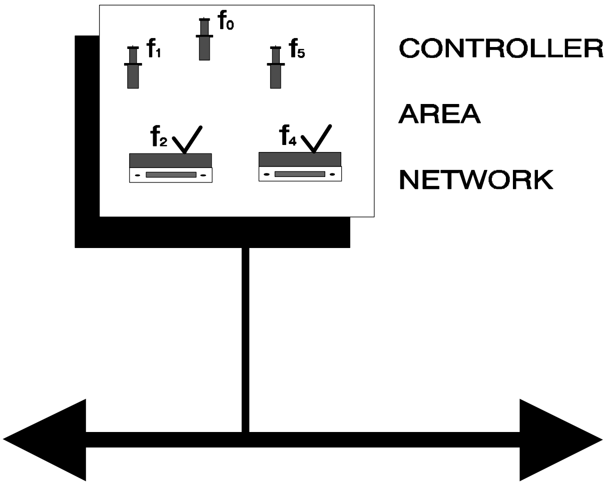
CAN in Automation (CiA)
International Users and Manufacturers Group e.V.



CAN Application Layer for Industrial Applications
CiA/DS203-2
February 1996

NMT Protocol Specification

1. SCOPE

This document contains the protocol specification of the Network Management (NMT). NMT is part of the CAN Application Layer. This document is part of a set of documents that standardize the CAN Application Layer for Industrial Applications.

2. REFERENCES

/1/: CiA/DS201, CAN Reference Model

/2/: CiA/DS203-1, NMT Service Specification

/3/: CiA/DS207, Application Layer Naming Conventions

3. GENERAL DESCRIPTION

3.1 NMT Protocol Perspective

The Network Management (NMT) service element in the CAN Reference Model (see /1/), provides the NMT services. The NMT Protocol is executed between between the NMT Master and each of the NMT Slaves (see /2/) to implement these services.

3.2 NMT Slave Synchronization

Since in the NMT Protocol all NMT Slaves use the same COB to send information to the NMT Master, there must be only one NMT Slave at a time that communicates with the NMT Master. For all protocols except the Identify Node protocol, the NMT Master takes the initiative. The Identify Node protocol does not transfer information and will therefore cause no synchronization conflicts. For all other services an NMT Slave is only allowed to respond if it has first been addressed by the NMT Master through its unique NMT Address or Node-ID. Since there can be atmost one confirmed NMT service outstanding at a time (see /2/), the synchronization is established.

3.3 NMT Protocol Descriptions

A protocol description specifies the sequence of COB's and their format that are exchanged between the NMT Master and NMT Slave(s) for a particular NMT service. In the description of the COB data format, bytes are numbered from 0 to and including 7. Bits within a byte are numbered from 0 to and including 7. Byte 0 is transmitted first, byte 7 is transmitted last. Within a byte, bit 0 is the least significant bit, bit 7 is the most significant bit. In the protocol descriptions [a, b] denotes the range of integers from a to b with a and b included. If a > b, the range is empty.

The terms 'lsb' and 'msb' stand for 'least significant byte' and 'most significant byte' respectively and are used to define how an integer number is stored in more than one byte for the NMT Protocol. The order of significance is from lsb to msb.

The NMT protocols transfer the NMT Address (both module-name and module-ID) of an NMT Slave. The transfer syntax of these attributes is defined in /3/.

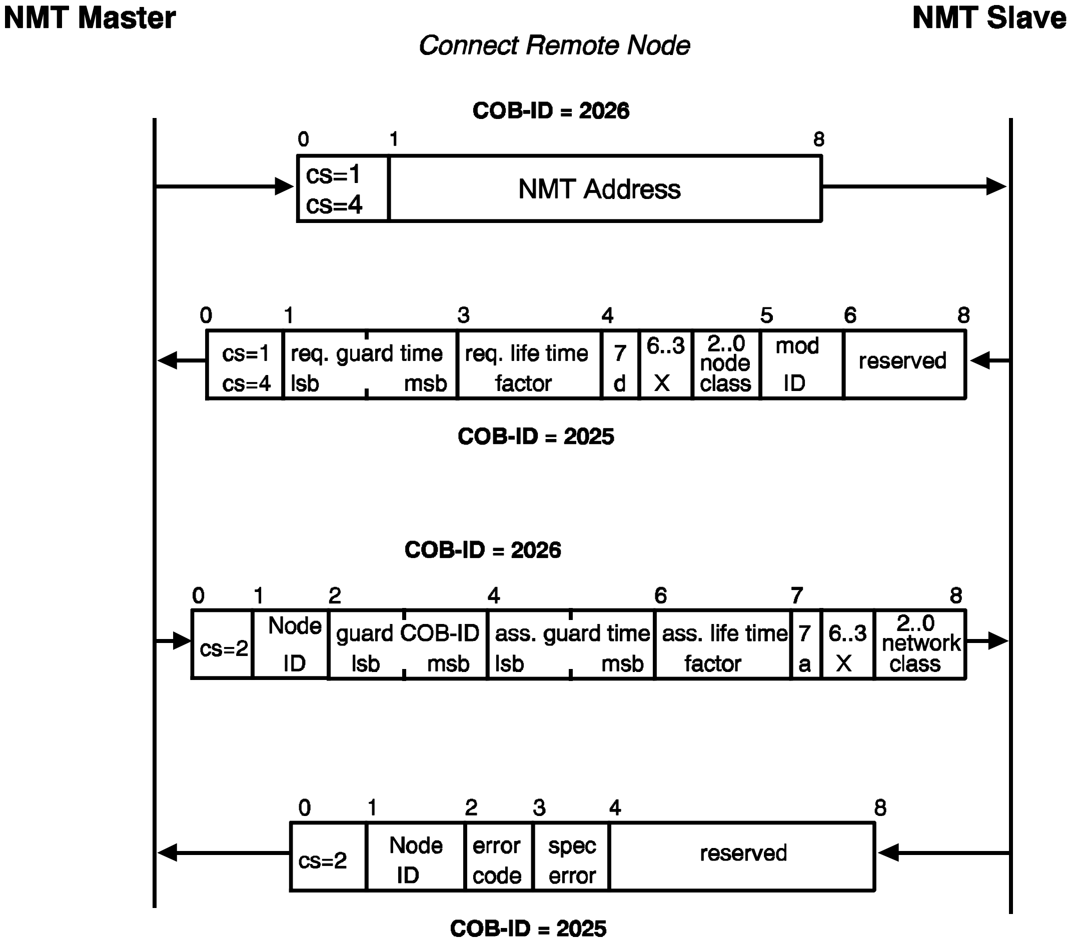
3.4 Usage of Command Specifiers

The command specifier is transmitted within the first data byte of the NMT protocol in COB-ID=2026/2025. For each service a unique value is specified for the command specifier. Unused values are reserved up to 191 for further use by CiA. Values from 192 to 255 are user specific.

