INTERNATIONAL STANDARD

ISO 11898-1

First edition
2003-12-01

Road vehicles — Controller area network (CAN) —

Part 1:
Data link layer and physical signalling

Véhicules routiers — Gestionnaire de réseau de communication (CAN) —

Partie 1: Couche liaison et signalisation physique

Reference number
ISO 11898-1:2003(E)

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 11898-1 was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 3, *Electrical and electronic equipment*.

This first edition of ISO 11898-1, together with ISO 11898-2, replaces ISO 11898:1993, which has been technically revised. Whereas the replaced International Standard covered both the CAN DLL and the high-speed PL, ISO 11898-1 specifies the DLL, including LLC and MAC sublayers, as well as the PLS sublayer, while ISO 11898-2 specifies the high-speed MAU.

ISO 11898 consists of the following parts, under the general title *Road vehicles — Controller area network (CAN)*:

— *Part 1: Data link layer and physical signalling*

— *Part 2: High-speed medium access unit*

— *Part 3: Low-speed, fault-tolerant, medium dependent interface*

— *Part 4: Time-triggered communication*

Road vehicles — Controller area network (CAN) —

Part 1:
Data link layer and physical signalling

1 Scope

This part of ISO 11898 specifies the data link layer (DLL) and physical signalling of the controller area network (CAN): a serial communication protocol that supports distributed real-time control and multiplexing for use within road vehicles. While describing the general architecture of CAN in terms of hierarchical layers according to the ISO reference model for open systems interconnection (OSI) established in ISO/IEC 7498-1, it provides the characteristics for setting up an interchange of digital information between modules implementing the CAN DLL — itself specified according to ISO/IEC 8802-2 and ISO/IEC 8802-3 — with detailed specification of the logical link control (LLC) sublayer and medium access control (MAC) sublayer.

2 Conformance

The conformance of the DLL shall be tested according to ISO 16845.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 7498-1, *Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model*

ISO/IEC 8802-2, *Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part 2: Logical link control*

ISO/IEC 8802-3, *Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*

ISO 16845, *Road vehicles — Controller area network (CAN) — Conformance test plan*

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

4.1
bit rate
number of bits per time during transmission, independent of bit representation

4.2
bit stuffing
filling using bits to provide bus state changes required for periodic resynchronization when using an NRZ bit representation

NOTE Whenever the transmitting logic encounters a certain number (stuff width) of consecutive bits of equal value in the data, it automatically stuffs a bit of complementary value — a stuff bit — into the outgoing bit stream. Receivers destuff the frame, i.e. the inverse procedure is carried out.

4.3
bit time
t_B
duration of one bit

4.4
bus
topology of a communication network, where all nodes are reached by passive links which allow transmission in both directions

4.5
bus comparator
device converting physical signals used for transfer across the communication medium back into logical information or data signals

4.6
bus driver
device converting information or data signals into physical signals so that these signals can be transferred across the communication medium

4.7
bus state
one of two complementary logical states: dominant or recessive

NOTE The dominant state represents the logical 0, and the recessive state represents the logical 1. During simultaneous transmission of dominant and recessive bits, the resulting bus state is dominant. When no transmission is in progress, the bus is idle. During that time it is in the recessive state.

4.8
contention-based arbitration
CSMA arbitration procedure where simultaneous access of multiple nodes results in a contention

4.9
frame
data link PDU specifying the arrangement and meaning of bits or bit fields in the sequence of transfer across the transmission medium

4.10
multicast
addressing where a single frame is addressed to a group of nodes simultaneously

NOTE *Broadcast* is a special case of multicast, whereby a single frame is addressed to all nodes simultaneously.

4.11
multimaster
system partitioned into several nodes where every node may temporarily control the action of other nodes

4.12
node
assembly, linked to a communication network, capable of communicating across the network according to a communication protocol specification

NOTE A CAN node is a node communicating across a CAN network.

4.13
non-return-to-zero
NRZ
method of representing binary signals, i.e. within one and the same bit time the signal level does not change, where a stream of bits having the same logical value provides no edges

4.14
priority
attribute to a frame controlling its ranking during arbitration, a high priority increases the probability that a frame wins the arbitration process

4.15
protocol
formal set of conventions or rules for the exchange of information between nodes, including the specification of frame administration, frame transfer and PL

4.16
receiver
node when if it is not a transmitter and the bus is not idle

4.17
time-triggered communication
option where a frame can be transmitted at a specific time slot, also providing a global synchronization of clocks and allowing the disabling of the automatic retransmission of frames

4.18
transmitter
node originating a data frame or remote frame, which stays transmitter until the bus is idle again or until the node loses arbitration

5 Symbols and abbreviated terms

ACK	acknowledgement
BCH	Bose-Chaudhuri-Hocquenghem
BR	bit rate
t_B	bit time
CAN	controller area network
CRC	cyclic redundancy check
CSMA	carrier sense multiple access
DLC	data length code
DLL	data link layer

EOF	end of frame
FCE	fault confinement entity
IC	integrated circuit
IDE	identifier extension flag
LAN	local area network
LLC	logical link control
LME	layer management entity
LPDU	LLC protocol data unit
LSB	least significant bit
LSDU	LLC service data unit
MA	medium access
MAC	medium access control
MAU	medium access unit
MDI	medium dependent interface
MPDU	MAC protocol data unit
MSB	most significant bit
MSDU	MAC service data unit
NRZ	non-return-to-zero
OSI	open system interconnection
OVLD	overload
PCI	protocol control information
PDU	protocol data unit
PL	physical layer
PLS	physical signalling
PMA	physical medium attachment
REC	receive error counter
RTR	remote transmission request
SDU	service data unit
SJW	synchronization jump width
SOF	start of frame
SRR	substitute remote request
TEC	transmit error counter
TTC	time triggered communication

6 Basic concepts of CAN

6.1 CAN properties

CAN shall have the following properties:

— multi-master priority-based bus access;

— non-destructive contention-based arbitration;

— multicast frame transfer by acceptance filtering;

— remote data request;

— configuration flexibility;

— system-wide data consistency;

— error detection and error signalling;

— automatic retransmission of frames that have lost arbitration or have been destroyed by errors during transmission;

— distinction between temporary errors and permanent failures of nodes and autonomous switching-off of defective nodes.

6.2 Frames

Information on the bus shall be sent in fixed format frames of different but limited length. When the bus is idle, any connected node may start to transmit a new frame.

6.3 Bus access method

When the bus is idle, any node may start to transmit a frame. If two or more nodes start to transmit frames at the same time, the bus access conflict shall be resolved by contention-based arbitration using the identifier. The mechanism of arbitration shall ensure that neither information nor time is lost. The transmitter with the frame of highest priority shall gain the bus access.

6.4 Information routing

In CAN systems a node shall not make use of any information about the system configuration (e.g. node address). Instead, receivers accept or do not accept information based upon a process called frame acceptance filtering, which decides whether the received information is relevant or not. There is no need for receivers to know the transmitter of the information and vice versa.

6.5 System flexibility

Nodes may be added to the CAN network without requiring any change in the software or hardware of any node, if the added node is not the transmitter of any data frame and if the added node does not require any additional transmitted data.

6.6 Data consistency

Within CAN a frame shall simultaneously be accepted either by all nodes or by none. Thus data consistency shall be a property of the system achieved by the concepts of multicast and by error handling.

6.7 Remote data request

By sending a remote frame, a node requiring data may request another node to send the corresponding data frame. The data frame and the corresponding remote frame shall be named by the same identifier.

6.8 Error detection

For detecting errors, the following measures shall be provided:

— monitoring (transmitters compare the bit levels to be transmitted with the bit levels detected on the bus);

— 15-bit CRC;

— variable bit stuffing with a stuff width of 5;

— frame check;

— acknowledge check.

6.9 Error signalling and recovery time

Corrupted frames shall be flagged by any transmitting node and any normally operating (error-active) receiving node. Such frames shall be aborted and retransmitted according to the implemented recovery procedure (see 8.3.4). The recovery time from detecting an error until the possible start of the next frame shall be typically seventeen (17) to twenty-three (23) bit times [in the case of a heavily disturbed bus, up to thirty-one (31) bit times], if there are no further errors.

6.10 ACK

All receivers shall check the consistency of the received frame and shall acknowledge a consistent frame and shall flag an inconsistent frame. A frame that is not acknowledged is corrupted and shall be flagged by the transmitting node.

6.11 Automatic retransmission

Frames that have lost arbitration and frames that have been disturbed by errors during transmission shall be retransmitted automatically when the bus is idle again. A frame that will be retransmitted shall be handled as any other frame, i.e. it participates in the arbitration process to gain bus access. In case of TTC, the automatic retransmission shall be disabled (see 9.2.5).

6.12 Fault confinement

CAN nodes shall be able to distinguish short disturbances from permanent failures. Defective transmitting nodes shall be switched off. *Switched off* means a node is logically disconnected from the bus, so that it can neither send nor receive any frames (see 13.1.4.3).

6.13 Error-active

An error-active node shall normally take part in bus communication and send an active error flag when an error has been detected. The active error flag shall consist of six (6) consecutive dominant bits and shall violate the rule of bit stuffing and all fixed formats appearing in a regular frame (see 13.1.4.2).

6.14 Error-passive

An error-passive node shall not send an active error flag. It takes part in bus communication, but when an error has been detected a passive error flag shall be sent. The passive error flag shall consist of six (6) consecutive recessive bits. After transmission, an error-passive node shall wait some additional time before initiating a further transmission (see suspend transmission in 10.4.6.4, and 13.1.4.2).

6.15 Bus-off

A node shall be in the bus-off state when it is switched off from the bus due to a request of FCE. In the bus-off state, a node shall neither send nor receive any frames. A node shall start the recovery from bus-off state only upon a user request.

7 Layered architecture of CAN

7.1 Reference to OSI model

According to the OSI reference model, the CAN architecture of this part of ISO 11898 shall represent the two layers,

— DLL, and

— PL.

See Figure 1.

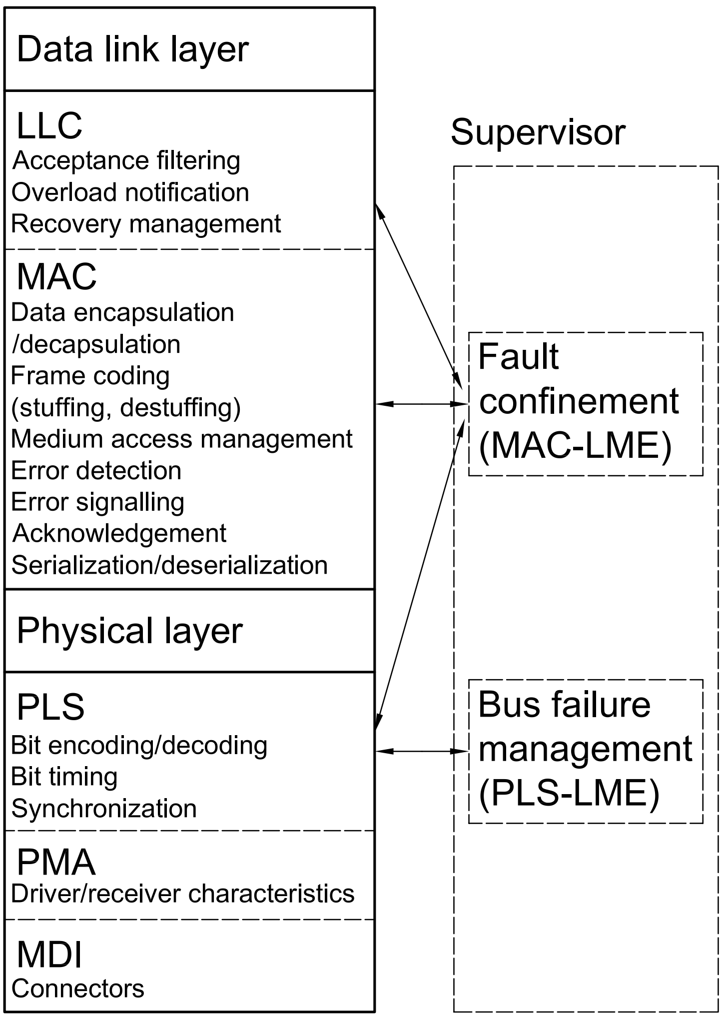

Figure 1 — Layered architecture of CAN

According to ISO/IEC 8802-2 and ISO/IEC 8802-3, the DLL has been subdivided into

— LLC, and

— MAC.

