



**International
Standard**

ISO 11898-1

Road vehicles — Controller area network (CAN) —

**Part 1:
Data link layer and physical coding sublayer**

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

Partie 1: Couche liaison de données et sous-couche de codage physique

**Third edition
2024-05**



COPYRIGHT PROTECTED DOCUMENT

© ISO 2024

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword	v
Introduction	vi
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Symbols and abbreviated terms	6
5 Basic concepts of CAN	8
5.1 CAN properties	8
5.2 Frame transmissions	9
5.3 Bus access method	11
5.4 Information routing	11
5.5 Network flexibility	11
5.6 Remote data request	11
5.7 Error detection	11
5.8 Error signalling and recovery time	11
5.9 Fault confinement	12
5.10 Error-active	12
5.11 Error-passive	12
5.12 Bus-off	12
5.13 ACK	12
5.14 Repetition of transmission attempts	12
5.15 Network-wide data consistency	12
5.16 Switchable operating modes of the PMA	13
5.17 Bus states and MAC sub-layer phases	13
6 CAN DLL specification	14
6.1 General	14
6.2 Time stamping	14
6.3 DLL protocol	14
6.4 LLC sub-layer	15
6.4.1 Overview	15
6.4.2 Notifications	16
6.4.3 Structure of LLC frames	16
6.4.4 Limited LLC frames	17
6.4.5 Services of LLC sub-layer	17
6.5 Functions of the LLC sub-layer	21
6.5.1 General	21
6.5.2 Flow control on re-arbitration	21
6.5.3 Flow control on retransmission	21
6.5.4 Frame acceptance filtering	22
6.5.5 Overload notification	22
6.5.6 Recovery management	22
6.5.7 Time stamping	22
6.6 MAC sub-layer	23
6.6.1 Functions and rules	23
6.6.2 Services of the MAC sub-layer	23
6.6.3 Time reference points	23
6.6.4 Functional model of MAC sub-layer architecture	23
6.6.5 Specification of EF	27
6.6.6 Specification of OF	28
6.6.7 Inter-frame space specification	28
6.6.8 SOF	30
6.6.9 Elements of the MAC frame	30
6.6.10 MAC frame in CBFF and CEFF	31

6.6.11	MAC frame in FBFF and FEFF	33
6.6.12	MAC frame in XLFF	37
6.6.13	MAC frame coding	42
6.6.14	Data frame acknowledgement	43
6.6.15	Frame validation	43
6.6.16	Order of bit transmission	43
6.6.17	Medium access method	43
6.6.18	MAC data consistency	45
6.6.19	Restricted operation	45
6.6.20	Bus monitoring	45
6.6.21	Error handling and overload handling	45
7	PL specifications	49
7.1	General and functional modelling	49
7.2	Services of the PCS interface	50
7.2.1	Requirements	50
7.2.2	PCS_Data.Request	50
7.2.3	PCS_Data.Indicate	50
7.2.4	PCS_Mode.Request	50
7.2.5	PCS_Status.Transmitter	51
7.2.6	PCS_Status.Receiver	51
7.3	PCS	51
7.3.1	Bit encoding/decoding	51
7.3.2	Bit time	51
7.3.3	Configuration of the bit time parameters	55
7.3.4	Transmitter delay compensation	56
7.3.5	Synchronization	58
7.3.6	Tolerance range of the oscillator frequencies	60
7.4	Attachment unit interface	61
7.4.1	General	61
7.4.2	PCS to PMA symbols	61
7.4.3	PMA to PCS symbol	62
7.5	PWM encoding	62
7.5.1	General function and definitions	62
7.5.2	PCS without PWM encoding	63
7.5.3	PCS sub-layer with PWM encoding	63
8	Description of supervisor FCE	66
8.1	Fault confinement	66
8.1.1	Objectives	66
8.1.2	Strategies	66
8.1.3	Fault confinement interface specification	66
8.1.4	Rules of fault confinement	69
8.1.5	Node start-up	71
8.2	Bus failure management	71
Annex A (normative) CAN FD light — Data link layer and physical coding sub-layer requirements of responder nodes	72	
Annex B (informative) Configuration interface	80	
Annex C (informative) Additional information	81	
Bibliography	82	

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 31, *Data communication*.

This third edition cancels and replaces the second edition (ISO 11898-1:2015), which has been technically revised.

The main changes are as follows:

- CAN XL requirements added;
- CAN FD light protocol ([Annex A](#)) requirements added;
- editorial corrections.

A list of all parts in the ISO 11898 series can be found on the ISO website.

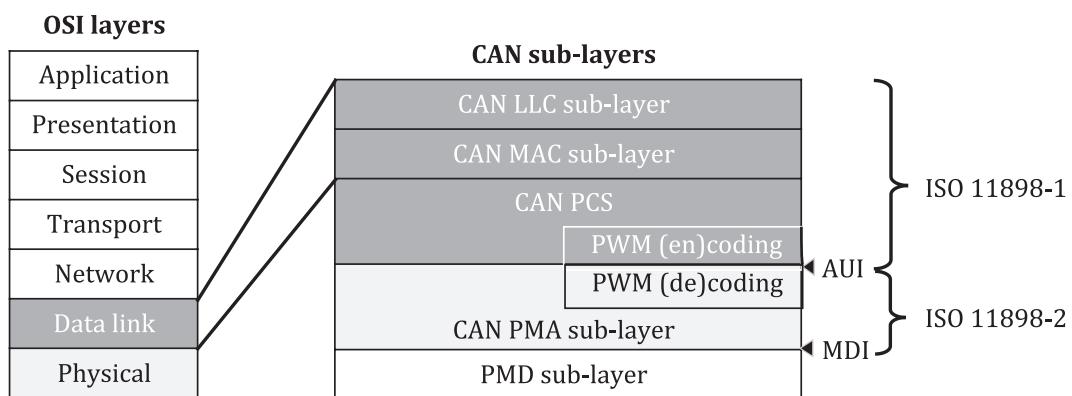
Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

The ISO 11898 series provides requirement specifications for the controller area network (CAN) data link layer and physical layer. It is intended for chip implementers, e.g. this document can be used for CAN protocol controllers and ISO 11898-2 for CAN transceivers. The CAN data link layer models the open systems interconnection (OSI) data link layer; it is internally subdivided into logic link control (LLC) and medium access control (MAC). This document also specifies the physical coding sub-layer (PCS) by means of the attachment unit interface (AUI). The PCS also provides the pulse-width modulation (PWM) encoding to be linked to a CAN SIC XL transceiver, which provides the PWM decoding.

The OSI layers above the data link layer (e.g. the network layer) are not specified in the ISO 11898 series of specifications.

[Figure 1](#) shows the relationship between the OSI layers and the CAN sub-layers.



Key

AUI	attachment unit interface
LLC	logic link control
MAC	medium access control
MDI	medium dependent interface
PCS	physical coding sub-layer
PMA	physical medium attachment
PMD	physical medium dependent
PWM	pulse-width modulation

Figure 1 — CAN data link and physical sub-layers in relation to the OSI model

Road vehicles — Controller area network (CAN) —

Part 1: Data link layer and physical coding sublayer

1 Scope

This document specifies the controller area network (CAN) data link layer (DLL) and the physical coding sub-layer (PCS). The CAN DLL features data fields of up to 2048 byte when the CAN extended data field length (XL) frame format is used.

This document divides the CAN DLL into the logical link control (LLC) and the medium access control (MAC) sub-layers. The DLL's service data unit (SDU), which interfaces the LLC and the MAC, is implemented by means of the LLC frame. The LLC frame also features the service data unit type (SDT) and the virtual CAN channel identifier (VCID), which provide higher-layer protocol configuration and identification information. How the higher-layer functions are handled is not specified in this document. There are five implementation options:

- 1 support of the CAN classic frame format only, not tolerating the CAN flexible data rate (FD) frame format;
- 2 support of the CAN classic frame format and tolerating the CAN FD frame format;
- 3 support of the CAN classic frame format and the CAN FD frame format;
- 4 support of the CAN classic frame format, the CAN FD frame format and the CAN XL frame format;
- 5 support of the CAN FD frame format for CAN FD light responders ([Annex A](#)).

NOTE Nodes of the first option can communicate with nodes of the third and fourth option when only the CAN classic frame format is used. Nodes of the first option cannot communicate with nodes of the fifth option: any attempt at communication generates error frames. Therefore, new designs implementing the fourth option can communicate with all other nodes.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

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

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1

arbitration mode

operating mode of the physical coding sub-layer (PCS) in which it is allowed that dominant bits can overwrite recessive bits

3.2**arbitration phase**

phase in which the *nominal bit time* (3.42) is used

3.3**attachment unit interface****AUI**

interface between the physical coding sub-layer (PCS) and the physical medium attachment (PMA) sub-layer

3.4**bit stuffing**

frame coding method providing *bus state* (3.8) changes required for periodic resynchronization when using a *non-return-to-zero* (NRZ) (3.43) bit representation

Note 1 to entry: Two types of bit stuffing exist: dynamic bit stuffing and fixed bit stuffing. The *transmitter* (3.54) adds stuff bits into the outgoing bit stream and *receivers* (3.48) de-stuff the *data frames* (3.16) and the *remote frames* (3.49) i.e. the inverse procedure is carried out.