4. MODULE CONTROL PROTOCOLS

4.1 Node Connect Protocol

This protocol is used to implement the 'Connect Remote Node' service, see /2/. This protocol also serves to exchange and negotiate parameters for the other NMT protocols.



- **cs:** NMT command specifier.

1: select an NMT Slave by the module-name of its NMT-Address. Bytes [1, 7] contain the module name, see /3/. The selected NMT Slave responds with the same cs in case of a positive response.

2: assign NMT protocol parameters

4: select an NMT Slave by the module-ID of its NMT-Address. Byte 1 contains the module-ID, see /3/. Bytes [2, 7] are reserved for further use by CiA. The selected NMT Slave responds with the same cs in case of a positive response.

- **mod-ID:** the module-ID of the NMT Address of the NMT Slave, see /3/.

- **req. guard time:** the guard time in milli-seconds for the Node Guarding Protocol as requested by the NMT Slave. It is valid if and only if the node class indicates that the Node Error capability has been configured on the NMT Slave, see /2/, otherwise it is reserved for further use by CiA.

- **req. life time factor:** when multiplied with the requested guard time gives the life time for the Node Guarding Protocol. It is valid if and only if the node class indicates that the Node Error capability has been configured on the NMT Slave, see /2/, otherwise it is reserved for further use by CiA.

- **node class:** indicates the node capabilities that have been configured on the NMT Slave according to the definition in /2/.

- **d:** indicates whether or not the NMT Slave needs a configuration to be downloaded
  0: no download requested
  1: download requested

- **Node-ID:** the value of the Node-ID attribute that the NMT Master assigns to the NMT Slave, see /2/. The Node-ID is equal to the module-ID which is passed to the NMT-Master with the NMT-Address attribute when the corresponding remote node object is created.

- **guard COB-ID:** the value of the COB-ID for the Node Guarding Protocol. It must be a value between 1761 and 2015 inclusive. It is valid if and only if the network class indicates that the Network Error capability has been configured on the NMT Master, see /2/, otherwise it is reserved for further use by CiA.

- **ass. guard time:** the guard time in milli-seconds for the Node Guarding Protocol as assigned by the NMT Master. It is valid if and only if the network class indicates that the Network Error capability has been confirgured on the NMT Master, see /2/, otherwise it is reserved for further use by CiA.

- **ass. life time factor:** when multiplied with the assigned guard time gives the life time for the Node Guarding Protocol as assigned by the NMT Master. It is valid if and only if the network class indicates that the Network Error capability has been confirgured on the NMT Master, see /2/, otherwise it is reserved for further use by CiA.

- **network class:** indicates the network capabilities that have been configured on the NMT Master according to the definition in /2/.

- **a**: abortion flag used by the NMT master for signalling an abort of the Connect Remote Node Protocol if the NMT master detects a protocol inconsistency during the first request-response cycle; the NMT slave has to respond with error code 253.

0:	second request valid
1:	abort Connect Remote Node request

- **error code:**

0:	protocol successfully completed
1..252:	reserved for further use by CiA
253:	protocol error indication by NMT master
254:	request not allowed by the node state of NMT Slave
255:	other error occurred

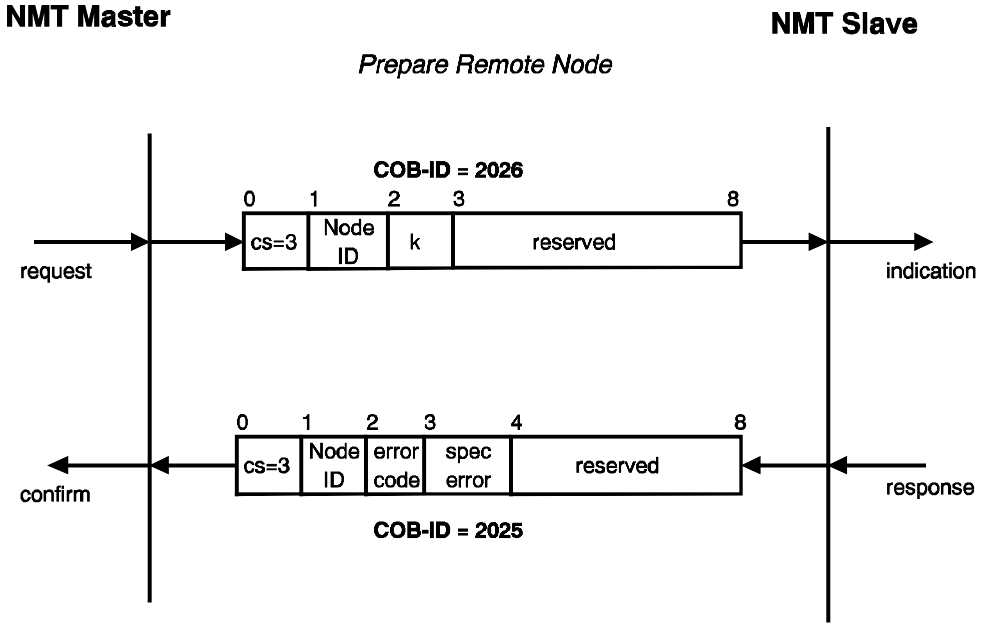
- **specific error code:** a value between 0 and 255 inclusive. If the error code equals 255, it gives an implementation specific error code, otherwise it is reserved for further use by CiA.

- **X:** not used, always 0

- **reserved:** reserved for further use by CiA.

4.2 Node Prepare Protocol

This protocol is used to implement the 'Prepare Remote Node' service, see /2/.



- **cs:** NMT command specifier

3:	prepare

- **Node-ID:** the Node-ID of the NMT Slave as assigned by the NMT Master in the Node Connect Protocol.