The PL has been subdivided into

— PLS,

— PMA, and

— MDI.

The MAC sublayer operations shall be supervised by an FCE. Fault confinement shall be a self-checking mechanism that distinguishes short disturbances from permanent failures (for fault confinement, see 13.1).

The PL may be supervised by an entity that detects and manages failures of the physical medium.

7.2 Protocol specification

Two peer protocol entities shall communicate with each other by exchanging frames or PDUs.

An NPDU shall consist of N-PCI and (N)-user data. NPDU shall be passed to a ($N-1$)-layer entity via a ($N-1$)-SAP. The NPDU shall be passed by means of the ($N-1$)-SDU to the ($N-1$)-layer, the services of which allow the transfer of the NPDU. The SDU shall be the interface data whose identity is preserved between (N)-layer entities, i.e. it represents the logical data unit transferred by a service. The DLL of the CAN protocol shall not provide either the means for mapping one SDU into multiple PDUs or for mapping multiple SDUs into one PDU, i.e. an NPDU is directly constructed from the associated NSDU and the layer-specific control information N-PCI. Figure 2 illustrates the data link sublayer interactions.

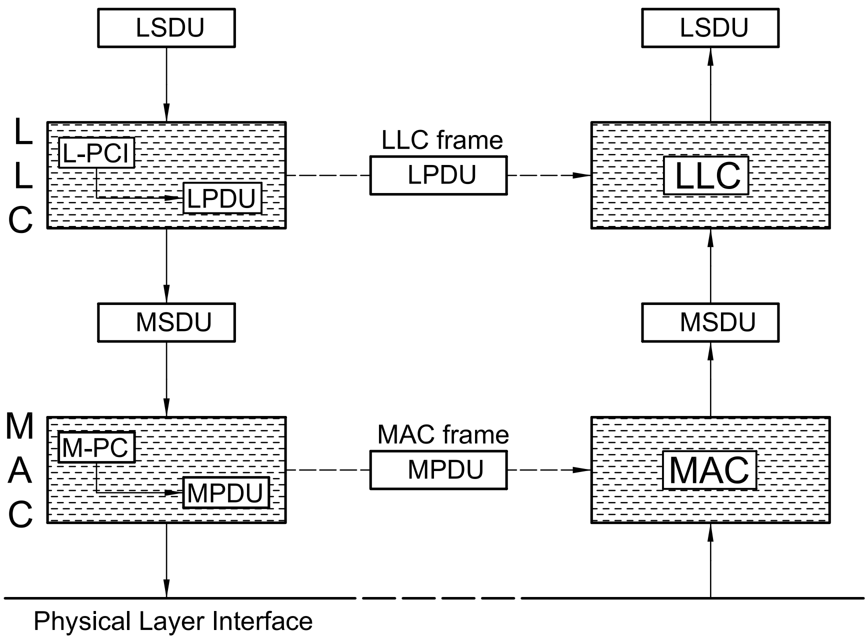

Figure 2 — Protocol layer interactions

7.3 Format description of services

7.3.1 Format description of service primitives

Service primitives shall be written as

service.type (

[parameter1,...]

)

where *service* indicates the name of the service, e.g. L_Data for data transfer service provided by the LLC sublayer, *type* indicates the type of the service primitives (see 7.3.2), and [parameter1,...] is the list of values passed to the service primitives. The square brackets indicate that this parameter list may be empty.

7.3.2 Types of service primitives

Service primitives shall be of three generic types:

a) Service.Request

The request primitive shall be passed from the (N)-user (service user) to the (N)-layer (service provider) to request initiation of the service.

b) Service.Indication

The indication primitive shall be passed from the (N)-layer to the (N)-user to indicate an internal (N)-layer (or sublayer) event which is significant to the (N)-user. This event may be logically related to a remote service request, or may be caused by an event internal to the (N)-layer (or sublayer).

c) Service.Confirm

The confirm primitive shall be passed from the (N)-layer (or sublayer) to the (N)-user to convey the results of one or more associated previous service request(s). This primitive may indicate either failure to comply or some level of compliance. It shall not necessarily indicate any activity at the remote peer interface.

7.4 LLC interface

The LLC sublayer shall offer two types of connectionless transmission services to the LLC user:

a) unacknowledged data transfer service;

b) unacknowledged remote data request service.

The interface service data sent from or to the user shall be as in 8.2.2. The messages sent between LLC user and LLC sublayer shall be as shown in a) and b) of Table 1.

The LLC interface messages sent from and to the supervisor FCE shall be as in 13.1.3.

Table 1 — Messages between LLC user and LLC sublayer

a) Message sent from LLC user to LLC sublayer	
User to LLC message	**Meaning**
Reset_Request	Request to set the node into an initial state
b) Messages sent from LLC sublayer to LLC user	
LLC to user message	**Meaning**
Reset_Response	Response to the Reset_Request
Node_Status	Indicates the current status of the node, i.e. it signals whether or not the node is in the bus-off state.

8 Description of LLC sublayer

8.1 General

The LLC sublayer describes the upper part of the OSI DLL. It is related with those protocol issues that are independent of the type of medium access method.

8.2 Services of LLC sublayer

8.2.1 Types of connectionless-mode transmission services

The LLC sublayer shall offer two types of connectionless-mode transmission services:

a) Unacknowledged data transfer service

This service shall provide means by which LLC users exchange LSDU without establishing a data-link connection. The data transfer may be point-to-point, multicast or broadcast.

b) Unacknowledged remote data request service

This service shall provide means used by an LLC user to request a remote node for an LSDU transmission without establishing a data-link connection.

The remote node shall basically serve the data request in the following two ways:

1) The requested data may be prepared by the remote user for transmission. In this case the data shall be located in a remote node buffer and shall be transmitted by the remote user LLC entity upon reception of the remote request frame.

2) The requested data shall be transmitted by the remote user upon reception of the remote request frame.

According to the two different LLC services, two types of frames sent from or to the user shall be used:

— LLC data frame;

— LLC remote frame.

The LLC data frame shall carry data from a transmitter to a receiver. The LLC remote frame shall be transmitted to request transmission of a data frame (with the same identifier) from a (single) remote node. In both cases, the LLC sublayer shall notify the successful transmission or reception of a frame to the user.

8.2.2 Service primitive specification

8.2.2.1 General

The service primitive specification of this subclause describes in detail the LLC service primitives and their associated parameters. The complete list of LLC service primitives shall be as given in Table 2.

Table 2 — LLC service primitives overview

Unacknowledged data transfer service	
L_Data.Request	Request for data transfer
L_Data.Indication	Indication of data transfer
L_Data.Confirm	Confirm data transfer
Unacknowledged remote data request service	
L_Remote.Request	Request for remote data request
L_Remote.Indication	Indication of remote data request
L_Remote.Confirm	Confirmation remote data request

The parameters associated with the different LLC service primitives shall be as given in Table 3.

Table 3 — List of LLC service primitive parameters

LLC service primitive parameters	
Identifier	Identifies the data and its priority
DLC	DLC
Data	Data the user wants to transmit
Transfer_Status	Confirmation parameter

8.2.2.2 L_Data.Request

8.2.2.2.1 Function

The L_Data.Request primitive shall be passed from the LLC user to the LLC sublayer to request that an LSDU be sent to one or more remote LLC entities.

8.2.2.2.2 Semantics of L_Data.Request primitive

The primitive shall provide parameters as follows

```
L_Data.Request (
                Identifier
                DLC
                Data
                )
```

The parameter Data shall be insignificant if the associated LLC data frame is of data length zero.

8.2.2.2.3 Effect on receipt

Receipt of this primitive shall cause the LLC sublayer to initiate the transfer of an LLC data frame by use of the data transfer service provided by the MAC sublayer (see Table 5).

8.2.2.3 L_Data.Indication

8.2.2.3.1 Function

The L_Data.Indication primitive shall be passed from the LLC sublayer to the LLC user to indicate the arrival of an LSDU.

8.2.2.3.2 Semantics of L_Data.Indication primitive

The primitive shall provide parameters as follows

```
L_Data.Indication (
                  Identifier
                  DLC
                  Data
                  )
```

The parameter Data shall be insignificant if the associated LLC data frame is of data length zero.

8.2.2.3.3 Effect on receipt

The effect on receipt of this primitive by the LLC user is unspecified.

8.2.2.4 L_Data.Confirm

8.2.2.4.1 Function

The L_Data.Confirm primitive shall be passed from the local LLC sublayer to the LLC user to convey the results of the previous L_Data.Request primitive. This primitive shall be a local confirmation, i.e. it shall not imply that the remote LLC entity or entities have passed the associated indication primitive to the corresponding LLC user(s).

8.2.2.4.2 Semantics of L_Data.Confirm primitive

The primitive shall provide parameters as follows:

```
L_Data.Confirm (
               Identifier
               Transfer_Status
               )
```

The Transfer_Status shall be used to indicate the completion of the transaction initiated by the previous L_Data.Request primitive.

Transfer_Status:[Complete, Not_Complete]

8.2.2.4.3 Effect on receipt

The effect on receipt of this primitive by the LLC user is unspecified.

8.2.2.5 L_Remote.Request

8.2.2.5.1 Function

The L_Remote.Request primitive shall be passed from the LLC user to the LLC sublayer to request a single remote LLC entity to transmit an LSDU.

8.2.2.5.2 Semantics of L_Remote.Request primitive

The primitive shall provide parameters as follows:

```
L_Remote.Request (
                 Identifier
                 DLC
                 )
```

The value of DLC equals the length of the data field of the requested data frame.

8.2.2.5.3 Effect on receipt

Receipt of this primitive shall cause the LLC sublayer to initiate the transfer of an LSDU by use of the remote data transfer service provided by the MAC sublayer (see Table 5).

8.2.2.6 L_Remote.Indication

8.2.2.6.1 Function

The L_Remote.Indication primitive shall be passed from the LLC sublayer to the LLC user to indicate the arrival of a request for transmission of an LSDU.

8.2.2.6.2 Semantics of L_Remote.Indication primitive

The primitive shall provide parameters as follows:

```
L_Remote.Indication (
                     Identifier
                     DLC
                     )
```

The identifier shall identify the LSDU to be sent. The value of DLC equals the length of the data field of the requested data frame.

8.2.2.6.3 Effect on receipt

The effect on receipt of this primitive by the LLC user is unspecified.

8.2.2.7 L_Remote.Confirm

8.2.2.7.1 Function

The L_Remote.Confirm primitive shall be passed from the local LLC sublayer to the LLC user to convey the results of the previous L_Remote.Request primitive. This primitive shall be a local confirmation, i.e. it does not imply that the remote LLC entity has passed the associated indication primitive to the corresponding LLC user.

