the implementation proposal for one master 0x0F. Master 0x0F monitors master 0x03, master 0x07 as well as itself.
Within max. 1 second, a message is expected from master 0x03, within max. 10 minutes, a message is expected from master 0x07. Master 0x0F shall send a message not later than after 5 minutes.

Resource requirement without source code: 69 bits

Note: when depicting the cycle time through 4 bits, the resource requirement for the example is about 45 bits



Picture 11: Static Target_Configuration with specific cycle times – Realisation of an implementation proposal

10.3 Proposal of an Implementation with dynamic Generation of the Target Configuration and a Default Cycle Time

Data Definition of Interfaces:

Generator list	Two bytes are needed per required service for primary and secondary commands, plus one bit for the flag 'Received/not received'
Target configuration	One byte is needed per monitored node for respective eBUS address Zero bytes are needed for the eBUS address of the own node as own address is known
Status chart	One bit is needed per monitored node: The status of the respective node is represented by the bit (1 = OK / 0 = NOK) One bit is needed for status of own node
Cycle time	One byte
Start flag	One bit (1 = ON / 0 = OFF)
Net status	One bit (1 = OK / 0 = NOK)

Definition of internal Variables of Network Management Implementation

Current time data (for monitoring of cycle times)	One byte per monitored node (as cycle time) One byte for own node

For each monitored node, 17 bits are needed, namely:

Target configuration	8	bits
Status chart	1	bit
Current time data	8	bits
	∑17	**bits**

17 bits are needed in the generator list for each required service, namely:

Primary command	8	bits
Secondary command	8	bits
Flag	1	bit
	∑ 17	**bits**

Additional 11 bits are needed for monitoring own node, start flag and net status, namely:

Status chart	1	bit
Current time data	8	bits
Start flag	1	bit
Net status	1	bit
	∑11	**bits**

The default cycle time is represented by 8 bits.

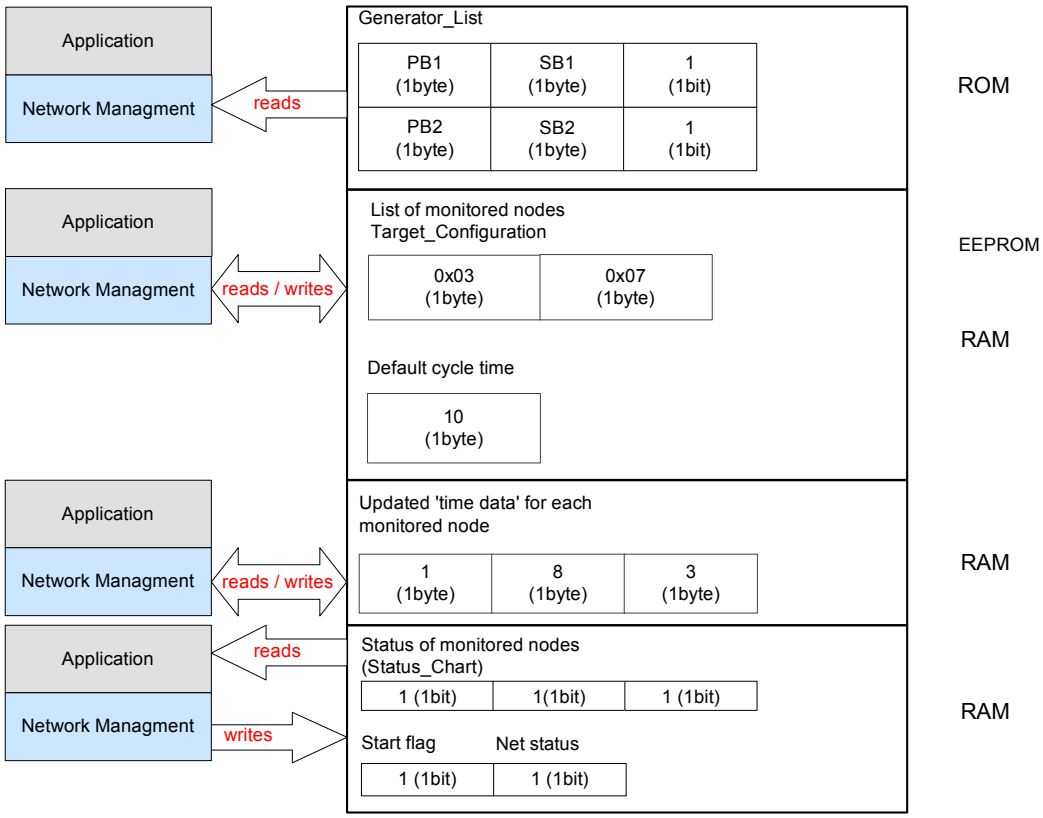
Sum: 19 bits + (Number of monitored nodes · 17 bits) + (Number of required services · 17 bits)[5]

[5] 'Worst Case' view: each of the required services is provided by another node ⇒ Number of required services = Number of monitored nodes (max. memory requirement may be calculated)

The following illustration again shows the potential realisation of the implementation proposal for one master 0x0F.
For its proper functionality, master 0x0F requires service 1 and service 2. Throughout the operation, it receives service 1 data from master 0x03 as well as data of service 2 from master 0x07.
In this configuration, master 0x0F monitors master 0x03, master 0x07 as well as itself.
The default cycle time is 10 minutes.