- **k:** indicates whether the COB-ID's previously obtained from the DBT must be discarded or not

0:	discard previous obtained identifiers
1:	the NMT Slave may decide to keep the old identifiers

- **error code:**

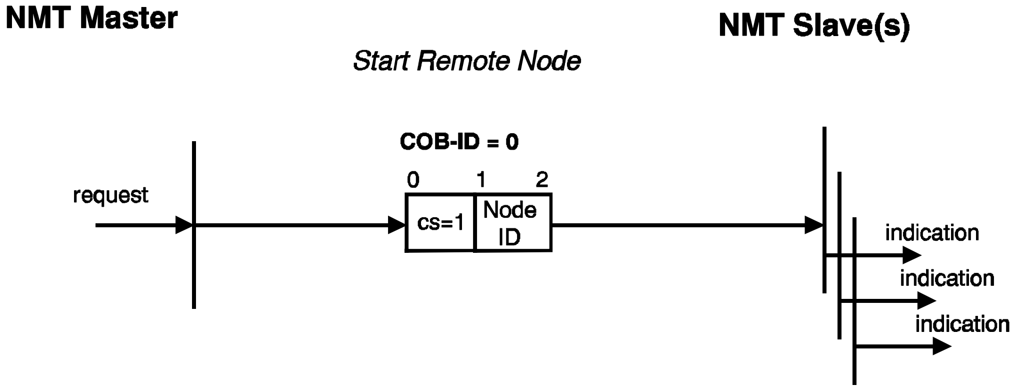
0:	Prepare Protocol completed successfully
1:	DBT protocol error occurred
2:	DBT Master is not available
3..253:	reserved for further use by CiA
254:	request not allowed by the node state of the NMT Slave
255:	other error occurred

- **specific error code:** a value between 0 and 255 inclusive. If the error code equals 1 it gives the error code of the DBT protocol. If error code equals 255, it gives an implementation specific error code. Otherwise it is reserved for further use by CiA.

- **reserved:** reserved for further use by CiA.

4.3 Node Start Protocol

This protocol is used to implement the 'Start Remote Node' service, see /2/.



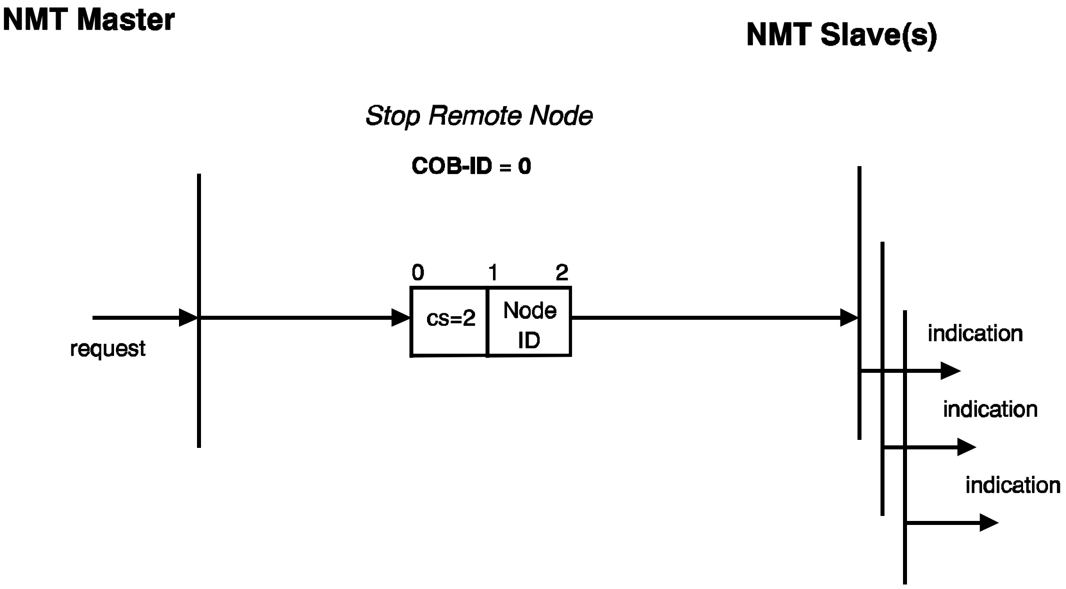
- **cs:** NMT command specifier

1:	start

- **Node-ID:** the Node-ID of the NMT Slave as assigned by the NMT Master in the Node Connect Protocol, or 0. If 0, the protocol addresses all NMT Slaves.

4.4 Node Stop Protocol

This protocol is used to implement the 'Stop Remote Node' service, see /2/.

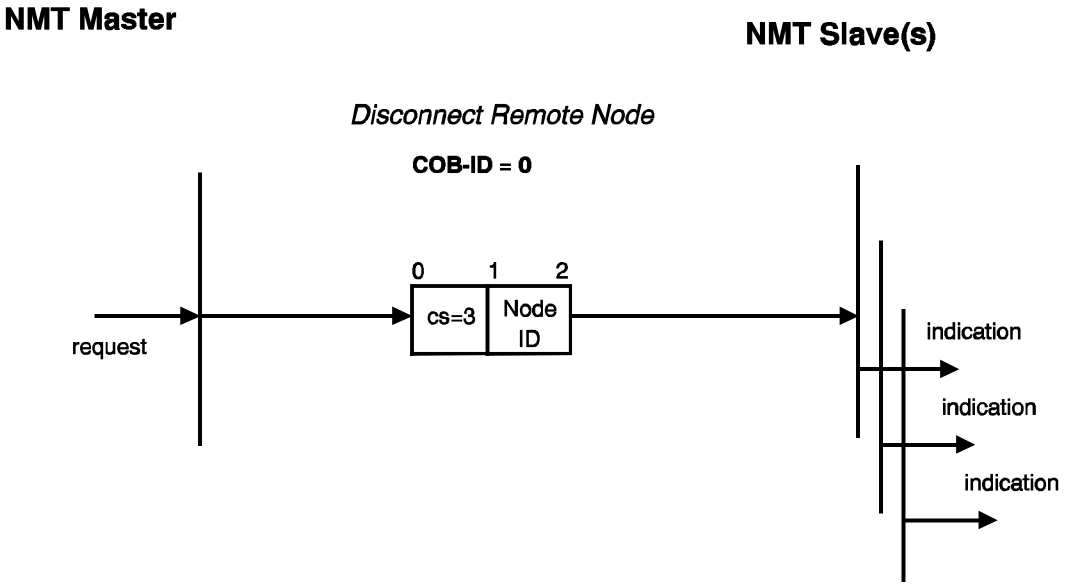


- **cs:** NMT command specifier
  2: stop

- **Node-ID:** the Node-ID of the NMT Slave as assigned by the NMT Master in the Node Connect Protocol, or 0. If 0, the protocol addresses all NMT Slaves.