8.2.2.7.2 Semantics of L_Remote.Confirm primitive

The primitive shall provide parameters as follows:

```
L_Remote.Confirm (
                  Identifier
                  Transfer_Status
                  )
```

The Transfer_Status shall be used to indicate the completion of the transaction initiated by the previous L_Remote.Request primitive.

Transfer_Status:[Complete, Not_Complete]

8.2.2.7.3 Effect on receipt

The effect on receipt of this primitive by the LLC user is unspecified.

8.3 Functions of LLC sublayer

8.3.1 General

The LLC sublayer shall provide the following functions:

a) frame acceptance filtering;

b) overload notification;

c) recovery management.

8.3.2 Frame acceptance filtering

A frame transaction initiated at the LLC sublayer shall be a single, self-contained operation independent of previous frame transactions. The content of a frame shall be named by its identifier. The identifier does not indicate the destination of the frame but describes the meaning of the data. Each receiver may decide by frame acceptance filtering whether the frame is relevant or not.

8.3.3 Overload notification

The transmission of a MAC overload frame shall be initiated by the LLC sublayer if internal conditions of a receiver require delay of the next LLC data or LLC remote frame.

At most two MAC overload frames may be generated to delay the next data frame or remote frame.

8.3.4 Recovery management

The LLC sublayer shall provide the means for automatic retransmission of frames that have lost arbitration or have been disturbed by errors during transmission (see 6.11). The frame transmission service shall not be confirmed to the user before the transmission has been successfully completed.

8.4 Structure of LLC frames

8.4.1 General

LLC frames shall be the data units exchanged between peer LLC entities (LPDU). The structure and format of the LLC data and remote frame shall be specified subsequently.

8.4.2 Specification of LLC data frame

8.4.2.1 General

An LLC data frame shall consist of three bit fields (see Figure 3):

— identifier field;

— DLC field;

— LLC data field.

Identifier field	DLC field	LLC data field

Figure 3 — LLC data frame

8.4.2.2 Identifier field

The identifier field shall be composed of three segments: base identifier, extension flag and identifier extension. The length of the base identifier shall be eleven (11) bits (ID-28 to ID-18), the extension flag one bit, and the length of the identifier extension shall be eighteen (18) bits (ID-17 to ID-0). The identifier extension shall be ignored if the extension flag is logic zero (0).

8.4.2.3 DLC field

The number of bytes in the data field shall be indicated by the DLC (see Table 4). This DLC shall consist of four (4) bits. The admissible number of data bytes for a data frame shall range from zero (0) to eight (8). DLCs in the range of zero (0) to seven (7) shall indicate data fields of length of zero (0) to seven (7) bytes. All other DLCs shall indicate that the data field is eight (8) bytes long.

8.4.2.4 Data field

The data field shall consist of the data to be transferred within a data frame. It may contain from zero (0) bytes to eight (8) bytes, where each byte contains eight (8) bits.

8.4.3 Specification of LLC remote frame

An LLC remote frame shall be composed of two bit fields (see Figure 4):

- identifier field;
- DLC field.

Table 4 — Coding of the numbers of data bytes by the DLC

Number of data bytes	DLC			
	DLC3	DLC2	DLC1	DLC0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0 or 1	0 or 1	0 or 1

Identifier field	DLC field

Figure 4 — LLC data frame

The formats of both the LLC remote frame identifier field and DLC field shall be identical to the formats of the LLC data frame identifier field (see 8.4.2.2) and DLC field (see 8.4.2.3). There shall be no data field, independent of the value of the DLC.

Remote frames may only be transmitted with a system-wide determined DLC, which is the DLC of the corresponding data frame (see 10.8.8).

8.5 Limited LLC frames

The full range of possible identifiers or DLCs is not required to be implemented.

If an LLC sublayer is restricted to the use of a subrange of identifiers (e.g. only base identifiers), then it shall be limited to LLC data frames and LLC remote frames with identifiers of that subrange (e.g. identifiers with their extension flag set to logic 0 and their identifier extension ignored).

If an LLC sublayer is restricted to the use of less than eight (8) data bytes, then it shall be limited to LLC data frames and LLC remote frames with DLCs of that restricted range.

9 Interface between LLC and MAC

9.1 Services

The MAC sublayer shall provide services to the local LLC for

— (MAC-) acknowledged transfer of LLC frames, and

— transmission of MAC overload frames.

The interface service data from or to the LLC sublayer shall be in accordance with 8.3.

9.2 TTC option

9.2.1 Description

The TTC option describes the prerequisites for the synchronization of all nodes in the network. With the synchronization of node, any message may be transmitted at a specific time slot, where it has not to compete for the bus with other messages thus providing predictable latency times by avoiding the loss of arbitration. In order to synchronize the activities of the nodes within the network, a common reference point is needed. The SOF bit or the sample point of the last bit of EOF of any message shall be used as the reference point. The individual presence of a single message at a time shall be referred to as frame scheduling. Based on the synchronization of the nodes, the TTC also facilitates the establishment of a global time system in higher-layer protocols.

The hardware needed to establish TTC shall be included between LLC and MAC.

9.2.2 Time base

Any node that supports TTC option shall provide a time base. The time base is a cyclic up counter of at least 16 bits with either an internal or an external clock.

9.2.3 Time reference point

Any message received or transmitted shall invoke a capture of the time base taken at the SOF recognition of the respective message or at the sample point of the last bit of EOF. After successful message reception, the capture value shall be provided to the CPU for at least one message and shall be readable until the next message is received.

9.2.4 Event generation

It shall be possible to generate at least one programmable event trigger from the above-mentioned time base. The trigger shall be freely programmable by the CPU in the range of at least zero (0) to $(2^{16} - 1) \times$ timer clocks.

9.2.5 Retransmission of frames

Disabling of the automatic retransmission shall be supported (see 6.11).

10 Description of MAC sublayer

10.1 General

The MAC sublayer represents the lower part of the OSI DLL. It shall service the interface to the LLC sublayer and the PL, and comprise the functions and rules related to

— encapsulation/decapsulation of the transmit/receive data,

— error detection and signalling, and

— management of the transmit/receive medium access.

10.2 Services of MAC sublayer

10.2.1 Service description

The services provided by the MAC sublayer shall allow the local LLC sublayer entity to exchange MSDU with the peer LLC sublayer entities. The MAC sublayer services shall be the following.

a) **Acknowledged data transfer**

This service shall provide means by which LLC entities exchange MSDUs without the establishment of a data link connection. The data transfer may be point-to-point, multicast or broadcast.

b) **Acknowledged remote data request**

This service shall provide the means by which an LLC entity can request another remote node to transmit an LSDU without the establishment of a data-link connection. The remote LLC entity shall use the MAC service “acknowledged data transfer” for the transmission of the requested data. The ACK of a service shall be generated by the MAC sublayer(s) of the remote node(s). The ACK shall not contain any data of the remote node user.

c) **Overload frame transfer**

This service shall provide means by which an LLC entity initiates the transmission of an overload frame, a special fixed format LPDU, causing the delay of the next data frame or remote frame.

10.2.2 Service primitives specification

10.2.2.1 General

The service primitives of the MAC sublayer provided to the LLC sublayer shall be as given in Table 5.

Table 5 — MAC sublayer service primitives

Acknowledged data transfer		
MA_Data.request	MA_Data.indication	MA_Data.confirm
Acknowledged remote data request		
MA_Remote.request	MA_Remote.indication	MA_Remote.confirm
Overload frame transfer		
MA_OVLD.request	MA_OVLD.indication	MA_OVLD.confirm

10.2.2.2 MA_Data.Request

10.2.2.2.1 Function

The MA_Data.Request primitive shall be passed from the LLC sublayer to the MAC sublayer to request that an MSDU be sent to one or more remote MAC entities.

10.2.2.2.2 Semantics of MA_Data.Request primitive

The primitive shall provide parameters as follows:

```
MA_Data.Request (
                Identifier
                DLC
                Data
                )
```

The parameter Data is insignificant for MAC data frames of data length zero.

10.2.2.2.3 Effect on receipt

Receipt of this primitive shall cause the MAC sublayer to prepare a PDU by adding all MAC-specific control information (SOF, SRR bit, IDE bit, RTR bit, reserved bits, CRC, recessive bit during ACK-Slot, EOF) to the MSDU coming from the LLC sublayer. The MAC PDU shall be serialized and passed bit by bit as an SDU to the PL for transfer to the peer MAC sublayer entity or entities.

10.2.2.3 MA_Data.Indication

10.2.2.3.1 Function

The MA_Data.Indication primitive shall be passed from the MAC sublayer to the LLC sublayer to indicate the arrival of an MSDU.

10.2.2.3.2 Semantics of MA_Data.Indication primitive

The primitive shall provide parameters as follows:

```
MA_Data.Indication (
                   Identifier
                   DLC
                   Data
                   )
```

The parameter Data is insignificant if the associated MAC data frame is of data length zero. The arrival of an MSDU shall be indicated to the LLC sublayer only if it has been received correctly.

10.2.2.3.3 Effect on receipt

The effect on receipt of this primitive by the LLC sublayer is unspecified.

10.2.2.4 MA_Data.Confirm

10.2.2.4.1 Function

The MA_Data.Confirm primitive shall be passed from the local MAC sublayer to the LLC sublayer to convey the results of the previous MA_Data.Request primitive. This primitive is a remote confirmation, i.e. it shall indicate that the remote MAC entity or entities have passed the associated indication primitive to the corresponding user(s).

10.2.2.4.2 Semantics of MA_Data.Confirm primitive

The primitive shall provide parameters as follows:

```
MA_Data.Confirm (
                Identifier
                Transmission_Status
                )
```

The Transmission_Status shall be used to indicate the success or failure of the previous MA_Data.Request primitive.

Transmission_Status:[Success, No_Success]

Failures are either errors which occurred during transmission or loss of the arbitration.

10.2.2.4.3 Effect on receipt

The effect on receipt of this primitive by the LLC sublayer is unspecified.

10.2.2.5 MA_Remote.Request

10.2.2.5.1 Function

The MA_Remote.Request primitive shall be passed from the LLC sublayer to the MAC sublayer to request a single remote MAC entity to transmit an MSDU.

10.2.2.5.2 Semantics of MA_Remote.Request primitive

The primitive shall provide parameters as follows:

```
MA_Remote.Request (
                  Identifier
                  DLC
                  )
```

The identifier shall identify the MSDU to be sent. The value of DLC shall be equal to the length of the requested MSDU data.

10.2.2.5.3 Effect on receipt

Receipt of this primitive shall cause the MAC sublayer to prepare a PDU by adding all MAC-specific control information (SOF, SRR bit, IDE bit, RTR bit, reserved bits, CRC, recessive bit during ACK-Slot, EOF) to the MSDU coming from the LLC sublayer. The MAC PDU shall be serialized and passed bit by bit as an SDU to the PL for transfer to the peer MAC sublayer entity or entities.

10.2.2.6 MA_Remote.Indication

10.2.2.6.1 Function

The MA_Remote.Indication primitive shall be passed from the MAC sublayer to the LLC sublayer to indicate the arrival of a request for transmission of an MSDU.

10.2.2.6.2 Semantics of MA_Remote.Indication primitive

The primitive shall provide parameters as follows:

```
MA_Remote.Indication  (
                      Identifier
                      DLC
                      )
```

The arrival of an MSDU transmission request shall be indicated to the LLC sublayer only if it has been received correctly.