3.5**bus**

shared medium of any topology

3.6**bus comparator**

physical coding attachment comparator

electronic circuit converting analogue signals used for transfer across the communication medium back into digital signals

3.7**bus driver**

electronic circuit converting digital signals into analogue signals so that these signals can be transferred across the communication medium, i.e. the *bus* (3.5)

3.8**bus state**

one of two complementary logical states: dominant or recessive

Note 1 to entry: The dominant state represents the logical 0, and the recessive state represents the logical 1. This document uses the terms "dominant" and "recessive" for the bit values of the medium access control (MAC) frame, independent of the *transceiver mode* (3.53). When no transmission is in progress, the *bus* (3.5) is *idle* (3.33). During idle time, it is in recessive state.

3.9**bus-off**

state of a *node* (3.39) in which it does not influence the *bus* (3.5)

3.10**classic base frame format****CBFF**

format for *data frames* (3.16) or *remote frames* (3.49) using an 11-bit *identifier* (3.31) and which are transmitted with one single bit rate and up to and including eight data bytes

3.11**classic extended frame format****CEFF**

format for *data frames* (3.16) or *remote frames* (3.49) using a 29-bit *identifier* (3.31) and which are transmitted with one single bit rate and up to and including eight data bytes

3.12**classic frame**

data frame (3.16) or *remote frame* (3.49) using the *CC base frame format* (3.10) or the *classic extended frame format* (3.11)

3.13**commander node**

node (3.39) that sends data frames to *responder nodes* (3.50) to initiate a controller area network (CAN) *FD light* (3.26) communication

3.14**data bit rate**

number of bits per time during *data phase* (3.17), independent of bit encoding/decoding

3.15**data bit time**

duration of one bit in *data phase* (3.17), defined by a number of data time quanta in the bit

3.16**data frame****DF**

frame (3.28) containing application content

3.17**data phase**

phase in which the *data bit time* (3.15) is used

3.18**data RX mode**

operating mode of the physical medium attachment (PMA) sub-layer in which the *bus states* (3.8) can be different from the bus states in the *arbitration mode* (3.1)

3.19**data TX mode**

operating mode of the physical medium attachment (PMA) sub-layer in which it can drive the *bus states* (3.8) differently than it drives them in the *arbitration mode* (3.1)

3.20**edge**

difference in *bus states* (3.8) between two consecutive time quanta

3.21**error frame****EF**

frame (3.28) indicating the detection of an error condition

3.22**FD base frame format****FBFF**

format for *data frames* (3.16) using an 11-bit *identifier* (3.31), which can be transmitted with a flexible bit rate

3.23**FD enabled**

able to receive and to transmit *FD frames* (3.25) as well as *CC frames* (3.12)

3.24**FD extended frame format****FEFF**

format for *data frames* (3.16) using a 29-bit *identifier* (3.31), which can be transmitted with a flexible bit rate

3.25**FD frame**

data frame (3.16) using the *FD base frame format* (3.22) or *FD extended frame format* (3.24)

3.26**FD light**

implementation option for a *responder node* (3.50) that is based on a subset of the controller area network (CAN) functionality.

3.27**FD tolerant**

not able to receive or to transmit *FD frames* (3.25), but not disturbing them

3.28**frame**

protocol data unit of the data link layer specifying the arrangement and meaning of bits or bit fields in the sequence of transfer across the transmission medium

3.29**handle**

label of one or multiple logical link control (LLC) frames, or data link layer service data units (LSDU), the data link layer (DLL)'s interface data coming from the higher open systems interconnection (OSI) layers (network layer or transport layer)

3.30**higher-layer protocol****HLP**

protocol (3.46) above the data link layer protocol, e.g. transport layer protocol or network layer protocol according to the open system interconnection model

3.31**identifier****ID**

unique label reflecting the *priority* (3.45) of a particular *frame* (3.28)

3.32**identifier-based arbitration**

carrier sense multiple access/collision resolution arbitration procedure resolving *bus* (3.5) contention when multiple *nodes* (3.39) simultaneously access the bus

3.33**idle**

operating condition of the *bus* (3.5) after the completion of a *frame* (3.28) until the next frame starts

3.34**idle condition**

detection of a consecutive sequence of 11 sampled recessive bits

3.35**integrating**

status of a *node* (3.39) waiting on an *idle condition* (3.34) after starting the protocol operation during *bus-off* (3.9) recovery or after a *protocol exception event* (3.47)

3.36**minimum time quantum**

smallest time quantum that can be configured for the specific *node* (3.39)

3.37**multi master**

network with several nodes where every *node* (3.39) is allowed to start sending a *frame* (3.28) when the controller area network (CAN) *bus* (3.5) is *idle* (3.33)

3.38**multicast**

address method transmitting a single *frame* (3.28) to a group of *nodes* (3.39)

Note 1 to entry: A broadcast is a special case of multicast, whereby a single frame is addressed to all nodes simultaneously.

3.39**node**

assembly, linked to a communication network, capable of communicating across the network according to a communication protocol specification (a node can be in one of four states: *integrating* (3.35), *idle* (3.33), *receiver* (3.48), or *transmitter* (3.54))

Note 1 to entry: A node operating in a controller area network (CAN) is called a CAN node.

3.40**node clock**

time base to coordinate the bit-time-related state machines in controller area network (CAN) *nodes* (3.39)

3.41**nominal bit rate**

number of bits per time during an *arbitration phase* (3.2), independent of the bit-encoding/decoding

3.42**nominal bit time**

duration of one bit in an *arbitration phase* (3.2), defined by a number of nominal time quanta in the bit

3.43**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* (3.20)

3.44**overload frame****OF**

frame (3.28) indicating an overload condition

3.45**priority**

attribute to a *data frame* (3.16) and to a *remote frame* (3.49) controlling its ranking during the arbitration

Note 1 to entry: A high priority increases the probability that a data frame or a remote frame wins the arbitration process. Further details are given in 6.6.17.5.

3.46**protocol**

formal set of conventions or rules for the exchange of information between *nodes* (3.39), including the specification of frame administration, frame transfer and physical medium attachment (PMA)

3.47**protocol exception event**

either exception from the formal set of conventions or rules to be able to tolerate future new frame formats, or reaction to errors when controller area network (CAN) XL is used with error signalling disabled

3.48**receiver**

node (3.39) that, while the *bus* (3.5) is not *idle* (3.33), is neither transmitting nor *integrating* (3.35)

3.49**remote frame****RF**

frame (3.28) that requests the transmission of a dedicated *data frame* (3.16)

3.50**responder node**

node (3.39) that is controlled by a *commander node* (3.13) using controller area network (CAN) *FD light* (3.26) communication

3.51**stuff bit count****SBC**

number of stuff bits in a *frame* (3.28) before the cyclic redundancy check (CRC) field, not including fixed stuff bits

3.52**transceiver**

electronic circuit, implementing the physical medium attachment (PMA) sub-layer, that connects controller area network (CAN) *nodes* (3.39) to a CAN bus (3.5) consisting of a *bus comparator* (3.6) and a *bus driver* (3.7)

3.53**transceiver mode**

operating mode of the physical medium attachment (PMA) sub-layer

3.54**transmitter**

node (3.39) sending a *data frame* (3.16) or a *remote frame* (3.49)

Note 1 to entry: A node remains a transmitter until the *bus* (3.5) is *idle* (3.33) again or until the node loses arbitration.