4.5 Node Disconnect Protocol

This protocol is used to implement the 'Disconnect Remote Node' service, see /2/.

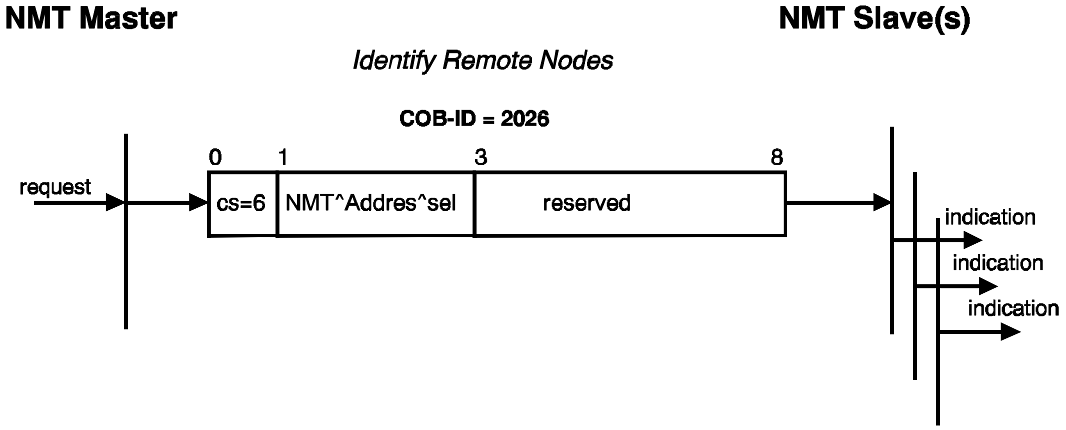


- **cs:** NMT command specifier
  3: disconnect

- Node-ID: the Node-ID of the NMT Slave as assigned by the NMT Master in the Node Connect Protocol, or 0. If 0, the protocol addresses all NMT Slaves.

4.6 Identify Remote Nodes Protocol

This protocol is used to implement the 'Identify Remote Nodes' service, see /2/.



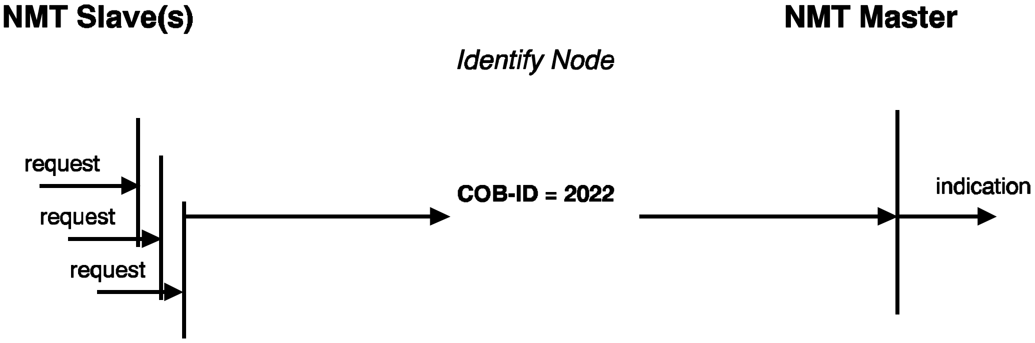
- **cs:** NMT command specifier
  6: identify

- **NMT_Address_sel:** selects a range of module-ID's. Byte 1 contains the lower boundary. Byte 2 contains the upper boundary. The boundaries are included in the range. All NMT Slaves whose NMT Address has a module-ID that lies within this range, are requested to identify themselves, see /2/.

- **reserved:** reserved for further use by CiA.

4.7 Identify Node Protocol

This protocol is used to implement the 'Identify Node' service, see /2/.

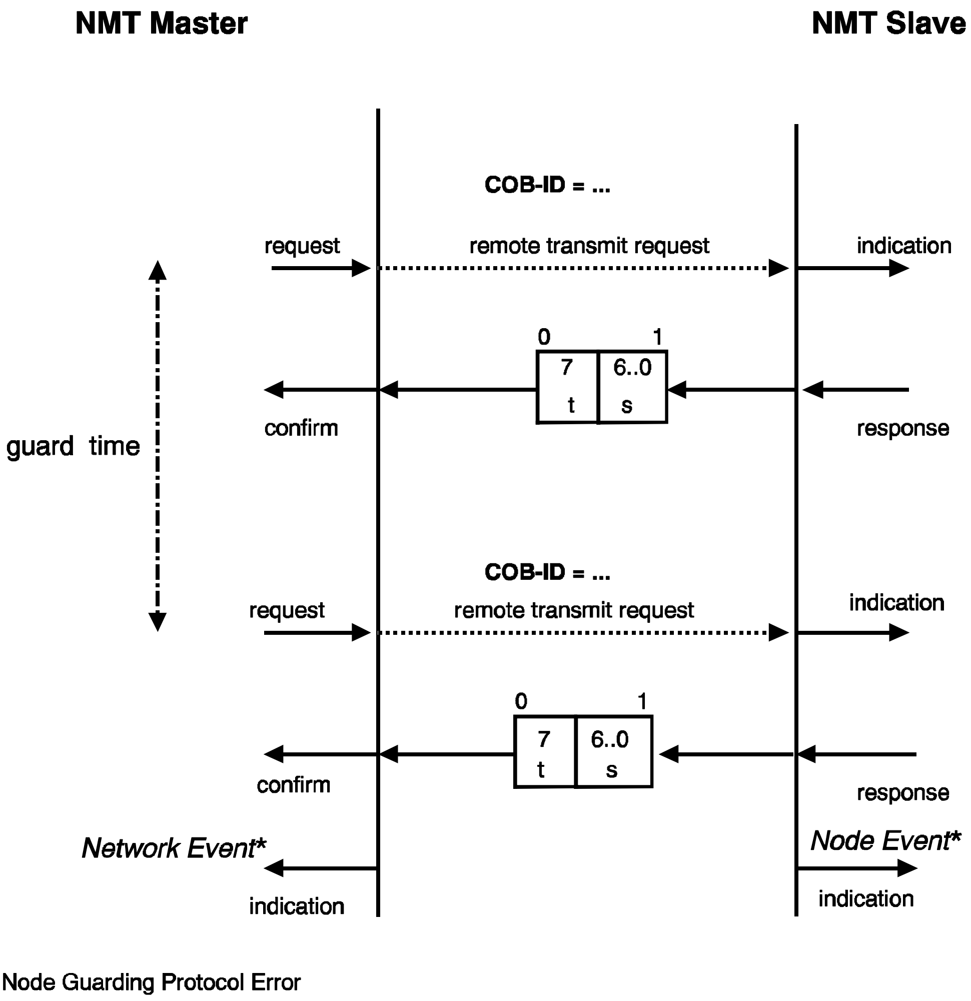


- NOTE: there is no protocol data to allow several NMT Slaves to execute this protocol at the same time. For this service, the NMT Slave takes the initiative.