10.2.2.6.3 Effect of receipt

The effect of receipt on this primitive by the LLC sublayer is unspecified.

10.2.2.7 MA_Remote.Confirm

10.2.2.7.1 Function

The MA_Remote.Confirm primitive shall be passed from the local MAC sublayer to the LLC sublayer to convey the results of the previous MA_Remote.Request. This primitive is a remote confirmation, i.e. it shall indicate that the remote MAC entity or entities have passed the associated indication primitive to the corresponding user(s).

10.2.2.7.2 Semantics of MA_Remote.Confirm primitive

The primitive shall provide parameters as follows:

```
MA_Remote.Confirm  (
                   Identifier
                   Transmission_Status
                   )
```

The Transmission_Status shall be used to indicate the success or failure of the previous MA_Remote.Request primitive.

Transmission_Status:[Success, No_Success]

Failures are either errors which occurred during transmission or loss of the arbitration.

10.2.2.7.3 Effect on receipt

The effect on receipt of this primitive by the LLC sublayer is unspecified.

10.2.2.8 MA_OVLD.Request

10.2.2.8.1 Function

The MA_OVLD.Request primitive shall be passed from the LLC sublayer to the MAC sublayer to request transmission of a MAC OVLD frame (see 10.4.5). The OVLD frame shall be a fixed format frame and completely constructed in the MAC sublayer.

10.2.2.8.2 Semantics of MA_OVLD.Request primitive

The primitive shall not provide any parameter:

```
MA_OVLD.request  (
                 )
```


10.2.2.8.3 Effect on receipt

Receipt of this primitive shall cause the MAC sublayer to form an overload frame. The overload frame shall be passed to the lower protocol layers for transfer to the peer MAC sublayer entities.

10.2.2.9 MA_OVLD.Indication

10.2.2.9.1 Function

The MA_OVLD.Indication primitive shall be passed from the MAC sublayer to the LLC sublayer to indicate that an overload frame has been received (see 10.4.5).

10.2.2.9.2 Semantics of MA_OVLD.Indication primitive

The primitive shall not provide any parameters:

```
MA_OVLD.Indication (
                   )
```

10.2.2.9.3 Effect on receipt

The effect on receipt of this primitive by the LLC sublayer is unspecified.

10.2.2.10 MA_OVLD.Confirm

10.2.2.10.1 Function

The MA_OVLD.Confirm primitive shall be passed from the MAC sublayer to the LLC sublayer to indicate that an overload frame has been sent. This confirmation shall be local, i.e. it shall not imply that the remote peer protocol entities have received the overload frame correctly.

10.2.2.10.2 Semantics of MA_OVLD.Confirm primitive

The primitive shall provide parameters as follows.

```
MA_OVLD.Confirm (
                Transmission_Status
                )
```

The Transmission_Status shall be used to indicate the success or failure of the previous MA_OVLD.request primitive.

Transmission_Status:[Success, No_Success]

10.2.2.10.3 Effect on receipt

The effect on receipt of this primitive by the LLC sublayer is unspecified.

10.3 Functional model of MAC sublayer architecture

10.3.1 Capability

The functional capabilities of the MAC sublayer are described by use of the functional model specified in ISO/IEC 8802-3 (see also Figure 5). In this model the MAC sublayer is divided into two fully independently-operating parts: the transmit and the receive. The functions of both transmit and receive parts shall be as given in this clause and Figure 5.

10.3.2 Frame transmission

Frame transmission shall fulfil the following requirements.

a) Transmit data encapsulation:

   1) acceptance of LLC frames and interface control information;

   2) CRC sequence calculation;

   3) construction of MAC frame by adding SOF, SRR bit, IDE bit, RTR bit, reserved bits, CRC, ACK and EOF to the LLC frame (restricted LLC sublayers may not request the transmission of MAC frames with identifiers or data fields outside their restrictions, see 8.5).

b) Transmit medium access management:

   1) initiation of the transmission process after recognizing bus idle (compliance with interframe space);

   2) serialization of the MAC frame;

   3) insertion of stuff bits (bit stuffing);

   4) arbitration and passing into receive mode in case of loss of arbitration;

   5) error detection (monitoring, format check);

   6) ACK check;

   7) recognition of an overload condition;

   8) overload frame construction and initiation of transmission;

   9) error frame construction and initiation of transmission;

   10) presentation of a serial bit stream to the PL for transmission.

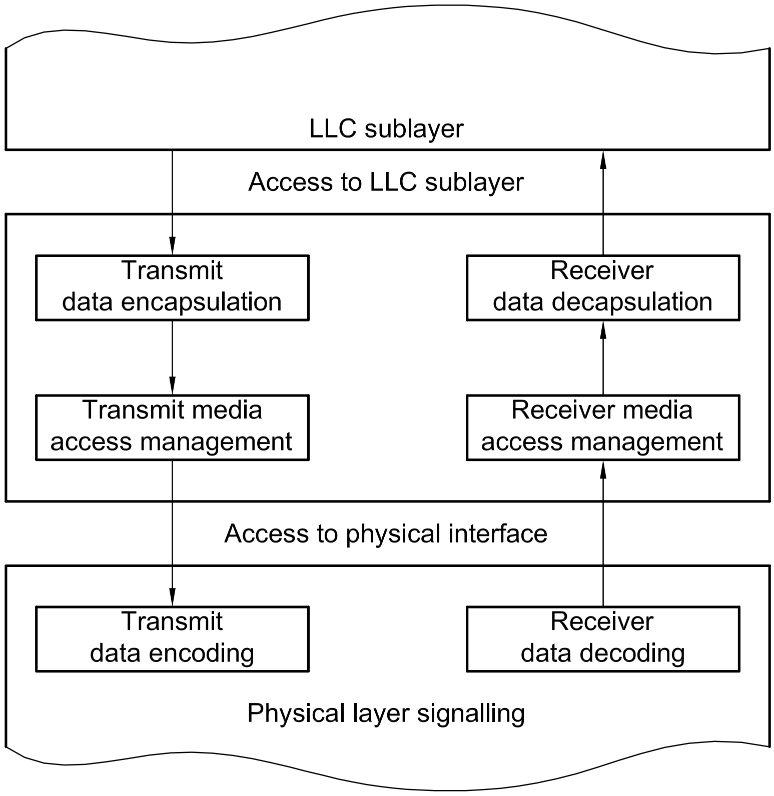

Figure 5 — MAC functions

10.3.3 Frame reception

Frame reception shall fulfil the following requirements.

a) Receive medium access management:

1) reception of a serial bit stream from the PL;

2) deserialization and recompiling of the frame structure;

3) deletion of stuffbits (bit destuffing);

4) error detection (CRC, format check, stuff rule check);

5) transmission of ACK;

6) error frame construction and initiation of transmission;

7) recognition of an overload condition;

8) reactive overload frame construction and initiation of transmission;

b) Receive data decapsulation:

1) removing the MAC-specific information from the received frame;

2) presenting the LLC frame and interface control information to the LLC sublayer (for restricted LLC sublayers only limited parts of the LLC frame are presented, see 8.5).

10.4 Structure of MAC frames

10.4.1 Description

Data transmission and reception between nodes in a CAN system shall be performed and controlled by four different frame types:

— a data frame that carries data from a transmitter to all receivers;

— a remote frame transmitted by a node for requesting transmission of the data frame with the same identifier;

— an error frame transmitted by any node (transmitter or receiver) in case of a bus error detected;

— an overload frame used for providing an extra delay between the preceding and succeeding data frames or remote frames.

Data frames and remote frames shall be separated from preceding frames by an interframe space.

10.4.2 Specification of MAC data frame

10.4.2.1 Description

MAC data frames shall be composed of seven different bit fields (see also Figure 6):

— SOF;

— arbitration field;

— control field;

— data field;

— CRC field;

— ACK field;

— EOF.

Start of frame	Arbitration field	Control field	Data field	CRC field	ACK field	End of frame

Figure 6 — MAC data frame

10.4.2.2 SOF

SOF shall mark the beginning of data and remote frames. It shall consist of a single dominant bit.

A node shall send a SOF only when the bus is idle (see 10.4.6.3). A node sampling a dominant bit during suspend transmission or at the third bit of intermission shall accept it as SOF.

If a node has a pending transmission that is error-active or which has been the receiver of the previous frame samples, a dominant bit at the third bit of intermission shall, with the next bit, start transmitting its message with the first bit of its identifier without first transmitting a SOF bit and without becoming the receiver.

All nodes shall synchronize to the leading edge caused by SOF of the node starting transmission first.

10.4.2.3 Arbitration field

The arbitration field shall be composed of the identifier field, passed from the LLC sublayer, and the RTR bit. The value of the RTR bit shall be dominant in a MAC data frame. Depending on the identifier's extension flag, the arbitration field may have two formats: the base format and the extended format.

a) In the base format (where the extension flag is dominant), the arbitration field shall consist of the eleven (11) bit base identifier and the RTR bit. The identifier bits shall be denoted ID-28 to ID-18.

b) In the extended format (where the extension flag is recessive), the arbitration field shall consist of the base identifier (ID-28 to ID-18), the SRR and IDE bits (both bits recessive), the identifier extension (ID-17 to ID-0), and the RTR bit.

   1) SRR bit (extended format only)

      The SRR bit shall be transmitted in extended format frames at the position of the RTR bit in base format frames, and so substitutes the RTR bit in the base format frame. A transmitter shall only send recessive SRR bits, but receivers shall accept recessive or dominant SRR bits.

   2) IDE bit

      The IDE bit shall distinguish between base format and extended format, i.e. whether it belongs to

      i) the arbitration field for the extended format, or

      ii) the control field for the base format.

      The IDE bit in the extended format shall be transmitted as recessive, whereas in the standard format the IDE bit shall be dominant. It shall be transmitted after the base identifier and the RTR bit (base format) or after the base identifier and the SRR bit (extended format).

      Therefore, collisions of a base format frame and an extended format frame, with both frames having the same base identifier, shall be resolved such that the base format frame prevails over the extended format frame.

10.4.2.4 Control field

The control field shall consist of six bits, where the last four bits shall be the DLC, passed from the LLC sublayer (see 8.4.2) and the format of the first two bits shall be different for base format frame and extended format frame.

a) In the base format frame, the first two bits shall be the IDE bit, transmitted as dominant, and an additional bit reserved r0 for future expansion. Receivers shall accept recessive and dominant bits as bit reserved r0. Until the function of the reserved bit is specified, the transmitter shall only send a dominant bit.

b) In the extended format frame, the first two bits r1 and r0 shall be reserved for future expansion. Receivers shall accept recessive and dominant bits as reserved bits in all combinations. Until the function of the reserved bits are specified, the transmitter shall only send dominant bits.

10.4.2.5 Data field

The MAC frame data field shall be equivalent to the LLC data field (see 8.4.2).

10.4.2.6 CRC field

The CRC field shall contain the CRC sequence followed by a CRC delimiter.

a) CRC sequence

The frame check sequence shall be derived from a CRC (BCH-code) best suited for frames with bit counts less than 127 bits.

In order to carry out the CRC calculation, the polynomial to be divided shall be specified as the polynomial, the coefficients of which are given by the destuffed bit stream consisting of SOF, arbitration field, control field, data field (if present) and, for the 15 lowest coefficients, by 0. This polynomial shall be divided (the coefficients are calculated modulo-2) by the generator-polynomial:

$$X^{15} + X^{14} + X^{10} + X^{8} + X^{7} + X^{4} + X^{3} + 1$$

The remainder of this polynomial division shall be the CRC sequence transmitted over the bus.