3.55**XL frame**

data frame (3.16) using the *XL frame format* (3.56)

3.56**XL frame format****XLFF**

format for *data frames* (3.16) using an 11-bit *identifier* (3.31), including up to 2048 data bytes, where the bit rate is switched at the beginning and at the end of the *data phase* (3.17)

4 Symbols and abbreviated terms

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

ACK	acknowledgement
ADH	arbitration to data high
ADS	arbitration to data sequence
AF	acceptance field
AH1	arbitration high 1
AH2	arbitration high 2
AL1	arbitration low 1
BCH code	Bose-Chaudhuri-Hocquenghem code

BRS	bit rate switch
CAN	controller area network
CC	classic
CRC	cyclic redundancy check
DAH	data arbitration high
DAS	data to arbitration sequence
DF	data frame
DH	data high bit
DL	data low bit
DLC	data length code
DLL	data link layer
EOF	end of frame
ESI	error state indicator
FCE	fault confinement entity
FCP	format check pattern
FCRC	frame CRC
FD	flexible data rate
FDF	flexible data rate format indicator
FTYP	frame type
IDE	identifier extension
IPT	information processing time
LLC	logical link control
LME	layer management entity
LPDU	logical link control protocol data unit
LSB	least significant bit
LSDU	logical link control service data unit
MAC	medium access control
MPDU	medium access control protocol data unit
MSB	most significant bit
nCRC	order of the generated polynomia
NRZ	non-return-to-zero

OF	overload frame
OSI	open systems interconnection
PCRC	preface cyclic redundancy check
PCS	physical coding sub-layer
PDU	protocol data unit
PL	physical layer
PMA	physical medium attachment
PMD	physical medium dependent
PWM	pulse width modulation
PWML	pulse width modulation long phase
PWMO	pulse width modulation offset time
PWMS	pulse width modulation short phase
resXL	reserved bit extended data field length format
RRS	remote request substitution
RTR	remote transmission request
SDT	service data unit type
SDU	service data unit
SEC	simple/extended content
SIC	signal improvement capability
SJW	synchronization jump width
SOF	start of frame
SSP	secondary sample point
t_q	time quantum/time quanta
$t_{q,min}$	minimum time quantum
VCID	virtual controller area network channel identifier
XL	extended data field length
XLF	extended data field length format indicator

5 Basic concepts of CAN

5.1 CAN properties

A CAN has the following properties:

- multi-master priority-based bus access;

- non-destructive content-based arbitration;
- all frame transfers are done as broadcast;
- multicast frame transfer by acceptance filtering;
- remote data request;
- configuration flexibility;
- error detection and error signalling;
- distinction between temporary errors and permanent failures of nodes and autonomous switching-off of defective nodes;
- retransmission of frames that lose bus arbitration, are not acknowledged, or are invalidated by errors;
- network-wide data consistency.

The following properties can be omitted by configuration: retransmission, error signalling, network-wide data consistency and remote data request.

A subset of the CAN FD data link layer may be implemented as a CAN FD light responder node. [Annex A](#) shall be followed when a CAN FD light responder node is implementing a subset of the CAN FD data link layer.

Configuration options for CAN implementations are described in [Annex B](#).

The differences between implementation options are described in [Annex C](#).

5.2 Frame transmissions

The DLL sends different frames: data frames (DF), remote frames (RF), error frames (EF) and overload frames (OF). When the bus is idle, a CAN node may start the transmission of a DF or an RF. The bus is idle when no frame is transmitted. A node starts to send an EF or an OF when an error condition or an overload condition, respectively, is detected by the node.

[Figure 2](#) shows the DFs and RFs specified in this document. This figure is also provided in electronic form at: <https://standards.iso.org/iso/11898/-1/ed-3/en>.

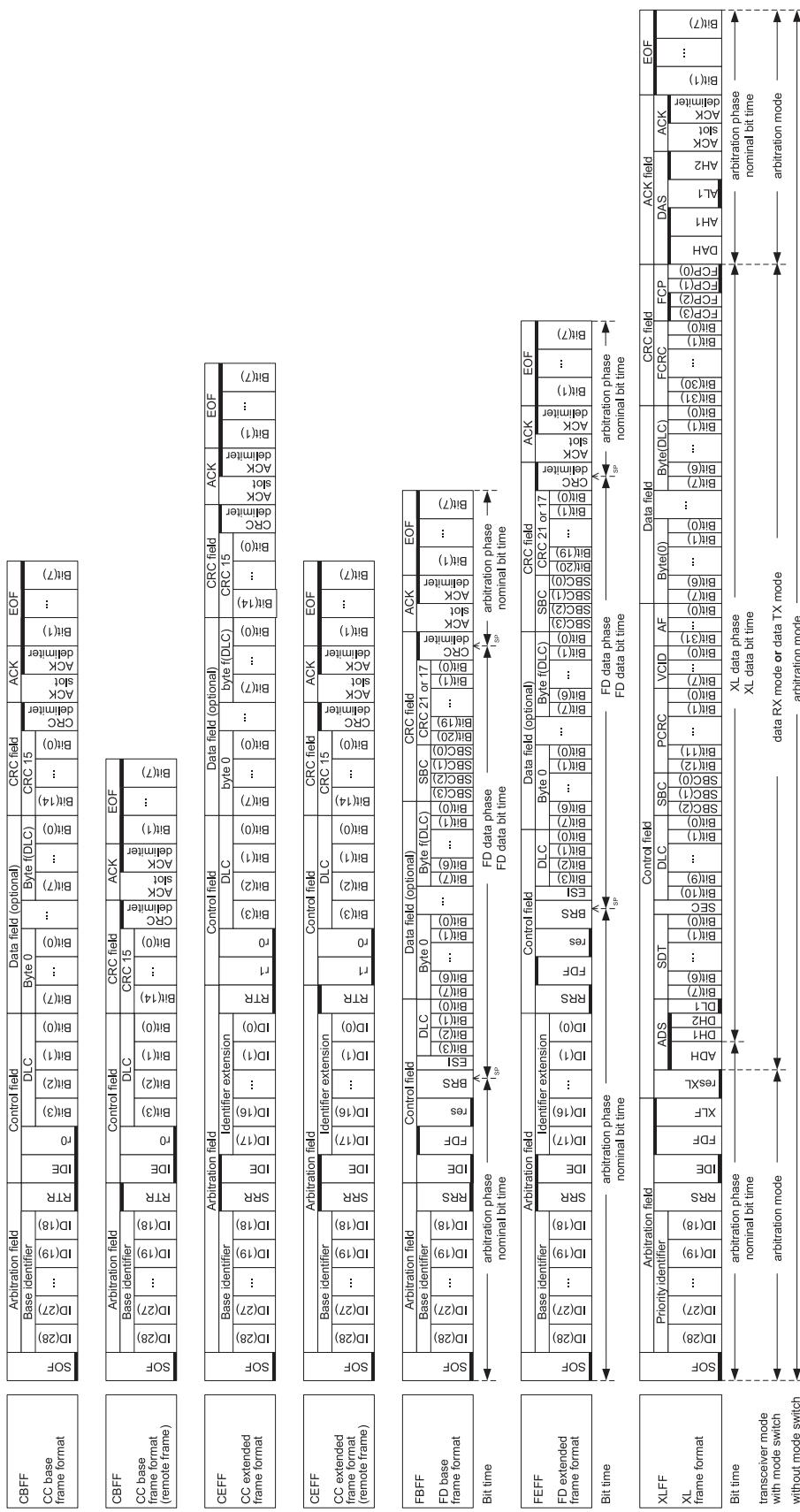


Figure 2 — CAN frame formats

NOTE Whether the FDF bit in FBFF and FEFF take part in arbitration or not, is specified in [6.6.11](#).

5.3 Bus access method

If two or more nodes start to transmit DFs or RFs at the same time, the bus access conflict is resolved by arbitration using the identifier. The mechanism of arbitration ensures that neither information nor time is lost. The transmitter with the DF or RF of the highest priority gains the bus access. A DF with the same ID as an RF wins bus arbitration.

5.4 Information routing

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.

5.5 Network flexibility

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

5.6 Remote data request

By sending an RF, a node requiring data requests another node to send the corresponding DF: the RF and the corresponding DF are named by the same identifier and using the same DLC.

NOTE 1 The node providing the remotely requested DF (same ID) decides whether the updated data is mapped into the DF or the data in the transmit buffer is sent.

NOTE 2 The node providing the remotely requested DF (same ID) decides how to respond to an RF with mismatching DLC.

5.7 Error detection

For detecting errors, the following measures are provided:

- monitoring (transmitters compare the transmitted bit levels with the bit levels detected on the network);
- 15-bit CRC for CC frames, 17-bit CRC for FD frames with up to 16 data-field bytes, 21-bit CRC for FD Frames with 20 to 64 data-field bytes, 13-bit PCRC and 32-bit frame CRC for XL frames;
- stuff bit count check for FD frames and XL frames;
- dynamic bit stuffing with a stuff width of five (except in the CRC field of FD frames and after the arbitration field of XL frames);
- fixed bit stuffing with a stuff rate of 5 (used in the FD CRC field);
- fixed bit stuffing with a stuff rate of 11 (used in the XL data phase);
- frame format check;
- ACK check.

5.8 Error signalling and recovery time

Corrupted frames are flagged by transmitting nodes and by error-active receiving nodes. Such frames are aborted and retransmitted according to the implemented recovery procedure (see [6.6.21](#)). The recovery time from detecting an error until the possible start of the next frame is 17 to 23 nominal bit times (in the case of nodes in error passive mode up to 31 nominal bit times) if there are no further errors.

For CAN XL nodes, error signalling depends on the node configuration. It can be enabled or disabled as specified in [6.6.21.1](#). If error signalling is disabled, the node does not send EFs or OFs. The node reacts on error conditions and overload conditions as specified in [6.6.21.4](#).

NOTE Nodes with disabled error signalling are not interoperable with CAN nodes with enabled error signalling.