5. ERROR CONTROL PROTOCOLS

5.1 Node Guarding Protocol

This protocol is used to detect remote errors in the network, see /2/. Each NMT Slave uses one remote COB for the Node Guarding Protocol. This protocol implements the provider initiated Error Control services. The state diagrams in /2/ determine when this protocol is active or inactive.



The NMT Master polls each NMT Slave at regular time intervals. This time-interval is called the guard time and may be different for each NMT Slave. The response of the NMT Slave contains the state of the node object of that NMT Slave, see /2/. If this state does not match the state of the corresponding remote node object or if no response is received by the NMT Master, it will be indicated that a remote error has occurred through the 'Network

Event' service, see /2/. If the NMT Slave hasn't been polled during its life-time, it will be indicated that a remote error has occurred through the 'Node Event' service, see /2/. The life-time can be different for each NMT Slave. If it has been indicated that a remote error has occurred and the errors in the guarding protocol have disappeared, it will be indicated that the remote error has been resolved through the 'Network Event' and 'Node Event' services, see /2/.

The guard time, the life time, and the COB-ID for the Node Guarding Protocol are negotiated between the NMT Master and each NMT Slave in the Node Connect Protocol.

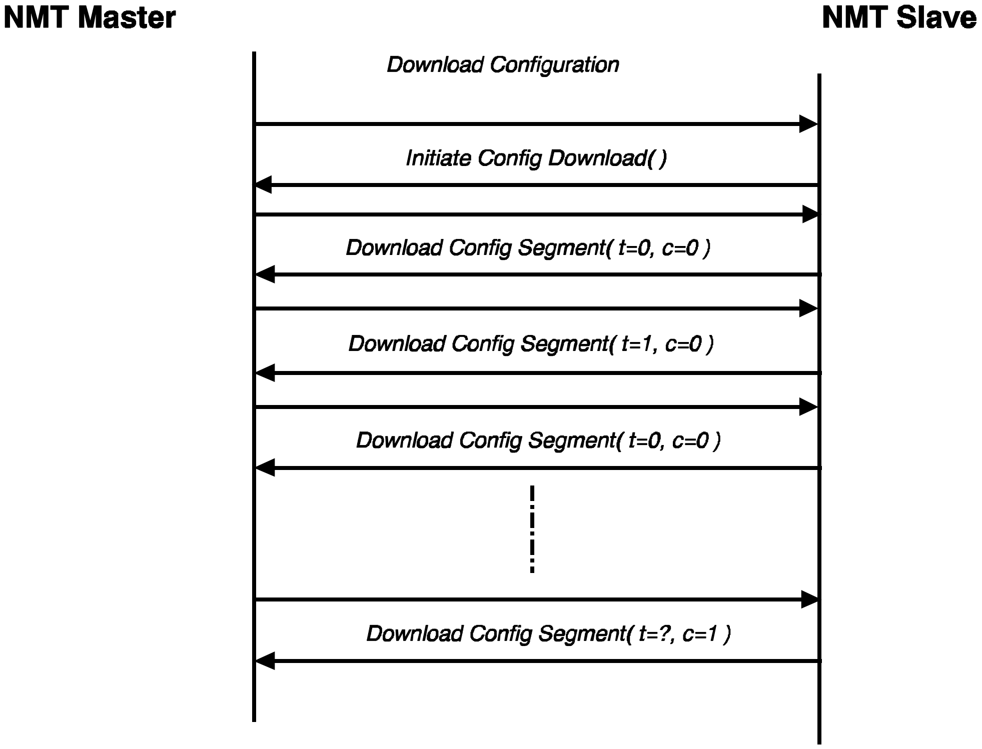
- s: the state of the node object on the NMT Slave

1:	DISCONNECTED
2:	CONNECTING
3:	PREPARING
4:	PREPARED
5:	OPERATIONAL

- t: toggle bit. The value of this bit must alternate between two consecutive responses from the NMT Slave. If it does not alter, it will be indicated that a remote error has occurred through the 'Network Event' service, see /2/. The value of the toggle-bit of the first reponse after the Guarding Protocol becomes active, is 0.

6. CONFIGURATION CONTROL PROTOCOLS

6.1 Download Configuration Protocol


NMT Master
NMT Slave
Download Configuration
Initiate Config Download( )
Download Config Segment(t=0, c=0)
Download Config Segment(t=1, c=0)
Download Config Segment(t=0, c=0)
Download Config Segment(t=?, c=1)

This protocol is used to implement the 'Configuration Download' service, see /2/. Configurations are downloaded as a sequence of 'Download Configuration Segment' services preceded by an 'Initiate Configuration Download' service. The sequence can be terminated by:

- a 'Download Configuration Segment' response/confirm with the c-bit set to 1, confirming the succesful completion of the download sequence.