In order to implement this function, a 15-bit shift register CRC_RG(14 : 0) may be used. If NXTBIT denotes the next bit of the bit stream given by the destuffed bit sequence from SOF until the end of the data field, the CRC sequence shall be calculated as follows.

```
CRC_RG(14 : 0) = (0,...,0);                          //initialize shift register
REPEAT
    CRCNXT = NXTBIT EXOR CRC_RG(14);
    CRC_RG(14 : 1) = CRC_RG(13 : 0);                 //shift left by one position
    CRC_RG(0) = 0;
    IF CRCNXT THEN
        CRC_RG(14 : 0) = CRC_RG(14:0) EXOR (4599hex);
    ENDIF
UNTIL (NXTBIT = [End of data] or there is an error condition).
```

After the transmission/reception of the last bit of the data field, CRC_RG shall contain the CRC sequence.

b) CRC delimiter

The CRC sequence shall be followed by the CRC delimiter consisting of a single recessive bit.

10.4.2.7 ACK field

The ACK field shall be two bits long and shall contain the ACK slot and the ACK delimiter. In the ACK field, the transmitter node shall send two recessive bits.

a) ACK slot

All nodes that have received the matching CRC sequence shall send an ACK within the ACK slot by overwriting the recessive bit of the transmitter by a dominant bit.

b) ACK delimiter

The ACK delimiter being the second bit of the ACK field shall be a recessive bit. As a consequence, the ACK slot shall be surrounded by two recessive bits (CRC delimiter, ACK delimiter).

10.4.2.8 EOF

Each data frame and remote frame shall be delimited by a flag sequence consisting of seven recessive bits forming the EOF.

10.4.3 Specification of MAC remote frame

10.4.3.1 Description

A node acting as a receiver for certain data may initiate the transmission of the respective data by its source node by sending a remote frame as given in Figure 7.

Start of frame	Arbitration field	Control field	CRC field	ACK field	End of frame

Figure 7 — MAC remote frame

10.4.3.2 Identical fields of MAC data frame and MAC remote frame

The bit fields SOF, CRC field, ACK field and EOF shall be equivalent to the corresponding bit fields of the MAC data frame (see 10.4.2). There shall be no data field in the remote frame.

10.4.3.3 Arbitration field

The arbitration field shall be composed of the identifier field, passed from the LLC sublayer, and the RTR bit. In both formats (base and extended — see 10.4.2.3), the value of the RTR bit in a MAC remote frame shall be recessive.

10.4.3.4 Control field

In base format and in extended format the control field of the MAC remote frame shall be equivalent to the control field of the MAC data frame (see 10.3.1.3). The collision resolution (see 10.8.8) requires that the value of a remote frame's DLC equal the DLC of the requested data frame.

10.4.4 Specification of error frame

10.4.4.1 Description

The error frame shall consist of two different fields. The first field shall be as given by the superposition of error flags contributed from different nodes. The then following second field shall be the error delimiter.

10.4.4.2 Error flag

Two forms of error flag may be used, the active error flag and the passive error flag, where

— the active error flag shall consist of six consecutive dominant bits, and

— the passive error flag shall consist of six consecutive recessive bits, of which some or all may be overwritten by dominant bits from other nodes.

An error-active node detecting an error condition shall signal this by sending an active error flag. The form of error flag violates the rule of bit stuffing or destroys the bit field requiring fixed form. As a consequence, all other nodes shall detect an error condition, too, and on their part shall start sending an error flag. So the sequence of dominant bits, which may actually be monitored on the bus, results from a superposition of different error flags sent by individual nodes. The total length of this sequence may vary between a minimum of six (6) and a maximum of twelve (12) bits.

Passive error flags initiated by a transmitter shall cause error(s) (with two exceptions) at the receiver(s) when they start in a frame field encoded by the method of bit stuffing, because then they lead to stuff errors detected by the receivers. The first exception is a passive error flag that starts during arbitration and another node continues transmitting; the second exception is a passive error flag that starts less than six bits before the end of the CRC sequence and the last bits of the CRC sequence happen to be all recessive.

Passive error flags initiated by receivers shall not be able to prevail over any activity on the bus. Therefore, error-passive receivers shall always wait for six (6) subsequent equal bits after detecting an error condition, until they have completed their error flag.

10.4.4.3 Error delimiter

The error delimiter shall consist of eight recessive bits. After sending an error flag, each node shall send recessive bits and monitor the bus until it detects a recessive bit. Afterwards, it shall start sending seven more recessive bits.

10.4.5 Specification of overload frame

10.4.5.1 Types

The following types of overload frame shall have the same format.

a) LLC requested overload frame: requested by the LLC sublayer to indicate an internal overload situation (see 10.11).

b) Reactive overload frame: the transmission of the reactive overload frame shall be initiated by the MAC sublayer upon certain error conditions (see 10.11).

The overload frame shall contain two bit fields: overload flag and overload delimiter. The overall flag shall correspond to that of the active error flag. The overload delimiter shall be the same as the error delimiter.

10.4.5.2 Overload flag

The overload flag shall consist of six dominant bits. It destroys the fixed form of the intermission field (see 10.4.6). As a consequence, all other nodes also detecting an overload condition shall, on their part, start sending an overload flag.

10.4.5.3 Overload delimiter

The overload delimiter shall consist of eight recessive bits. After sending an overload flag, every node shall monitor the bus until it detects a recessive bit. At this point in time, every node shall finish sending its overload flag and all nodes shall start sending seven more recessive bits simultaneously, to complete the eight-bit-long overload delimiter.

10.4.6 Specification of interframe space

10.4.6.1 Description

Data frames and remote frames shall be separated from preceding frames, regardless of type (data frame, remote frame, error frame, overload frame), by a bit field called interframe space. On the contrary, overload frames and error frames shall not be preceded by an interframe space, and multiple overload frames shall not be separated by an interframe space.

The interframe space shall contain the bit fields intermission and bus idle, and suspend transmission for error-passive nodes which have been the transmitter of the previous frames (see Figures 8 and 9).

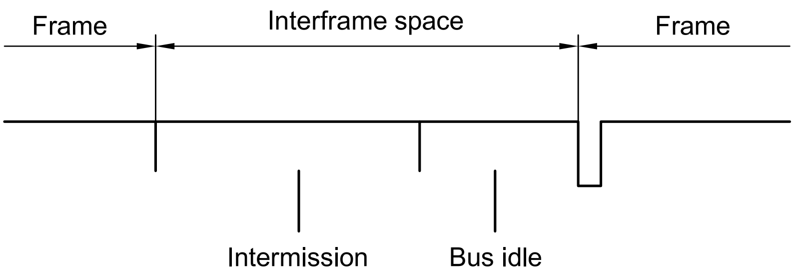

Figure 8 — Interframe space for nodes not error-passive or receiver of previous frame

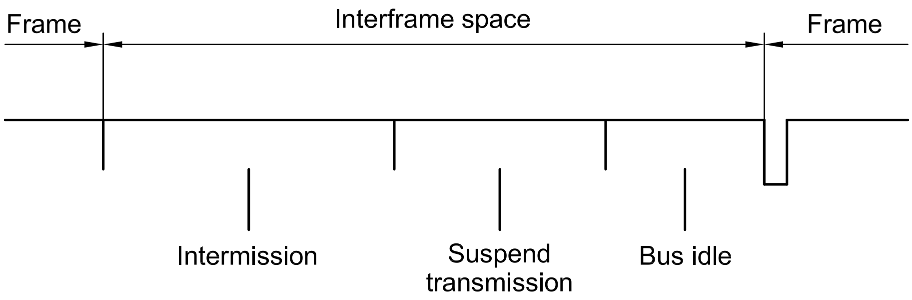

Figure 9 —Interframe space for error-passive nodes which were transmitter of previous frame

10.4.6.2 Intermission

The intermission field shall consist of three recessive bits. During intermission no node shall start transmission of a data frame or remote frame. Only signalling of an overload condition is allowed.

The detection of a dominant bit on the bus at the third bit of intermission shall be interpreted as SOF (see 10.4.2.2).

10.4.6.3 Bus idle

The period of bus idle may be of arbitrary length. The bus shall recognize the idle condition and any node may access the bus for transmission. A frame pending for transmission during the transmission of another frame shall be started in the first bit following intermission.

The detection of a dominant bit on the bus during bus idle shall be interpreted as SOF.

10.4.6.4 Suspend transmission

An error-passive node which has been the transmitter of the previous frame shall send eight (8) recessive bits following intermission before starting transmission of a further frame. If, meanwhile, a transmission (caused by another node) starts, the node shall become a receiver of this frame.

10.5 Frame coding

The SOF segments such as frame, arbitration field, control field, data field and CRC sequence shall be coded by the method of bit stuffing. Whenever a transmitter detects five consecutive bits (including stuff bits) of identical value in the bit stream to be transmitted, it shall automatically insert a complementary bit in the actual transmitted bit stream (see Table 6).

Table 6 — Bit stuffing

Destuffed bit stream	01011111010	10100000101	01011111000010	10100000111101
Stuffed bit stream	01011111o010	10100000I101	01011111o0000I10	10100000I1111o01
0 = dominant bit, o = dominant stuff bit, 1 = recessive bit, and I = recessive stuff bit.				

The remaining bit fields of the data frame or remote frame (CRC delimiter, ACK field and EOF) shall be of fixed form and not stuffed.

The error frame and the overload frame shall be of fixed form as well and shall not be coded by the method of bit stuffing.

The bit stream in a frame shall be coded according to the NRZ method. This means that the generated bit level is constant during the total bit time.

10.6 Order of bit transmission

A frame shall be transferred bit field by bit field, starting with its SOF field. Within a field the MSB shall be transmitted first (see Figures 10 and 11).

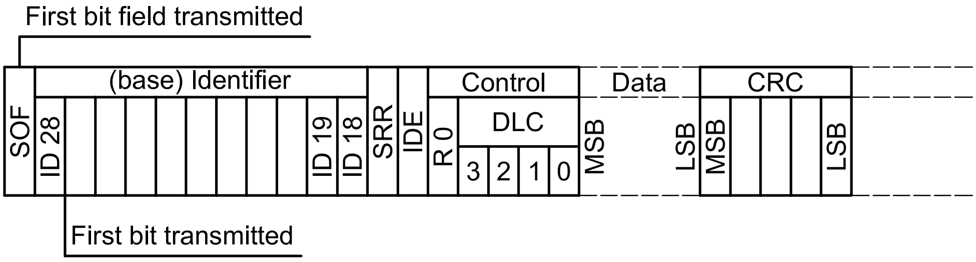

Figure 10 — Order of bit transmission (base format)

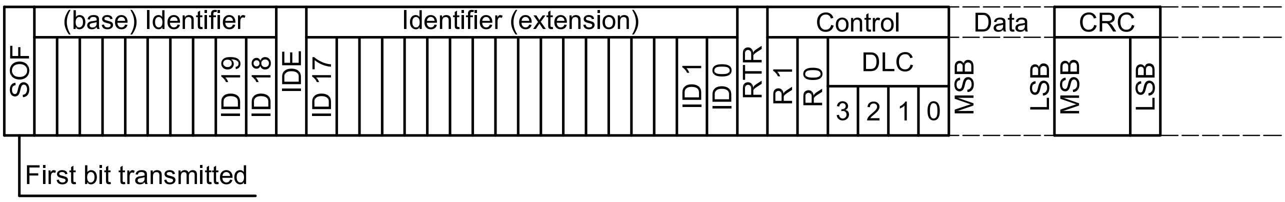

Figure 11 — Order of bit transmission (extended format)

10.7 Frame validation

The point of time at which a frame is taken to be valid shall be different for the transmitter and the receiver of the frame.

a) The frame shall be valid for a transmitter if there is no error until the end of EOF. If a frame is corrupted, recovery shall be processed as described in 8.3.3.

b) The frame shall be valid for receivers if there is no error until the last-but-one bit of EOF. The value of the last bit of EOF shall be treated as 'do not care', and a dominant value shall not lead to a form error. A receiver that detects a dominant bit at the last bit of EOF shall respond with an overload frame (see 10.11).