5.9 Fault confinement

CAN nodes can distinguish short disturbances from permanent failures. Defective transmitting nodes are switched off. Switched off means a node is logically disconnected from the bus, so that it can neither send nor receive frames (see [8.2](#)).

5.10 Error-active

An error-active node takes part in network communication and sends an active error flag when an error is detected. The active error flag consists of six consecutive dominant bits and violates the rule of bit-stuffing and all fixed formats appearing in a DF and RF (see [8.2](#)).

5.11 Error-passive

An error-passive node sends no active error flag. It takes part in network communication, but when an error is detected a passive error flag is sent. The passive error flag consists of six consecutive recessive bits. After transmission, an error-passive node waits additional time before initiating a further transmission (see suspend transmission in [6.6.7.4](#) and [8.2](#)).

Nodes with disabled error signalling do not enter error-passive state.

5.12 Bus-off

A node is in the bus-off state when it is switched off from the bus due to an FCE request. In the bus-off state, a node neither sends nor receives frames. In the bus-off state, a node does not send any dominant bits.

Nodes with disabled error signalling do not enter bus-off state.

5.13 ACK

Receivers check the consistency of the received DFs and RFs, acknowledge a consistent frame and, when error signalling is enabled, flag an inconsistent frame by means of an EF.

5.14 Repetition of transmission attempts

The transmission of a DF or RF losing arbitration is automatically restarted (re-arbitration). The number of automatic retransmissions of DFs or RFs that are not transmitted successfully by any reason, except the loss of bus arbitration, can be configurable to a dedicated number. That number can reach from zero to unlimited, while unlimited means that retransmissions are only limited by the error counting mechanisms.

NOTE CiA 612-1 gives system design recommendations on how to configure the number of retransmissions for the case, when error signalling is disabled.

5.15 Network-wide data consistency

Network configuration decides whether nodes signal the errors detected in frames. When error signalling is enabled within a network, a frame can be simultaneously accepted as a valid frame either by all nodes or by no node. Thus, data consistency is a property of the network achieved by the concepts of the broadcast and by the error handling. When error signalling is disabled within a network, local errors can prevent specific nodes from receiving frames that are received by other nodes.

5.16 Switchable operating modes of the PMA

While the CAN XL communication is possible using CAN transceivers compliant with ISO 11898-2, higher data bit rates require CAN transceivers that can be switched into a dedicated operating mode for the XL data phase. When that operating mode does not allow the dominant bus state to overwrite all other bus states, the error signalling is disabled by the node configuration. CAN XL nodes provide the PWM encoding function as specified in [7.5](#) and the PWM decoding function as specified in ISO 11898-2 to switch the operating modes of the CAN transceivers.

The PCS indicates the following CAN transceiver modes to the PMA sub-layer if the CAN transceiver mode switching is enabled:

- arbitration mode;
- data TX mode: active in the transmitter during ADH bit and XL data phase;
- data RX mode: active in the receiver during ADH bit and XL data phase.

[Table 1](#) links the CAN transceiver mode names used in this document at the MAC sub-layer to the names used in ISO 11898-2 at the PMA sub-layer.

Table 1 — CAN transceiver modes signalled to the PMA

CAN transceiver modes signalled to the PMA sub-layer	PMA operating modes for CAN SIC XL transceivers as specified in ISO 11898-2
Arbitration mode	SIC mode
Data TX mode	FAST TX mode
Data RX mode	FAST RX mode

NOTE Error signalling enabled combined with CAN transceiver mode switching enabled is not compatible with the PMA sub-layer specified in ISO 11898-2.

5.17 Bus states and MAC sub-layer phases

The MAC sub-layer distinguishes the following three phases, as shown in [Figure 2](#):

- arbitration phase, in which the nominal bit time is used;
- FD data phase, in which the FD data bit time is used;
- XL data phase, in which the XL data bit time is used.

This document uses the terms dominant (0) and recessive (1) for the bit values of the MAC frame, independent of the CAN transceiver mode. FD data bit time and XL data bit time may be identical.

If the CAN transceiver mode switching is enabled, the PMA sub-layer specified in ISO 11898-2 behaves in the following way. The CAN SIC XL transceiver of the transmitter sends the bus state level_1 instead of recessive and the bus state level_0 instead of dominant during the data TX mode. If the CAN transceiver mode switching is enabled, the CAN SIC XL transceiver of the receiver does not drive the bus during the data RX mode.

NOTE The bus states level_1 and level_0 do not overwrite each other. If two nodes are driving concurrently level_1 and level_0, receiving nodes detect unpredictably either level_1 or level_0.

6 CAN DLL specification

6.1 General

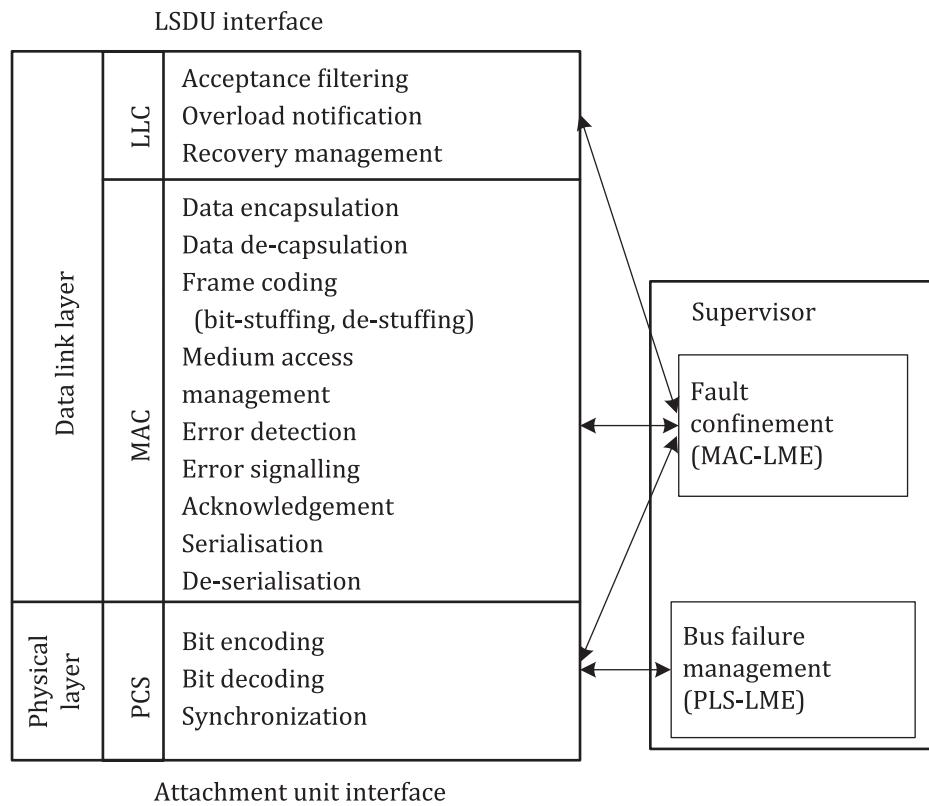


Figure 3 — Layered architecture of CAN

The DLL as shown in [Figure 3](#), is divided into the two sub-layers: the LLC sub-layer (responsible for flow control, recovery management and time-stamping) and the MAC sub-layer (responsible for data encapsulation, data de-capsulation, frame coding, bit-stuffing, de-stuffing, serialization, de-serialization, error detection, error signalling and acknowledgement).

The DLL has a cross-layer function interface to the FCE. The MAC sub-layer operations are supervised by the FCE. Fault confinement is a self-checking mechanism, as specified in [Clause 8](#), that distinguishes short disturbances from permanent failures.

6.2 Time stamping

The DLL may have a cross-layer function interface to a time-base for the time-stamping. When the time-stamping is not supported, time-base and time-stamps shall be treated as constant with the value 0.

6.3 DLL protocol

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

An (N)-layer protocol data unit (PDUN) consists of (N)-layer specific protocol control information (PCIN) and (N)-layer user data. PDUN shall be passed to a (N-1)-layer entity via an SAPN-1. The PDUN shall be passed by means of the SDUN-1 to the (N-1)-layer, the services of which allow the transfer of the PDUN. 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 provide neither means for mapping one SDU into multiple PDUs nor means for mapping multiple SDUs into one PDU, i.e. a PDUN is directly constructed from the associated SDUN and the layer specific control information PCIN. [Figure 4](#) illustrates the data link sub-layer interactions.