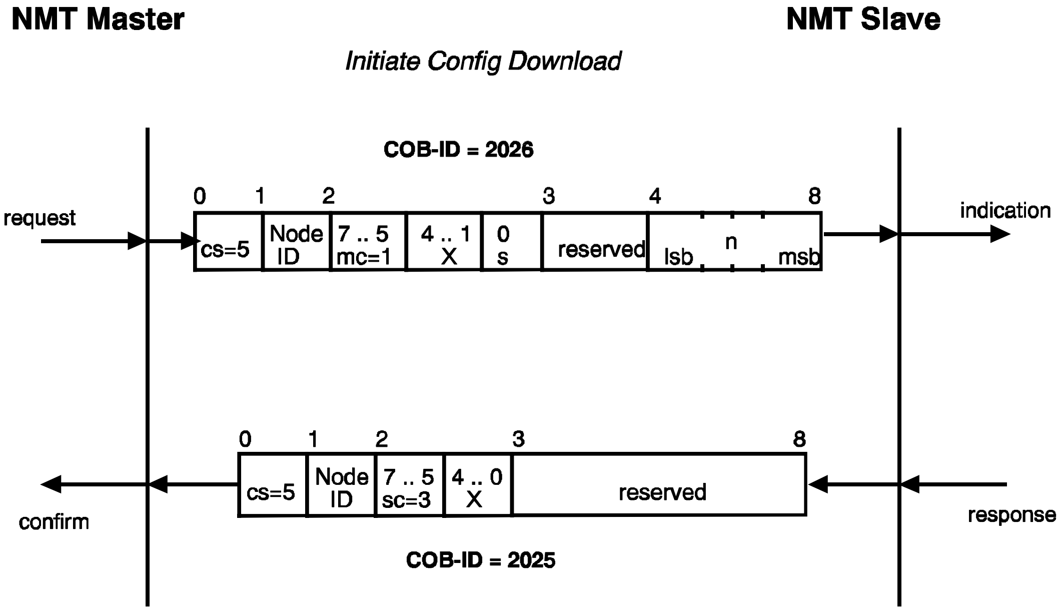
- an 'Abort Configuration Transfer' request/indication indicating the unsuccesful completion of the download sequence.

- a new 'Initiate Configuration Download' request/indication indicating the unsuccesful completion of the download sequence and starting a new sequence.

If in the download of two consecutive segments the toggle bit does not alter, this must be treated as if an invalid COB was received (see Annex I). If such an error is reported to the application, the application may decide to abort the download.

Initiate Configuration Download Protocol

This protocol is used to implement the 'Initiate Configuration Download' service, see /2/.



- **cs:** NMT command specifier
  5: Transfer Configuration

- **Node-ID:** the Node-ID of the NMT Slave as assigned by the NMT Master in the Node Connect Protocol.

- **mc:** NMT Master command
  1: initiate download request

- **sc:** NMT Slave command
  3: initiate download response

- **s:** size indicator
  0: configuration size is not indicated
  1: configuration size is indicated

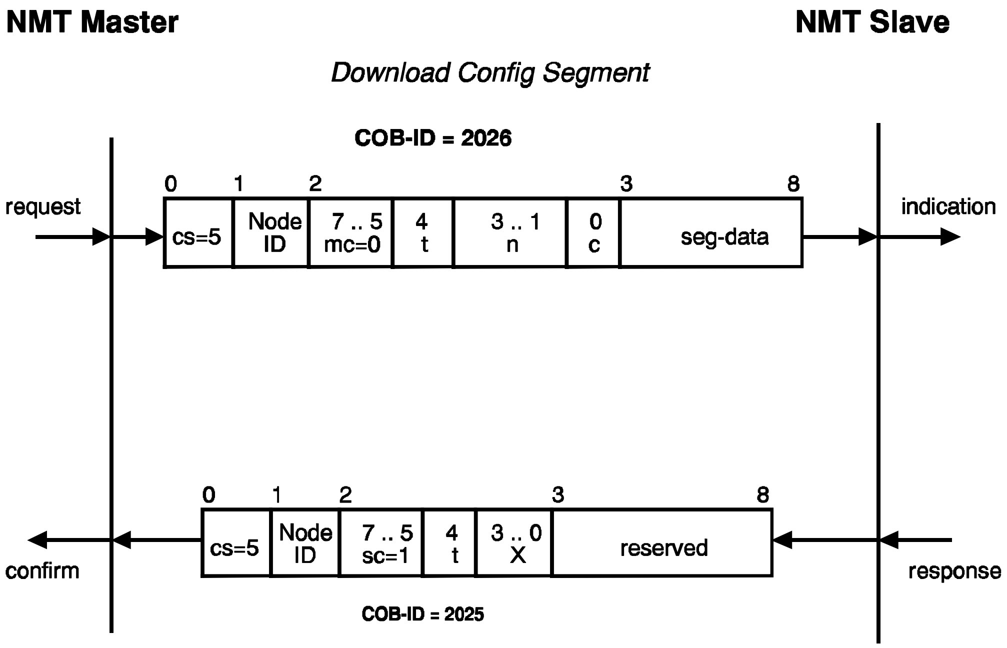
- **n:** only valid if s = 1, otherwise reserved for further use by CiA. If valid it contains the number of bytes to be downloaded

- **X:** not used, always 0

- **reserved:** reserved for further use by CiA.

Download Configuration Segment Protocol

This protocol is used to implement the 'Download Configuration Segment' service, see /2/.



- **cs:** NMT command specifier
  5: Transfer Configuration

- **Node-ID:** the Node-ID of the NMT Slave as assigned by the NMT Master in the Node Connect Protocol.

- **mc:** NMT Master command
  0: download segment request

- **sc:** NMT Slave command
  1: download segment response

- **c:** indicates whether there are still more segments to be downloaded.
  0: more segments to be downloaded
  1: no more segments to be downloaded

- **seg-data:** contains at most five bytes of segment data to be downloaded. Its meaning has to be specified by the application.

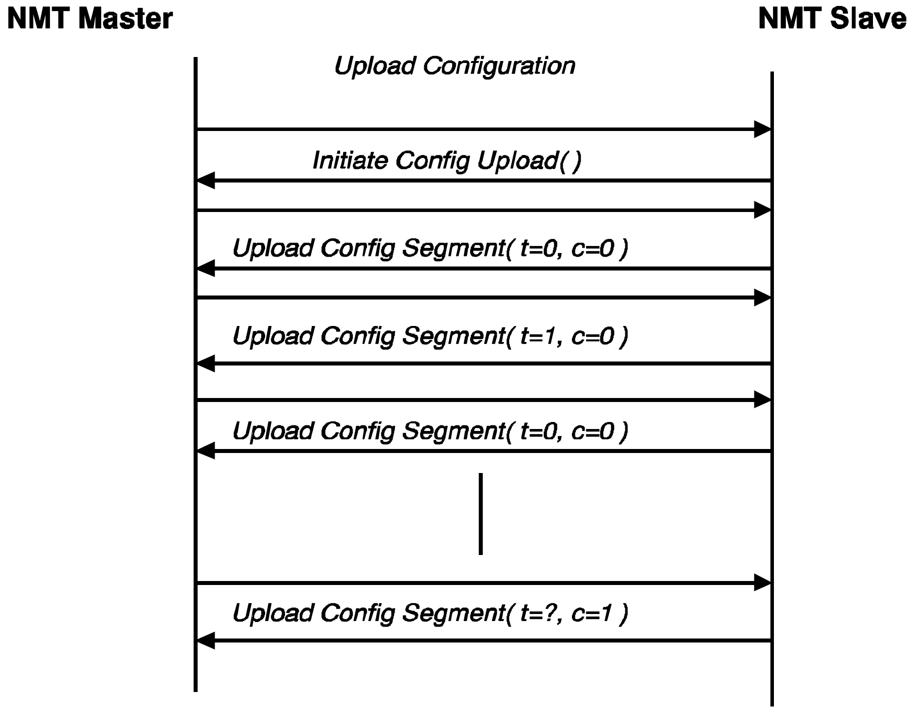
- **n:** contains the number of bytes in seg-data that do not contain segment data. Bytes [8-n, 7] do not contain segment data.