NOTE If both the receiver and the transmitter detect a dominant bit at the last bit of EOF (global error), then the frame is valid for the receiver, but not for the transmitter. The transmitter repeats the transmission, and the frame is received twice.

10.8 Medium access method

10.8.1 General

This clause describes the functions and characteristics related to the medium access method of CAN.

10.8.2 Multimaster

Every node transmitting a data frame or a remote frame shall be the bus master during that transmission.

10.8.3 Bus free detection

The bus shall be considered to be free for any node after having detected that the bit field intermission has not been interrupted by a dominant bit.

10.8.4 Bus access

An error-active node may access the bus as soon as the bus is free. An error-passive node that is the receiver of the current or previous frame may access the bus as soon as the bus is free. An error-passive node that is transmitter of the current frame or has been the transmitter of the previous frame may access the bus as soon as suspend transmission is finished, provided that no other node has started transmission meanwhile. Whenever several nodes start transmitting in coincidence, that node transmitting the frame with the highest priority at this time shall become the bus master. The mechanism to resolve the resulting bus access conflict shall be contention-based arbitration.

10.8.5 Transmission of MAC frames

MAC data frames and MAC remote frames may be started when the node is allowed to access the bus according to 10.8.4. A MAC error frame shall transmit as specified in 10.10. A MAC overload frame shall be transmitted as specified in 10.11.

10.8.6 Contention-based arbitration

During arbitration, every transmitter shall compare the level of the bit transmitted with the level monitored on the bus. If these levels are equal, the node may continue to send. When a recessive level is sent and a dominant level is monitored, the node loses arbitration and shall withdraw without sending any more bits.

When a dominant level is sent and a recessive level is monitored, the node shall detect a bit error.

Contention-based arbitration is performed on the identifier and on the RTR bit following the identifier.

10.8.7 Frame priority

Among two frames with different identifiers, the higher priority shall be assigned to the frame containing the identifier of lower binary value.

If a data frame and a remote frame with the same identifier are initiated at the same time, the data frame shall have higher priority than the remote frame. This shall be achieved by assigning according values to the RTR bits.

10.8.8 Collision resolution

Additional to the principle that transmission may be initiated only when the bus is free, there exist further principles for the resolution of collision.

— Within one system, each information bit shall be assigned a unique identifier.

— A data frame with a given identifier and a non-zero DLC may only be initiated by one node.

— Remote frames may only be transmitted with a system-wide determined DLC, which shall be the DLC of the corresponding data frame. Simultaneous transmission of remote frames with identical identifiers and different DLCs shall lead to irresolvable collisions (limitation of the contention-based arbitration to the arbitration field).

10.9 Error detection

The MAC sublayer shall provide the following mechanisms for error detection.

- monitoring;
- stuff rule check;
- frame check;
- 15 bit CRC;
- ACK check.

There are five error types, which are not mutually exclusive.

a) Bit error

A node sending a bit on the bus shall also monitor the bus. A bit error is detected at that bit time when the bit value that is monitored differs from the bit value sent.

Exceptions:

- a dominant bit shall not lead to a bit error when a recessive information bit is sent during arbitration, or a recessive bit is sent during ACK slot;
- a node sending a passive error flag and detecting a dominant bit shall not interpret this as a bit error.

b) Stuff error

A stuff error is detected at the bit time of the sixth consecutive equal bit level in a frame field that shall be coded by the method of bit stuffing.

c) CRC error

The CRC sequence shall consist of the result of the CRC calculation of the transmitter. The receivers shall calculate the CRC in the same way as the transmitter. A CRC error shall be detected when the calculated CRC sequence does not equal the received one.

d) Form error

A form error shall be detected when a fixed-form bit field contains one or more illegal bits.

Exception:

- a receiver monitoring a dominant bit at the last bit of EOF, or any node monitoring a dominant bit at the last bit of error delimiter or overload delimiter, shall not interpret this as a form error.

e) ACK error

An ACK error shall be detected by a transmitter whenever it does not monitor a dominant bit during the ACK slot.

Whenever one of these errors is detected, the LLC sublayer shall be informed. As a consequence, the MAC sublayer shall initiate the transmission of an error flag.

10.10 Error signalling

Whenever a bit error, stuff error, form error, or ACK error is detected by any node, an error flag shall be started by the respective node at the next bit.

Whenever a CRC error is detected, an error frame shall be started at the bit following the ACK delimiter, unless an error frame for another error condition has already been monitored.

10.11 Overload signalling

The following conditions shall lead to the transmission of an overload frame.

a) LLC-requested overload frame (initiated by the LLC sublayer): internal conditions of a receiver which require a delay of the next MAC data frame or MAC remote frame.

b) Reactive overload frame (initiated by the MAC sublayer): detection of a dominant bit during first two bits of intermission, detection of a dominant bit in the last bit of EOF by a receiver, or detection of a dominant bit by any node at the last bit of error delimiter or overload delimiter.

An LLC-requested overload frame shall only be started at the first bit of an expected intermission, whereas reactive overload frames shall start one bit after detecting the dominant bit due to condition b) above. The start of reactive overload frames due to condition a) above shall be allowed, but not required to be implemented.

At most, two LLC overload frames may be generated to delay the next MAC data frame or MAC remote frame.

10.12 Bus monitoring

In an optional bus monitoring mode, the CAN node shall be able to receive valid data frames and valid remote frames, but it sends only recessive bits on the CAN bus and does not start a transmission. If the MAC sublayer is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit shall be rerouted internally so that the MAC sublayer monitors this dominant bit, although the CAN bus may remain in recessive state.

11 LLC and MAC sublayer conformance

For an implementation to be in conformance, LLC and MAC sublayers shall comply with all specifications and values given in this part of ISO 11898.

12 Physical layer

12.1 General

The PL shall be an electrical circuit that connects a CAN node to a bus. The total number of CAN nodes shall be limited by the electric loads on the bus.

12.2 Functional model

The PL shall be modelled according to ISO/IEC 8802-3 (see also Figure 12). The PL shall be divided into three parts.

a) PLS shall encompass functions related to bit representation, timing and synchronization.

b) PMA sublayer shall encompass functional circuitry for bus line transmission/reception and may provide means for bus failure detection.

c) MDI shall encompass the mechanical and electrical interfaces between the physical medium and the MAU.

The MAU shall denote the functional part of the PL used to couple the node to the transmission medium. The MAU consists of PMA and MDI.

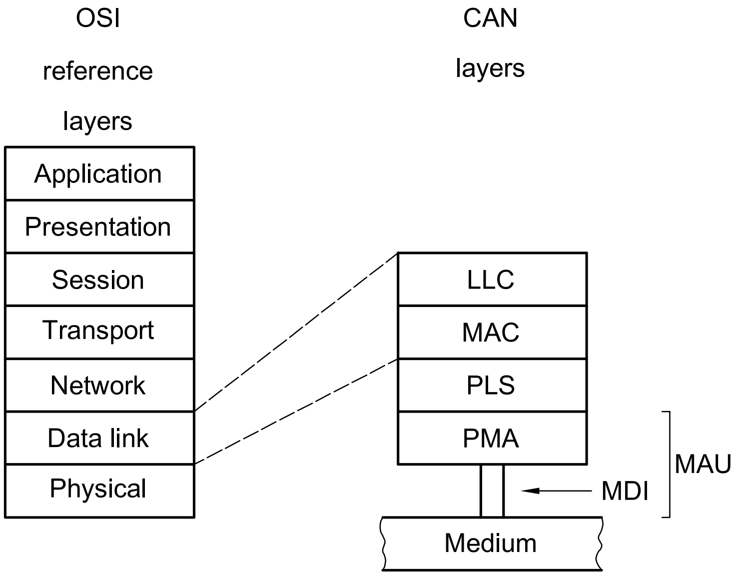

Figure 12 — Model of PL architecture

12.3 Services of PL

12.3.1 Description

The services of PL shall allow the local MAC sublayer entity to exchange data bits with peer MAC sublayer entities.

The PL shall provide the following service primitives to the MAC sublayer.

PLS_Data.request,

PLS_Data.indicate.

12.3.2 PLS_Data.Request

The PLS_Data.Request primitive shall be passed from the MAC sublayer to the PL to request transmission of a dominant or recessive bit. The primitive provides the following parameter:

```
PLS_Data.Request (
                  Output_Unit
                  )
```

The Output_Unit parameter may take on one of the two values: dominant or recessive.

12.3.3 PLS_Data.Indicate

The PLS_Data.Indicate primitive shall be passed from the PL to the MAC sublayer in order to indicate the arrival of a dominant or recessive bit. The primitive shall provide the following parameter.

```
PLS_Data.Indicate (
                   Input_Unit
                   )
```

The Input_Unit parameter may take on one of the two values each representing a single bit: dominant or recessive.

12.4 PLS sublayer specification

12.4.1 Bit encoding/decoding

12.4.1.1 Bit time

Bus management functions executed within the bit time frame, such as CAN node synchronization behaviour, network transmission delay compensation, and sample point positioning, shall be as given by the programmable bit timing logic of the CAN protocol IC.

a) **Nominal bit rate (BR)** gives the number of bits per second transmitted in the absence of resynchronization by an ideal transmitter.

b) **Nominal bit time**, t_B

$$t_B = \frac{1}{BR}$$

The nominal bit time may be thought of as being divided into separate non-overlapping time segments. These segments shall form the bit time as shown in Figure 13:

— synchronization segment (Sync_Seg),

— propagation time segment (Prop_Seg),

— phase buffer segment 1 (Phase_Seg1),

— phase buffer segment 2 (Phase_Seg2).

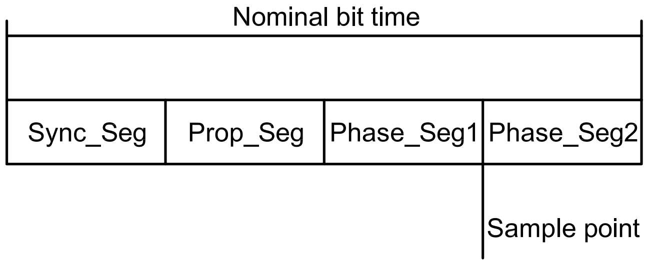

Figure 13 — Partition of bit time

— **Sync_Seg**

This part of the bit time shall be used to synchronize the various CAN nodes on the bus. An edge is expected within this segment.

— **Prop_Seg**

This part of the bit time shall be used to compensate physical delay times within the network. These delay times consist of the signal propagation time on the bus and the internal delay time of the CAN nodes.

— **Phase_Seg1, Phase_Seg2**

These phase buffer segments are used to compensate for edge phase errors. These segments may be lengthened or shortened by resynchronization.

— **Sample point**

The sample point shall be the point of time at which the bus level is read and interpreted as the value of that respective bit. Its location shall be at the end of Phase_Seg1.