Resource needs excluding source code: 87 bits

Note: in the realm of representation of cycle time by 4 bits, the resource need for this example is at 71 bits.



Picture 12: Dynamic generation of the Target_configuration with default cycle time – Realisation of an implementation proposal

10.4 Proposal of an Implementation with dynamic Generation of Target Configuration and specific Cycle Times

Data Definition of Interfaces:

Generator_list	Two bits are needed per required service for primary and secondary command, one bit for the flag 'Received/not received' and one byte for the specific cycle time
Target configuration	One byte is needed per monitored node for the respective eBUS address No bytes are needed for the eBUS address of own node as the own address is known
Status chart	One bit per monitored node: The status of the respective node is represented the bit (1 = OK / 0 = NOK) One bit is needed for the status of the own node
Cycle time	One byte per monitored node One byte for own node
Start flag	One bit (1 = ON / 0 = OFF)
Net status	One bit (1 = OK / 0 = NOK)

Definition of internal Variables of the Network Management Implementation

Current time data (ref. Monitoring of cycle times)	One byte per monitored node (as cycle time) One byte for own node

Thus, 25 bits are needed for each monitored node, namely:

Target configuration	8 bits
Status chart	1 bit
Cycle time	8 bits
Current time data	8 bits
	∑25 bits

In the generator list, 25 bits are needed for each required service, namely:

Primary command	8	bits
Secondary command	8	bits
Cycle time	8	bits
Flag	1	bit
	∑25	**bits**

Additional 19 bits are needed for monitoring own node, for start flag and net status, namely:

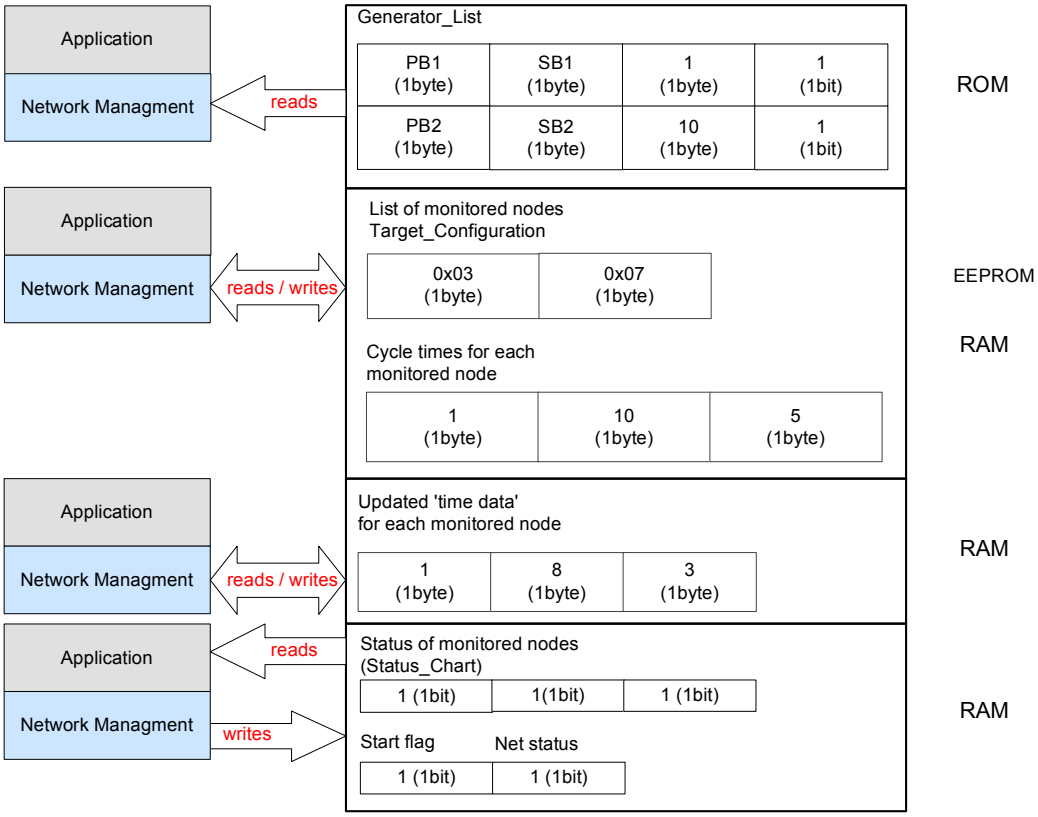
Status chart	1	bit
Cycle time	8	bits
Current time data	8	bits
Start flag	1	bit
Net status	1	bit
	∑19	**bits**

Sum: 19 bits + (Number of monitored nodes · 25 bits) + (Number of required services · 25 bits)[6]

The following illustration depicts the potential realisation of the implementation proposal for one master 0x0F. Master 0x0F needs service 1 after max. 1min. and service 2 after max. 10 min. for its functionality. During service, it receives the data of service 1 from master 0x03 and the data of service 2 from master 0x07. Herewith, master 0x0F monitors master 0x03, master 0x07 as well as itself.

Resource requirements excluding source code: 119 bits

Note: in the realm of representation of cycle time by 4 bits, the resource needs for the example is at 87 bits



Picture 13: Dynamic generation of the Target_Configuration with specific cycle times – Realisation of an implementation proposal

[6] 'Worst Case' Contemplation: each of the required services is supplied by another node ⇒ Number of required services = Number of monitored nodes (max. memory requirement may be calculated)

11 Picture Directory