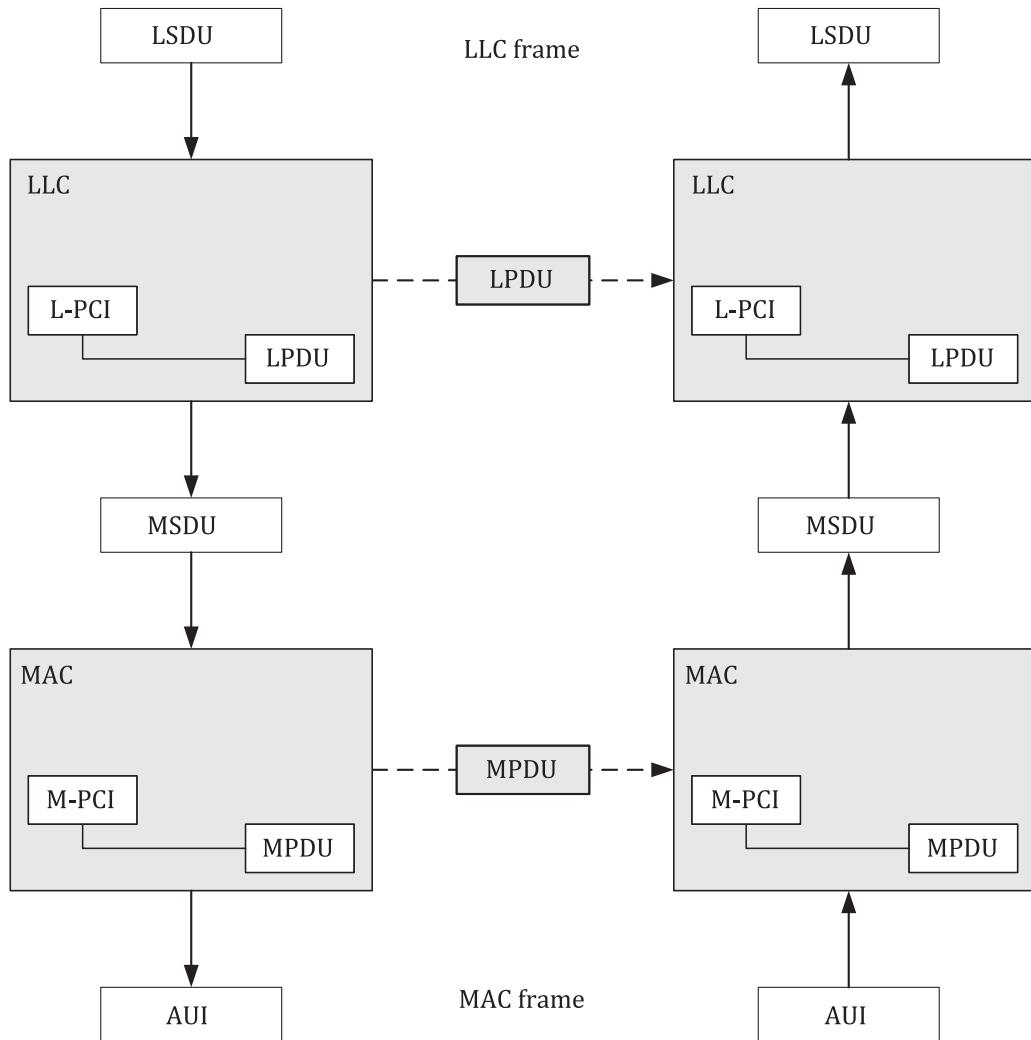


Figure 4 — Protocol layer interactions

The DLL's SDU is called the LLC frame; it also interfaces between the LLC sub-layer and the MAC sub-layer.

The DLL's PDU is called the MAC frame. This is the bit sequence seen on the CAN bus. The MAC frame may be in CBFF, CEFF, FBFF, FEFF, or XLFF.

6.4 LLC sub-layer

6.4.1 Overview

The LLC sub-layer describes the upper part of the DLL according to ISO/IEC 8802-2. It is associated with those protocol issues that are independent of the type of the medium access method.

The LLC sub-layer shall provide the following functions:

- flow control;
- recovery management;
- time-stamping.

The LLC sub-layer shall offer two types of connectionless transmission services to the LLC user, as specified in [6.4.5.2](#).

The interface service data sent from or to the user shall be as specified in [6.4.3](#).

6.4.2 Notifications

The notifications sent between the LLC user and the LLC sub-layer are specified in [Table 2](#) and in [Table 3](#).

Table 2 — Notification sent from LLC user to LLC sub-layer

Notification	Description
Reset_Request	Request to set the node into an initial state

Table 3 — Notification sent from LLC sub-layer to LLC user

Notification	Description
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.

6.4.3 Structure of LLC frames

LLC frames are SDUs between higher OSI layers and LLC entities. [Table 4](#) specifies the LLC frame.

The LLC frame structure shall contain all content needed for all CAN frame formats and types, including the selection of a specific CAN frame format. In the interaction between the LLC sub-layer and the MAC sub-layer, the content of that parts of the LLC frame that are not used for the selected CAN frame format shall be ignored. Only LLC frames with the LLC format specified in [Table 4](#) can be transferred to the LLC sub-layer.

Table 4 — LLC frame

Field	Size	Description
ID	(11 +18) bit	Priority identifier: ID(28) to ID(18) are used for all frames with 11-bit identifier format (CBFF, FBFF, XLFF) while ID(28) to ID(0) are used for all frames with 29-bit identifier format (CEFF, FEFF).
Format	3 bit	The frame format is coded by the three bits IDE, FDF, and XLF: <ul style="list-style-type: none"> — if 000_b the MAC frame format is CBFF; — if 100_b the MAC frame format is CEFF; — if 010_b the MAC frame format is FBFF; — if 110_b the MAC frame format is FEFF; — if 011_b the MAC frame format is XLFF. Other bit values do not code a valid frame format.
FTYP	1 bit	In CBFF and CEFF, the FTYP bit corresponds to the MAC frame's RTR bit, this means that FTYP distinguishes between the frame types RF (FTYP = 1 _b) and DF (FTYP = 0 _b). In FBFF and FEFF, the FTYP bit is ignored for transmission but corresponds to the MAC frame's RRS bit for reception. In XLFF, the FTYP bit corresponds to the MAC frame's RRS bit.
BRS	1 bit	Bit rate switch: only used in FBFF and FEFF frames.
ESI	1 bit	Error state indicator: only used in FBFF and FEFF frames.
SDT	1 byte	SDU type: describes the structure of the frame's data field content, only used in XLFF frames.
SEC	1 bit	Simple/extended content: flags if the data field contains simple or extended content. This is only used in XLFF frames.
DLC	(7+4) bit	Data length code: 7 MSB only used in XLFF frames. For XLFF frames the numbers 0 to 2047 code 1 to 2048 data bytes. For DFs of all other formats the 4 LSB encode the number of data bytes as specified in Table 5 .

Table 4 (continued)

Field	Size	Description
VCID	1 byte	Virtual CAN channel ID: only used in XLFF frames.
AF	4 byte	Acceptance field: only used in XLFF frames.
LLC data	0 to 2048 byte	Payload of the LLC frame

Table 5 — DLC: coding of the four LSB

DLC value [decimal]	CBFF and CEFF [byte]	FBFF and FEFF [byte]
0	0	0
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	8	12
10	8	16
11	8	20
12	8	24
13	8	32
14	8	48
15	8	64

6.4.4 Limited LLC frames

A subset of the complete range of identifiers and DLCs may be implemented.

If an LLC sub-layer is restricted to the use of a sub-range of identifiers (e.g. only 11-bit identifiers), then it shall be limited to LLC DFs and LLC RFs with identifiers of that sub-range (e.g. identifiers with their IDE bit set to logic 0 and their identifier extension is ignored).

If an LLC sub-layer is restricted to the use of less than the maximum number of data bytes, then it shall be limited to LLC DFs with several data bytes of that restricted range. If the DLC indicates a higher number of data bytes, the data bytes beyond the restricted range shall be replaced in the LLC DF by bytes with the value of CC_{HEX} (“padding”) both for received and for transmitted frames. The CAN node may support a configuration where the CAN node does not transmit a frame whenever the frame’s DLC indicates several data bytes beyond the restricted range.

NOTE The padding of the data field of received DFs does not need to be implemented in the LLC layer.

6.4.5 Services of LLC sub-layer

6.4.5.1 Service access points

In CAN nodes, the LLC’s service access point is the LLC storage unit, e.g. a memory buffer for the transmission or for the reception of a DF or RF.

6.4.5.2 Types of connectionless-mode transmission services

The LLC sub-layer shall offer two types of connectionless-mode transmission services.

- Unacknowledged data transfer service
This service provides means by which LLC users exchange LSDU without establishing a data link connection.
- Unacknowledged remote data request service
This service provides 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 serve the data request in the following two ways.

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

According to the two different LLC services, seven types of frames may be used for the communication between transmitting node and receiving nodes:

- LLC DF in CBFF;
- LLC DF in CEFF;
- LLC DF in FBFF;
- LLC DF in FEFF;
- LLC RF in CBFF;
- LLC RF in CEFF;
- LLC DF in XL frame format.