— **Information processing time**

The information processing time shall be the time segment starting with the sample point reserved for calculation of the subsequent bit level.

— **Resynchronization jump width (SJW)**

As a result of resynchronization, Phase_Seg1 may be lengthened or Phase_Seg2 may be shortened. The amount of lengthening and shortening of the phase buffer segments has an upper limit given by the resynchronization jump width.

— **Internal delay time**

The internal delay time of a CAN node, t_{node}, shall be the sum of all asynchronous delays that occur along the transmission and reception path, relative to the bit timing logic unit of the protocol IC of individual CAN nodes. For more details see Figure 14.

In relation to Figure 14:

1) the sum of output and input CAN node delays is critical relative to the bit timing logic, with the important characteristic parameter of a CAN node given by

$$t_{node} = t_{output} + t_{input}$$

2) for proper arbitration, the following conditions shall be met —

$$t_{Prop\_Seg} \geqslant t_{node\ A} + t_{node\ B} + 2t_{busline}$$

— i.e. the leading transmitting bit timing logic with respect to synchronization of CAN, node A shall be able to know the correct bus level of bit n at the sampling point. The tolerable values of t_{node} depend on the required bit rate and line length of the bus (maximum distance between any two nodes) and of the possible bit timing as shown by the arbitration condition.

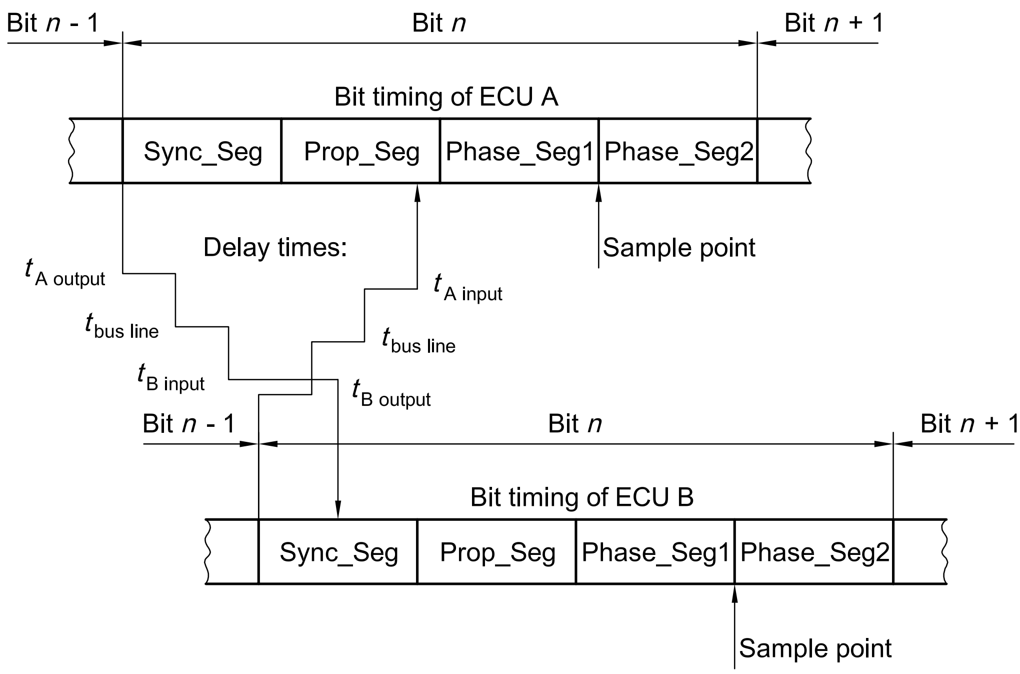

Figure 14 — Time relationship between CAN node bit timing A and B during arbitration, bit timing of two CAN nodes and delay times

12.4.1.2 Programming of bit time

Programming of bit time shall be performed using the following periods of time (see also Figure 14).

a) Time quantum

The time quantum shall be a fixed unit of time derived from the oscillator period. There shall exist a programmable prescaler, with integral values, ranging at least from one (1) to thirty-two (32). Starting with the minimum time quantum, the time quantum shall have a length of

Time quantum = $m \cdot$ minimum time quantum

where m is the value of the prescaler.

b) Nominal length of time segments (in absence of synchronization)

— Sync_Seg shall be one time quantum long.

— The information processing time shall be less than or equal to two time quanta long.

— Prop_Seg shall be programmable to be 1, 2, 3, . . ., 8 or more time quanta long. It shall be programmed to compensate the delay times of the actual network, rounded up to the nearest integer time quantum.

— Phase_Seg1 shall be programmable to be 1, 2, 3, . . ., 8 or more time quanta long.

— Phase_Seg2 shall be programmed to the maximum of Phase_Seg1 and the information processing time.

— SJW shall be programmable between 1 and the minimum of Phase_Seg1 and 4.

In a CAN implementation, Prop_Seg and Phase_Seg1 do not need to be programmable separately; it is sufficient to program the sum of Prop_Seg and Phase_Seg1.

The total number of time quanta in a bit time shall be programmable at least from 8 to 25.

In case of synchronization, Phase_Seg1 may be longer and Phase_Seg2 may be shorter than its programmed nominal value.

The frequencies of the oscillators in the different CAN nodes shall be coordinated in order to provide a system-wide specified time quantum. The acceptable oscillator tolerances of the protocol ICs (see 12.4.2.5) and the potential for incorrect synchronization is determined by Phase_Seg1 and Phase_Seg2.

12.4.2 Synchronization

12.4.2.1 Description

Hard synchronization and resynchronization shall be two forms of synchronization. They shall obey the following rules.

a) Only one synchronization within one bit time (between two sample points) shall be allowed.

b) A recessive to dominant edge shall be used for synchronization only if the bus state detected at the previous sample point (previous read bus state) differs from the bus state immediately after the edge.

c) Hard synchronization shall be performed during interframe space (with the exception of the first bit of intermission) whenever there is a recessive to dominant edge.

d) All other recessive to dominant edges fulfilling rules 1) and 2) shall be used for resynchronization with the exception that a node sending a dominant bit does not perform resynchronization as a result of a recessive to dominant edge with a positive phase error (see 12.4.2.2).

Clocking information shall be derived from transitions from one bit value to the other. The property (owing to the bit stuffing) that only a fixed maximum number of successive bits have the same value shall provide the possibility of resynchronizing a bus unit to the bit stream during a frame.

12.4.2.2 Phase error of synchronization edge

The phase error, e, of an edge is given by the position of the edge relative to Sync_Seg, measured in time quanta. The sign of phase error is given by

$e = 0$ if the edge lies within Sync_Seg,

$e > 0$ if the edge lies before the sample point, and

$e < 0$ if the edge lies after the sample point.

12.4.2.3 Hard synchronization

After a hard synchronization, the bit time shall be restarted by each bit timing logic unit with Sync_Seg. Thus hard synchronization shall force the edge which has caused the hard synchronization to lie within the synchronization segment of the restarted bit time.

12.4.2.4 Bit resynchronization

Resynchronization shall lead to a shortening or lengthening of the bit time such that the position of the sample point is correct, when the magnitude of the phase error of the edge which causes resynchronization is less than or equal to the programmed value of the resynchronization jump width. When the magnitude of the phase error is larger than the resynchronization jump width,

— if the phase error e is positive, Phase_Seg1 shall be lengthened by an amount equal to the resynchronization jump width, whereas

— if the phase error e is negative, then Phase_Seg2 shall be shortened by an amount equal to the resynchronization jump width.

If Phase_Seg2 is shortened to a value less than the information processing time, the calculation of the subsequent bit level may be completed after the end of Phase_Seg2.

12.4.2.5 Tolerance range of the oscillator frequencies

The tolerance range of an oscillator frequency f_{osc} around the nominal frequency f_{nom} depends on Phase_Seg1, Phase_Seg2, SJW, and the bit time. The maximum tolerance df of f_{osc} with

$[(1 - df) \cdot f_{\text{nom}} \leqslant f_{\text{osc}} \leqslant (1 + df) \cdot f_{\text{nom}}]$

shall take into consideration the following two conditions (both shall be met):

1) $df \leqslant \dfrac{(\text{Phase\_Seg1, Phase\_Seg2})_{\text{min}}}{2 \times (13 \times t_{\text{bit}} - \text{Phase\_Seg2})}$

2) $df \leqslant \dfrac{\text{SJW}}{20 \times t_{\text{bit}}}$

The maximum difference between two oscillators shall be

$$2 \times df \times f_{\text{nom}}$$

12.5 PLS-PMA interface specification

12.5.1 PLS to PMA messages

12.5.1.1 Output message

The PLS sublayer shall send an output message to the PMA sublayer whenever it receives an Output_Unit from the MAC sublayer. The output message causes the PMA to send a dominant or recessive bit.

12.5.1.2 Bus_off message

The PLS sublayer shall send a bus_off message to the PMA sublayer whenever it receives a bus_off_request from the supervisor (see 13.1).

12.5.1.3 Bus_off_release message

The PLS sublayer shall send a bus_off_release message to the PMA sublayer whenever it receives a bus_off_release_request from the supervisor (see 13.1).

12.5.2 PMA to PLS message

12.5.2.1 Input message

The PMA sublayer shall send an input message to the PLS sublayer whenever the MAU has received a bit from the medium. The input signal indicates to the PLS the arrival of a dominant or recessive bit.

13 Description of supervisor

13.1 Fault confinement

13.1.1 Objectives

The objective of fault confinement is to preserve a high availability of the data transmission system, even in the case of a node defect. Therefore the fault confinement strategies have to prove reliable on the

— distinction between temporary errors and permanent failures, and

— localization and switching-off of faulty nodes.

13.1.2 Strategies

Each node shall be supplied with a transmit error counter and a receive error counter. The former of these shall register the number of errors during the transmission, and the latter the errors during the reception of frames.

If frames are sent or received correctly, the counters shall be decreased. In case of errors, the counters shall be increased more than they are decreased in case of absence of errors. The ratio in which the counters are increased/decreased depends on the acceptable ratio of invalid/valid frames on the bus. At any time, levels of the error counters reflect the relative frequency of previous disturbances.

Depending on predetermined counter values, the behaviour of nodes in respect to errors shall be modified — this shall range from a prohibition of sending error flags to cancel frames, up to switching-off of nodes which often send invalid frames.

13.1.3 Fault confinement interface specification

13.1.3.1 Description

Fault confinement interface shall be as shown in Figure 15.

13.1.3.2 LLC sublayer/FCE interface

The messages interchanged between the FCE and the LLC sublayer shall be as given in Table 7.