- **t:** toggle bit. This bit must alternate for each subsequent segment that is downloaded. The first segment will have the toggle-bit set to 0. The toggle bit will be equal for the request and the response message.

- **X:** not used, always 0

- **reserved:** reserved for further use by CiA.

6.2 Upload Configuration Protocol


NMT Master
NMT Slave
Upload Configuration
Initiate Config Upload( )
Upload Config Segment(t=0, c=0)
Upload Config Segment(t=1, c=0)
Upload Config Segment(t=0, c=0)
Upload Config Segment(t=?, c=1)

This protocol is used to implement the 'Configuration Upload' service, see /2/. Configurations are uploaded as a sequence of 'Upload Configuration Segment' services preceded by an 'Initiate Configuration Upload' service. The sequence can be terminated by:

- an 'Upload Configuration Segment' response/confirm with the c-bit set to 1, indicating the succesful completion of the upload sequence.

- an 'Abort Configuration Transfer' request/indication indicating the unsuccess ful completion of the upload sequence.

- a new 'Initiate Configuration Upload' request/indication indicating the unsuc cesful completion of the upload sequence and starting a new sequence.

If in the upload of two consecutive segments the toggle bit does not alter, this must be treated as if an invalid COB was received (see Annex I). If such an error is reported to the application, the application may decide to abort the upload.

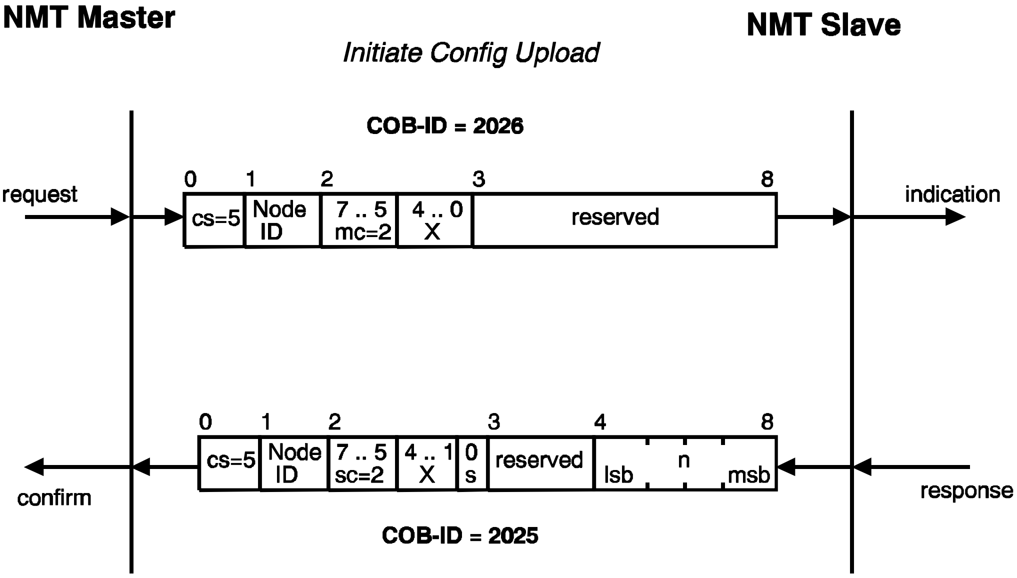
Initiate Configuration Upload Protocol

This protocol is used to implement the 'Initiate Configuration Upload' service, see /2/.

- **cs:** NMT command specifier
  5: Transfer Configuration

- **Node-ID:** the Node-ID of the NMT Slave as assigned by the NMT Master in the Node Connect Protocol.

- **mc:** NMT Master command
  2: initiate upload request



- **sc:** NMT Slave command
  2: initiate upload response

- **s:** size indicator
  0: configuration size is not indicated
  1: configuration size is indicated

- **n:** Only valid if s = 1, otherwise reserved for further use by CiA. If valid it contains the number of bytes to be uploaded

- **X:** not used, always 0

- **reserved:** reserved for further use by CiA.

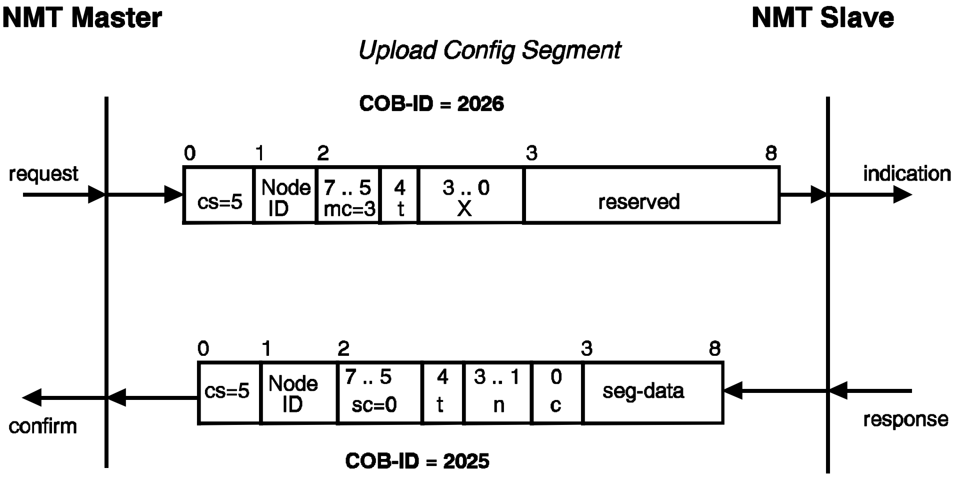
Upload Configuration Segment Protocol

This protocol is used to implement the 'Upload Configuration Segment' service, see /2/.