The LLC DF carries data from a transmitter to a receiver. The LLC RF requests transmission of a DF (with the same identifier) from a (single) remote node. In both cases, the LLC sub-layer shall notify the successful transmission or reception of a DF or RF to the LLC user.

6.4.5.3 Format description of service primitives

Service primitives are written as:

```
service.type(
    [parameter1, ]
)
```

where service indicates the name of the service, e.g. L_Data for data transfer;

 type indicates the type of the service primitives (see [6.4.5.4](#));

 [parameter1,] is the list of values passed to the service primitives.

The squared brackets indicate that this parameter list can be empty.

6.4.5.4 Types of services

There are four generic types of services, shown as follows:

- a) Service.Request The request service is passed from the (N)-user (service user) to the (N)-layer (service provider) to request initiation of the service.

- b) Service.AbortRequest The request service is passed from the (N)-user (service user) to the (N)-layer (service provider) to cancel a previously requested service initiation.
- c) Service.Confirm The confirm service is passed from the (N)-layer (or sub-layer) to the (N)-user to convey the results of one or more associated previous service request(s). This service can indicate either failure to comply or some level of compliance. It does not necessarily indicate any activity at the remote peer interface.
- d) Service.Indication The indication service is passed from the (N)-layer to the (N)-user to indicate an internal (N)-layer (or sub-layer) event, which is significant to the (N)-user. This event can be logically related to a remote service request or can be caused by an event internal to the (N)-layer (or sub-layer).

6.4.5.5 Service specification

6.4.5.5.1 General

[6.4.5.5](#) specifies in detail the LLC services and their associated parameters. [Table 6](#) defines the LLC services. [Table 7](#) defines the LLC service parameters.

Table 6 — LLC services overview

Service	Services	Description
Unacknowledged data transfer	L_Data.Request	Request for data transfer
	L_Data.AbortRequest	Request abortion of data transfer
	L_Data.Confirm	Confirmation of data transfer request
	L_Data.Indication	Indication of data transfer reception

Table 7 — LLC service parameters

Parameter	Description
Frame	LLC frame
Transfer_Status	Confirmation: ongoing, lost arbitration, transmitted, aborted, disturbed
Timestamp	Time value captured on trigger from MAC sub-layer
Handle	Identifies LLC storage unit used for transaction

6.4.5.5.2 L_Data.Request

Function

The L_Data.Request service shall be passed from the LLC user to the LLC sub-layer to request that an LLC frame is sent to one or more remote LLC entities.

Semantics of L_Data.Request service

The service shall provide parameters as follows:

```
L_Data.Request(
    Frame
    Handle
)
```

The LLC storage unit to be used for the transmission is identified by the handle.

Effect on receipt

Receipt of this service shall cause the LLC sub-layer to initiate the transfer of an LLC frame by use of the data transfer service provided by the MAC sub-layer (see [6.6](#)). Any L_Data.Request shall be processed not later than the second SOF after the request when there are no EFs present during this time.

6.4.5.5.3 L_Data.AbortRequest

Function

The L_Data.AbortRequest service shall be passed from the LLC user to the LLC sub-layer to abort a previous request of an LLC frame transmission.

Semantics of L_Data.AbortRequest service

The service request shall provide the following parameters:

```
L_Data.AbortRequest(
    Handle
)
```

The LLC storage unit for which the transmission shall be aborted is identified by the handle.

Effect on receipt

Receipt of this service shall cause the LLC sub-layer to abort the transfer of an LLC frame in the specified LLC storage unit. Pending transmissions, which are already passed to the MAC sub-layer, shall only be aborted if:

- an error in the MAC sub-layer during transmission occurs, or
- arbitration was lost in the MAC sub-layer,

which causes the LLC frame to wait for another transmission attempt.

This means that an abortion request shall be kept pending in the LLC layer until either one of the above situations occur, or until the transmission was completed.

The LLC sub-layer cannot abort transmissions that are already submitted to the MAC sub-layer (due to priority scheduling of the requested LLC storage unit indicated by the handle). Any L_Data.AbortRequest shall be processed prior to the second SOF after the request when there are no EFs present during this time. Any L_Data.AbortRequest shall be processed prior to the third SOF after the request when there are EFs present during this time.

6.4.5.5.4 L_Data.Confirm

Function

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

Semantics of L_Data.Confirm service

The service shall provide parameters as follows:

```
L_Data.Confirm(
    Transfer_Status
    Timestamp
    Handle
)
```

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

Transfer_Status: [Ongoing, Lost Arbitration, Transmitted, Aborted, Disturbed]

The timestamp is a 64-bit value, captured at one time reference point synchronized to MAC frame transmission. The timestamp is valid when the Transfer_Status is transmitted. Otherwise, the timestamp value is undefined.

Effect on receipt

The effect on receipt of this service by the LLC user is not in the scope of this document.

6.4.5.5.5 L_Data.indication

Function

The L_Data.Indication service shall be passed from the LLC sub-layer to the LLC user to indicate the arrival of the data of an LLC frame after the reception of a MAC frame.

Semantics of L_Data.Indication service

The service shall provide parameters as follows:

```
L_Data.Indication(  
    Frame  
    Timestamp  
)
```

Effect on receipt

The effect on receipt of this service by the LLC user is not in the scope of this document.

6.5 Functions of the LLC sub-layer

6.5.1 General

The LLC sub-layer provides the following functions:

- flow control;
- frame acceptance filtering;
- overload notification;
- recovery management;
- time-stamping.

6.5.2 Flow control on re-arbitration

The transmission of frames losing bus arbitration shall be restarted by the LLC entity unless the transmission of that frame is aborted by the LLC user. The restart of the transmission after the loss of arbitration is called a re-arbitration.

6.5.3 Flow control on retransmission

The restart of the unsuccessful transmission, except after the loss of arbitration, is called a retransmission. The retransmission of frames that are not aborted by the LLC user shall be attempted for the configured specific number of times. Retransmission attempts shall be counted in the retransmission counter. The node's retransmission counter shall be incremented by one at each retransmission when the node's preceding transmission attempt for the same frame was not successful for any reason except lost bus arbitration. The configured number shall be in the following range:

- from 0 (no retransmission),
- to at least 6 (6 retransmission attempts),

— and the highest number in this range may allow unlimited retransmission attempts.

The node's retransmission counter shall be reset to zero when a transmission is successful or when the LLC user requests a new transmission.

Nodes that do not support the XL frame format are not required to implement the retransmission counter, but they shall support unlimited retransmissions.

6.5.4 Frame acceptance filtering

Each receiver should decide by frame acceptance filtering whether the frame is relevant or not. Acceptance filtering is a frame transaction initiated at the LLC sub-layer, it should be a single, self-contained operation independent of previous frame transactions.

6.5.5 Overload notification

The transmission of a OF shall be initiated by the LLC sub-layer if internal conditions of a receiver require delay of the next LLC DF or LLC RF. If there are internal conditions of a CAN node that cause a OF to be initiated, these conditions shall be documented for that CAN node. In this case, the CAN node shall not generate more than two OFs.

6.5.6 Recovery management

The LLC sub-layer's transmission service shall report the Transfer_Status of a requested transmission as Ongoing when the MAC has started that transmission. It shall report the Transfer_Status as Transmitted when that transmission is successfully completed. It shall report the Transfer_Status as Lost_Arbitration when that transmission loses the bus arbitration, until the frame is started again or aborted. It shall report the Transfer_Status as Disturbed when that transmission is disturbed by an error or is not acknowledged, until the frame is started again or aborted. It shall report the Transfer_Status as Aborted when that transmission is not successful and is not retransmitted because the frame is aborted, or its retransmission limit is reached (see [Table 7](#)).

6.5.7 Time stamping

For the capturing of time stamps, the LLC may provide a cross-layer function interface to a time-base that is outside the DLL. The time-base and time-stamps shall have a length of 64 bit. When time-stamping is not supported, time-base and time-stamps shall be treated as constants with the value 0. The time resolution of the external time base is implementation-specific.

The time-base shall be captured at time reference points signalled by the MAC sub-layer (see [6.6.3](#)); the captured values shall be reported as time-stamps by the services L_Data.Confirm and L_Data.Indication. It shall be configurable whether the time reference point at the SOF recognition or at the position when the frame gets valid shall be used for capturing.

NOTE The position when the frame becomes valid depends on whether error signalling is enabled or disabled. When error signalling is enabled, the frame validation applies as specified in [6.6.15.2](#). When error signalling is disabled, the frame validation applies as specified in [6.6.15.3](#).

For the service L_Data.Confirm, a time-stamp shall be captured at the time reference point of the transmission of the MAC frame.

For the service L_Data.Indication, a time-stamp shall be captured at the time reference point of the reception of the MAC frame.

6.6 MAC sub-layer

6.6.1 Functions and rules

The MAC sub-layer represents the lower part of the OSI DLL. It services the interface to the LLC sub-layer and the PMA, and comprises the functions and rules related to:

- encapsulation/decapsulation of the transmit/receive data,
- error detection and signalling,
- management of the transmit/receive medium access.