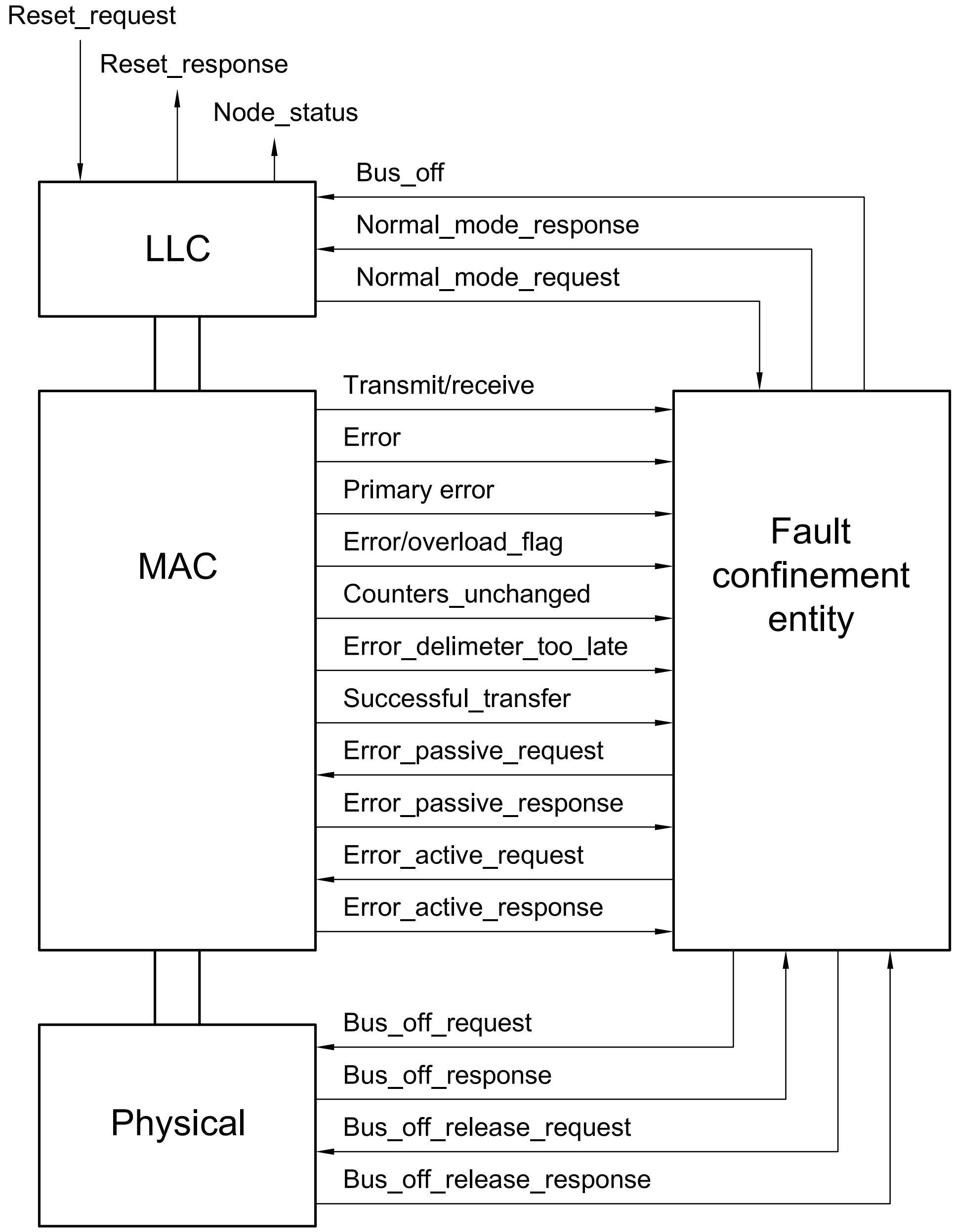

Figure 15 — Fault confinement interface

Table 7 — Messages between LLC and FCE

a) LLC to FCE message	
LLC to FCE message	**Meaning**
Normal_Mode_Request	Resets FCE to initial state[a]
b) FCE to LLC messages	
FCE to LLC message	**Meaning**
Normal_Mode_Response	Response to the Normal_Mode_Request
Bus_Off	Indicates that the node is in the bus-off state
[a] The values of the transmit error counter and the receive error counter are set to zero when the FCE passes into its initial state	

13.1.3.3 MAC sublayer/FCE interface

The message exchanges between the FCE and the MAC sublayer shall be as given in Table 8.

Table 8 — Messages between MAC and FCE

a) MAC to FCE messages	
MAC to FCE message	**Meaning**
Transmit/receive	Indicates the node's current transfer mode
Error	Indicates that the MAC sublayer has detected an error (bit error, stuff error, CRC error, form error, ACK error)
Primary_error	Signals that the MAC sublayer has detected a dominant bit after sending an error flag (indicates that the MAC sublayer has detected a primary error and not an error that is caused by the error flag of another node)
Error/overload flag	Indicates that the MAC sublayer is sending an error flag or an overload flag
Counters_unchanged	Indicates that the FCE counters remain unchanged [due to special cases — see 13.1.4.2, c)].
Error_delimiter_too_late	Indicates that the MAC sublayer is waiting too long for error delimiter. This signal is sent each time after a sequence of eight consecutive dominant bits have been received after sending an error flag
Successful_transfer	Indicates that transmission/reception was successfully completed
Error_passive_response	Indicates that the node was set into the error-passive state
Error_active response	Indicates that the node was set into the error-active state again
b) FCE to MAC message	
FCE to MAC messages	**Meaning**
Error_passive_request	Request to set the node into the error-passive state
Error_active_request	Request to set the node into the error-active state

13.1.3.4 PL/FCE interface

The messages exchanged between the FCE and the PL shall be as given in Table 9.

Table 9 — Messages between FCE and PL

a) FCE to PL messages	
FCE to PL message	**Meaning**
Bus_off_request	Request to switch off the node from the bus
Bus_off_release_request	Request to set the node into the normal transmit/receive node
b) PL to FCE messages	
PL to FCE message	**Meaning**
Bus_off_response	Response to bus_off_request
Bus_off_release_response	Response to bus_off_release_request

13.1.4 Rules of fault confinement

13.1.4.1 Description

With respect to fault confinement, a node may be in one of the three states, depending on the error counter levels (see 6.13 to 6.15):

— error-active;

— error-passive;

— bus-off.

13.1.4.2 Error counting

The error counters shall be modified according to the following rules (more than one rule may apply during a given frame transfer).

a) When a receiver detects an error, the receive error counter shall be increased by 1, except when the detected error was a bit error during the sending of an active error flag or an overload flag.

b) When a receiver detects a dominant bit as the first bit after sending an error flag, the receive error counter shall be increased by eight (8).

c) When a transmitter sends an error flag, the transmit error counter shall be increased by eight (8).

   1) Exception 1: if the transmitter is error-passive and detects an ACK error because of not detecting a dominant ACK and does not detect a dominant bit while sending its passive error flag.

   2) Exception 2: if the transmitter sends an error flag because a stuff error occurred during arbitration, whereby the stuffbit should have been recessive, and has been sent as recessive but is monitored as dominant.

   In both exceptions the transmit error counter remains unchanged.

d) If a transmitter detects a bit error while sending an active error flag or an overload flag, the transmit error counter shall be increased by eight (8).

e) If a receiver detects a bit error while sending an active error flag or an overload flag, the receive error counter shall be increased by eight (8).

f) Any node shall tolerate up to seven (7) consecutive dominant bits after sending an active error flag, passive error flag or overload flag. After detecting the fourteenth (14) consecutive dominant bit (in case of an active error flag or an overload flag) or after detecting the eighth consecutive dominant bit following a passive error flag, and after each sequence of additional eight consecutive dominant bits, every transmitter shall increase its transmit error counter by eight (8) and every receiver shall increase its receive counter by eight (8).

g) After the successful transmission of a frame (getting ACK and no error has been detected until EOF is finished), the transmit error counter shall be decreased by one (1) unless it was already zero (0).

h) After the successful reception of a frame (reception without error up to the ACK slot and the successful sending of the ACK bit), the receive error counter shall be decreased by 1, if it was between one (1) and one hundred and twenty-seven (127). If the receive error counter was zero (0), it shall stay at zero (0), and if it was greater than one hundred and twenty-seven (127), then it shall be set to a value between one hundred and nineteen (119) and one hundred and twenty-seven (127).

13.1.4.3 Transition between error-active and error-passive states

If the transmit error counter or the receive error counter of a node exceeds one hundred and twenty-seven (127) (carry condition in case of a 7-bit receive error counter) then the supervisor shall request the MAC sublayer to set the corresponding node into the error-passive state.

An error condition letting a node become error-passive shall cause the node to send an active error flag.

An error-passive node shall become error-active again when both the transmit error counter and the receive error counter are less than or equal to one hundred and twenty-seven (127) (see Figure 16).

When the receive error counter of a node exceeds the error-passive limit of one hundred and twenty-seven (127), further increments of this receive error counter shall be limited by the width of the counter. At the next successful reception of a frame (transition to error-active), the receive error counter shall be set to a value below the error-passive limit [see 13.1.4.2, h)].

13.1.4.4 Bus-off management

If the transmit error counter of a node is greater than two hundred and fifty-five (255) (carry condition in case of a 8-bit transmit error counter), then the supervisor shall request the PL to set the node into the bus-off state.

A bus-off state shall not have any influence on the bus. It shall neither send any frames nor send ACK, error frames, overload frames. Whether such a node receives frames from the bus depends on the implementation.

A node which is in the bus-off state may become error-active (no longer bus-off) with its error counters both set to zero (0) after having monitored one hundred and twenty-eight (128) occurrences of eleven (11) consecutive recessive bits on the bus (see Figure 16).

NOTE Existing implementations may be different. For example, 6.14 permits the switching off of transmitter and receiver; recovery may only start upon a user request. This is not covered by the conformance testing according to ISO 16845.

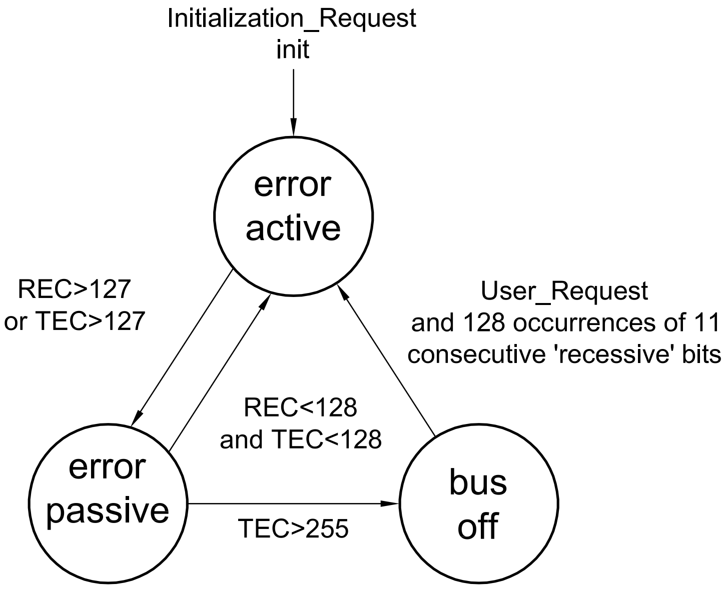

Figure 16 — Node status transition diagram

13.1.5 System start-up

If during system start-up only one node is on-line and if this node transmits some frame, it shall not get ACK, detect an error and repeat the frame. According to 13.1.4.2, c), 1), it may become error-passive but not bus-off.

A node that had been switched off shall run through a start-up routine in order to

a) synchronize with already available nodes before starting to transmit — synchronization is achieved when eleven (11) recessive bits are equivalent to

— ACK delimiter + EOF + Intermission, or

— error/overload delimiter + intermission detection, and

b) wait for other nodes without becoming bus-off if there is no other node available at the moment.

13.2 Bus failure management

During normal operation, several bus failures may occur that influence the bus operation. These failures and the resulting behaviour of the network shall be in accordance with ISO 11898-2.

Bibliography

[1] ISO 7637-3:1995, *Road vehicles — Electrical disturbance by conduction and coupling — Part 3: Vehicles with nominal 12 V or 24 V supply voltage — Electrical transient transmission by capacitive and inductive coupling via lines other than supply lines*

ICS 43.040.15

Price based on 45 pages