- **cs:** NMT command specifier
  5: Transfer Configuration

- **Node-ID:** the Node-ID of the NMT Slave as assigned by the NMT Master in the Node Connect Protocol.

- **mc:** NMT Master command
  3: upload segment request



- **sc:** NMT Slave command
  0: upload segment response

- **c:** indicates whether there are still more segments to be uploaded.
  0 : more segments to be uploaded
  1 : no more segments to be uploaded

- **seg-data:** contains at most five bytes of segment data to be uploaded. Its meaning has to be specified by the application.

- **n:** contains the number of bytes in seg-data that do not contain segment data. Bytes [8-n, 7] do not contain segment data.

- **t:** toggle bit. This bit must alternate for each subsequent segment that is uploaded. The first segment will have the toggle-bit set to 0. The toggle bit will be equal for the request and the response message.

- **X:** not used, always 0

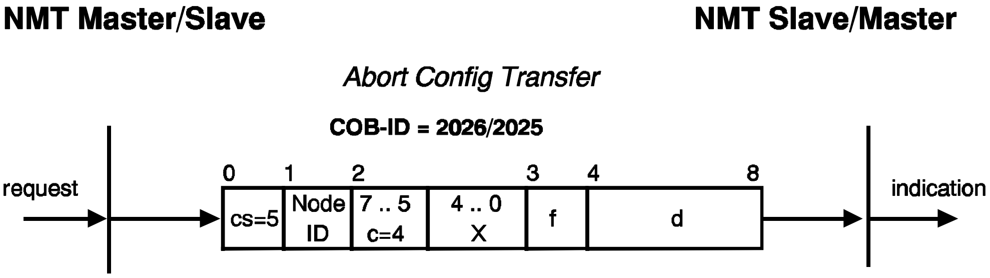
- **reserved:** reserved for further use by CiA.

6.3 Abort Configuration Transfer Protocol

This protocol is used to implement the 'Abort Configuration Transfer' service, see /2/. The NMT Master sends COB-ID 2026. The NMT Slave sends COB-ID 2025.

- **cs:** NMT command specifier
  5: Transfer Configuration

- **Node-ID:** the Node-ID of the NMT Slave as assigned by the NMT Master in the Node Connect Protocol.



- **c:** command
  4: abort configuration transfer request

- **f:** indicates the reason for the failure.

0:	unspecified error
1:	application request
2:	no resources
3..127:	reserved for further use by CiA
128..255:	implementation specific error codes

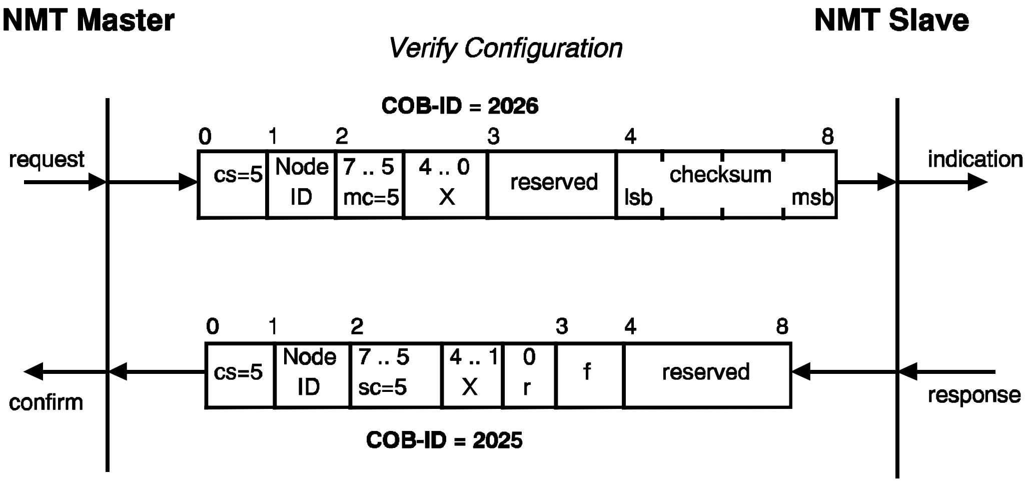
- **d:** only valid if f = 1 or f > 128, otherwise reserved for further use by CiA. If valid it contains application specific information about the reason for the abort.

- **X:** not used, always 0

- **reserved:** reserved for further use by CiA.

6.4 Verify Configuration Protocol

This protocol is used to implement the 'Verify Configuration' service, see /2/.



- **cs:** NMT command specifier
  5: Transfer Configuration

- **Node-ID:** the Node-ID of the NMT Slave as assigned by the NMT Master in the Node Connect Protocol.

- **mc:** NMT Master command
  5: verify configuration

- **sc:** NMT Slave command
  5: verify configuration

- **checksum:** the check-sum to be verified

- **r:** result
  0: verification successful
  1: verification failed

- **f:** Only valid if r = 1. If valid it indicates the reason for the failure.
  0: unspecified error
  1: checksum mismatch
  2: no configuration present
  3..127: reserved for further use by CiA
  128..255: implementation specific error codes

- **X:** not used, always 0

- **reserved:** reserved for further use by CiA.

ANNEX I

IMPLEMENTATION RULES

When implementing the NMT protocols, the following rules have to be followed to guarantee interoperability. These rules deal with the following implementation aspects:

Invalid COB's

A COB is invalid if it has a COB-ID that is used by the NMT Protocol, but it contains invalid parameter values according to the NMT Protocol. This can only be caused by errors in the lower layers (see /1/) or implementation errors. Invalid COB's must be handled locally in an implementation specific way that does not fall within the scope of the CiA Standard on the CAN Application Layer for Industrial Applications. As far as the NMT Protocol is concerned, an invalid COB must be ignored.

Time-out's

Since COB's may be ignored, the response of a confirmed NMT service may never arrive. To resolve this situation, an implementation may, after a certain amount of time, indicate this to the service user (time-out). A time-out is not a confirm of that NMT service. A time-out indicates that the service has not completed yet. The application must deal with this situation. Time-out values are considered to be implementation specific and do not fall within the scope of the CiA Standard on the CAN Application Layer for Industrial Applications. However, it is recommended that an implementation provides facilities to adjust these time-out values to the requirements of the application.