6.6.2 Services of the MAC sub-layer

The MAC sub-layer shall provide services to the local LLC for:

- (MAC-) acknowledged transfer of LLC frames;
- time reference points synchronized to MAC frame reception;
- time reference points synchronized to MAC frame transmission;
- (MAC-) acknowledged transfer of OFs.

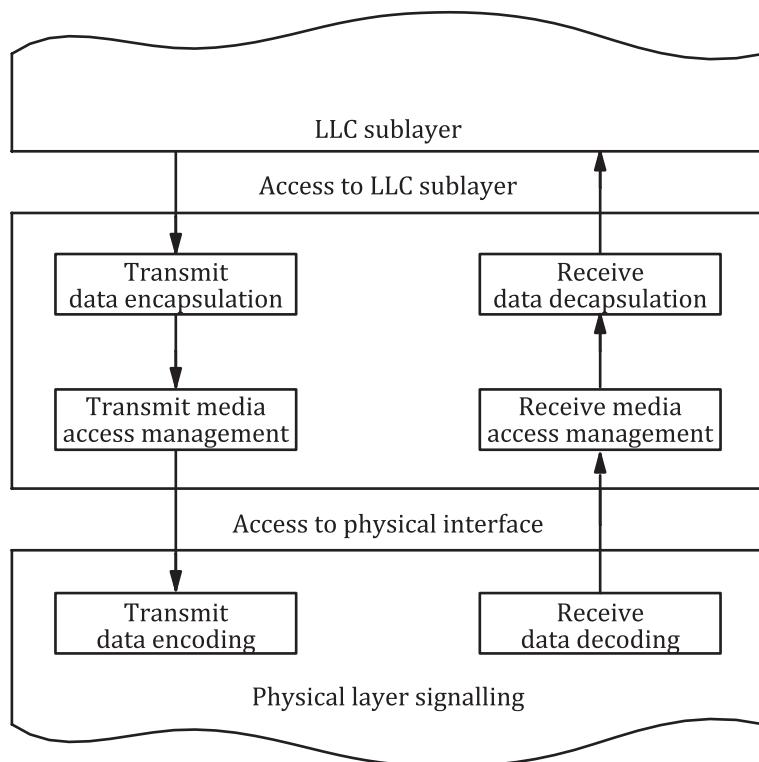
6.6.3 Time reference points

Any DF or RF received or transmitted by the MAC sub-layer shall cause a notification to be sent from the MAC sub-layer to the LLC sub-layer to signal a time reference point for the capture of the time-base as a timestamp. Time reference points shall be signalled both at the sample point of the SOF bit and at the sample point of the bit of the frame when that frame gets valid.

6.6.4 Functional model of MAC sub-layer architecture

6.6.4.1 Capability

The functional capabilities of the MAC sub-layer are described by use of the functional model specified in ISO/IEC/IEEE 8802-3. [Figure 5](#) shows the MAC function. In this model, the MAC sub-layer is divided into two independently operating parts, i.e. the transmit part and the receive part.

**Figure 5 — MAC function**

Data transmission and reception between nodes in a network is performed and controlled by four different frame types:

- a DF that carries data from a transmitter to all receivers;
- an RF transmitted by a node for requesting transmission of the DF with the same identifier;
- an EF transmitted by any node (transmitter or receiver) in case a bus error is detected;
- an OF used for providing an extra delay between the preceding and succeeding DFs or RFs.

DFs and RFs shall arbitrate for bus access and shall be separated from preceding frames by an inter-frame space.

There are five different DFs in CAN:

- DF in CC base frame format,
- DF in CC extended frame format,
- DF in FD base frame format,
- DF in FD extended frame format,
- DF in XL frame format.

There are two different RFs in CAN:

- RF in CC base frame format,
- RF in CC extended frame format.

6.6.4.2 MAC frame transmission

Frame transmission shall fulfil the following requirements.

Transmit data encapsulation:

- acceptance of LLC frames and interface control information,
- CRC sequence calculation including stuff bit count for FD frames and for XL frames,
- construction of MAC frames in the selected format by adding (as needed) SOF, RRS bit, IDE bit, RTR bit, SRR bit, FDF bit, XLF bit, resXL bit, BRS bit, ESI bit, ADS, PCRC, FCRC, FCP, DAS, ACK, and EOF to the LLC frame.

Transmit medium access management:

- initiation of the transmission process after recognizing bus idle (compliance with inter-frame space),
- serialization of the MAC frame,
- insertion of stuff bits (bit stuffing),
- arbitration and passing into receive mode in case of loss of arbitration,
- error detection (monitoring, format check),
- ACK checking,
- recognition of an overload condition,
- OF construction and initiation of transmission,
- EF construction and initiation of transmission,
- presentation of a serial bit stream to the PMA for transmission,
- switching of the bit rate for FD frames and for XL frames,
- signalling when the switching of the PMA operating mode for XL frames can occur.

6.6.4.3 MAC frame reception

Frame reception shall fulfil the following requirements.

Receive medium access management:

- reception of a serial bit stream from the PMA,
- switching of the bit rate for FD frames and for XL frames,
- signal when the switching of the PMA operating mode for XL frames should occur,
- deserialization and recompiling of the frame structure,
- deletion of stuff bits (bit de-stuffing),
- error detection (CRC, stuff bit count check, format check, stuff rule check),
- transmission of ACK,
- recognition of an overload condition,
- OF construction and initiation of transmission,
- EF construction and initiation of transmission.

Receive data de-capsulation:

- removing the MAC specific information from the received frame,
- presenting the LLC frame and interface control information to the LLC sub-layer.

6.6.4.4 MAC CRC calculation

DFs and RFs contain CRC sequences to enable the detection of disturbances during frame transmission. The frame check sequences shall be derived from a CRC (BCH code).

A CAN node uses different CRC generator-polynomials for different frame formats. The first polynomial, CRC_15, shall be used for CC frames. The second, CRC_17, shall be used for FD frames with a data field up to 16 byte long. The third, CRC_21, shall be used for FD frames with a data field longer than 16 byte. The fourth, CRC_13, shall be used for the arbitration and control fields of XL frames while the fifth, CRC_32, shall be used for the whole XL frame. Each polynomial results in a Hamming distance of six.

At the SOF, the five specified CRC sequences shall be calculated concurrently by the transmitting nodes and the receiving nodes. The node that wins the arbitration sends the CRC sequence selected by the values of the frame format and DLC. The receivers shall consider only the selected CRC polynomials to check for a CRC-error.

CRC_13	$39E7_{16}$	$(x^{13}+x^{12}+x^{11}+x^8+x^7+x^6+x^5+x^2+x^1+1)$ = $(x+1) \times (x^{12}+x^{10}+x^9+x^8+x^6+x^4+x^3+x^2+1)$
CRC_15	$C599_{16}$	$(x^{15}+x^{14}+x^{10}+x^8+x^7+x^4+x^3+1)$ = $(x+1) \times (x^7+x^3+1) \times (x^7+x^3+x^2+x^1+1)$
CRC_17	$3685B_{16}$	$(x^{17}+x^{16}+x^{14}+x^{13}+x^{11}+x^6+x^4+x^3+x^1+1)$ = $(x+1) \times (x^{16}+x^{13}+x^{10}+x^9+x^8+x^7+x^6+x^3+1)$
CRC_21	302899_{16}	$(x^{21}+x^{20}+x^{13}+x^{11}+x^7+x^4+x^3+1)$ = $(x+1) \times (x^{10}+x^3+1) \times (x^{10}+x^3+x^2+x^1+1)$
CRC_32	$1F4ACFB13_{16}$	$(x^{32}+x^{31}+x^{30}+x^{29}+x^{28}+x^{26}+x^{23}+x^{21}+x^{19}+x^{18}+x^{15}+x^{14}+x^{13}+x^{12}+x^{11}+x^9+x^8+x^4+x^1+1)$ = $(x+1) (x+1) (x^{15}+x^4+1) (x^{15}+x^{14}+x^{11}+x^6+x^4+x^3+x^2+x^1+1)$

There are different conventions how to represent a generator-polynomial in hexadecimal notation. The notation in this document includes the high-order bit.

The CRC_INIT_VECTOR shall be (0,..,0) for CRC_15.

The CRC_INIT_VECTOR shall be (1,0, ,0) for CRC_17 and for CRC_21; where the single 1 is at the most significant bit position.

The CRC_INIT_VECTOR shall be (0,0, ,1) for CRC_13 and for CRC_32; where the single 1 is at the least significant bit position.

The length of the CRC sequence (nCRC, the order of the generator-polynomial) is set to 13 for CRC_13, 15 for CRC_15, to 17 for CRC_17, to 21 for CRC_21, and to 32 for CRC_32.

The relevant bit streams for CRC calculation are specified in [6.6.10.5](#), [6.6.11.5](#), [6.6.12.3](#), and in [6.6.12.5](#), each supplemented with nCRC bits of value 0.

In order to carry out the CRC calculation, the polynomial to be divided is defined by the coefficients of the relevant bit stream. This polynomial is divided (the coefficients are calculated modulo-2) by the generator-polynomial.

The remainder of this polynomial division is the CRC sequence transmitted over the bus. In order to implement this function, an nCRC bit shift register CRC_RG(nCRC -1:0) can be used. Each CRC sequence is calculated in a separate shift register block. If NXTBIT denotes the next bit of relevant bit stream, the CRC sequences are calculated as follows:

```
CRC_RG(nCRC -1: 0) = CRC_INIT_VECTOR; //initialize shift register
```

```
REPEAT
```

```
    CRCNXT = NXTBIT EXOR CRC_RG(nCRC -1);
```

```
CRC_RG(nCRC -1: 1) = CRC_RG(nCRC -2: 0); //shift left by one position  
CRC_RG(0) = 0;  
IF CRCNXT THEN  
    CRC_RG(nCRC -1: 0) = CRC_RG(nCRC -1:0) EXOR (CRC polynomial);  
ENDIF
```

UNTIL (NXTBIT = [End of bit stream] or there is an error condition).

After the transmission/reception of the last bit of the relevant bit stream, each CRC_RG contains one of the five CRC SEQUENCES.

6.6.5 Specification of EF

6.6.5.1 Description

The EF consists of two different fields. The first field contains a superposition of error flags contributed from different nodes. The second field is the error delimiter.

6.6.5.2 Error flag

There are two forms of error flag used, the active error flag and the passive error flag:

- the active error flag shall consist of six consecutive dominant bits, and
- the passive error flag shall consist of six consecutive recessive bits unless it is 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 the error flag violates the rule of bit stuffing or destroys the bit field requiring fixed form. Consequently, all other nodes shall detect an error condition too and shall start sending an error flag. So, the sequence of dominant bits, which may 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 6 bit and a maximum of 12 bit.

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, and the second exception is a passive error flag that starts less than 6 bit before the end of the FCRC 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 subsequent equal bits after detecting an error condition. The passive error flag is complete when these six equal bits are detected.

6.6.5.3 Error delimiter

The error delimiter shall consist of eight recessive bits. After sending an error flag, each node shall monitor the bus until it detects a recessive bit. At this point in time, every node shall finish sending its error flag and all nodes shall start sending seven more recessive bits simultaneously, to complete the 8-bit error delimiter.

6.6.6 Specification of OF

6.6.6.1 Types

Following types of OF shall have the same format.

- LLC requested OF: This OF is requested by the LLC sub-layer to indicate an internal overload situation (see [6.6.21.4.3](#)).
- Reactive OF: The transmission of the reactive OF shall be initiated by the MAC sub-layer upon certain error conditions (see [6.6.21.4.3](#)).

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

6.6.6.2 Overload flag

The overload flag shall consist of six dominant bits. It destroys the fixed form of the intermission field (see [6.6.7.2](#)). Consequently, all other nodes also detect an overload condition and shall start sending an overload flag.

6.6.6.3 Overload delimiter

The overload delimiter shall consist of eight recessive bits. After sending an overload flag, each 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 8-bit overload delimiter.

6.6.7 Inter-frame space specification

6.6.7.1 Separation of frames

DFs and RFs shall be separated from preceding frames, whatever frame type the preceding frames are (DF, RF, EF, OF), by a time period called inter-frame space. In contrast to this, EFs and OFs shall not be preceded by inter-frame space, and multiple OFs shall not be separated by inter-frame space.

Inter-frame space shall contain the bit field intermission and bus idle time. For error-passive nodes, which have been transmitter of the previous frame, inter-frame space shall also contain the node's suspend transmission time. [Figure 6](#) specifies the inter-frame space for nodes, which are not error-passive or are receivers of a previous DF or RF. [Figure 7](#) specifies the inter-frame space for error-passive nodes, which transmitted the previous DF or RF.

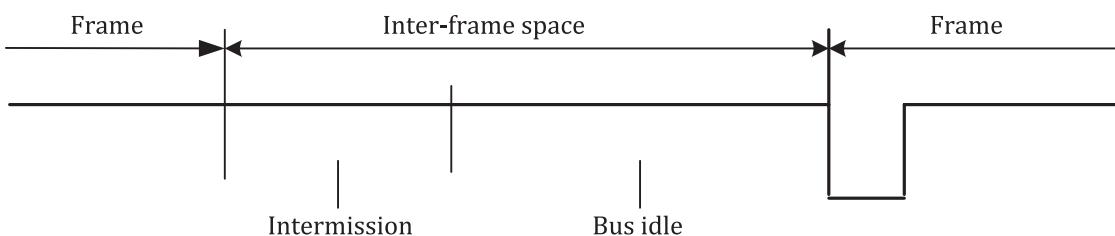


Figure 6 — Inter-frame space for nodes, which are not error-passive or are receivers of previous DF or RF

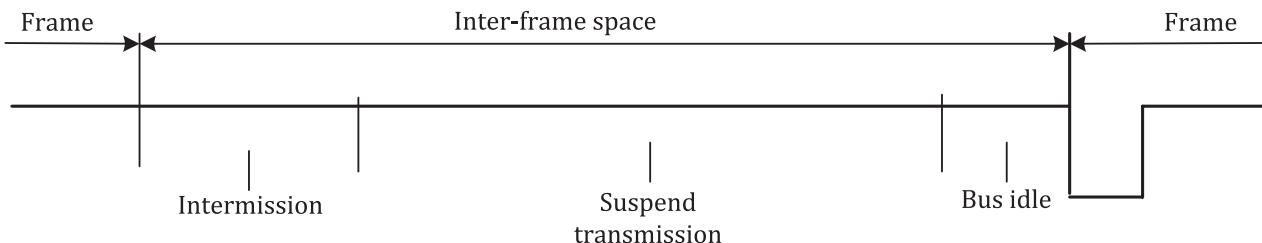


Figure 7 — Inter-frame space for error-passive nodes, which are transmitters of previous DF or RF

6.6.7.2 Intermission

The intermission field shall consist of three recessive bits. During intermission, a node shall not start transmission of DFs or RFs. An OF may be transmitted.

The detection of a dominant bit at the third bit of intermission shall be interpreted as SOF (see [6.6.8](#)).

6.6.7.3 Bus idle

The period of bus idle can be of arbitrary length. The bus shall be recognized to be idle by receivers and by error active transmitters when the third bit of intermission is seen recessive; by error passive transmitters when the eighth bit of suspend transmission time is seen recessive; or when the bus integrating state is left (see [6.6.7.5](#)). When the bus is idle, 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 time shall be interpreted as SOF.

6.6.7.4 Suspend transmission

An error-passive node, which has been transmitter of the previous frame, shall suspend the start of further frame transmissions for 8 bit times following intermission. If another node starts a transmission during that suspend transmission time, the node shall become a receiver of this DF or RF frame.

6.6.7.5 Bus re-integration

CAN nodes shall start the bus re-integration after starting the protocol operation, during bus-off recovery, or (for nodes that are FD tolerant, FD enabled, or XL enabled) after detecting a protocol exception event.

CAN nodes that are not in bus-off state shall finish the bus re-integration when the idle condition (see [3.34](#)) is detected.

There shall be a bit counter that is reset when the bus re-integration state is started or when the CAN bus is detected dominant at the sample point. The bit counter shall be incremented when the CAN bus is detected recessive at the sample point. The idle condition shall be detected when this bit counter reaches the value 11. For the detection of the bus-off recovery condition (see [8.2](#)), there shall be a second counter that is incremented once each time the idle condition is detected.

Nodes that are in bus-off state shall finish the bus re-integration when the bus-off recovery condition is met, and the idle condition is detected.

For nodes that are FD tolerant, FD enabled, or XL enabled, there shall be a third reset condition for the bit counter.

It shall be reset when detecting an edge that causes synchronization. When synchronization occurs, the counting of the sequence of 11 consecutive recessive bits shall be restarted.

For an optional edge filtering see [7.3.5.3](#).

NOTE When an FD tolerant, FD enabled, or XL enabled node is re-integrating, it reacts on dominant pulses that are shorter than a nominal bit time by restarting its bit counter for idle detection to ensure the shorter bits in the data phase of an FD frame or an XL frame are not mistakenly missed when waiting for an idle condition, regardless of their phase relationship.

6.6.8 SOF

SOF shall mark the beginning of DFs and RFs. It shall be dominant.

A node shall send an SOF only when the bus is idle (see [6.6.7.3](#)). A node sampling a dominant bit during its suspend transmission time or at the third bit of intermission shall accept it as SOF.

If a node samples a dominant bit at the third bit of intermission, this node shall, if it has a pending transmission and if the node is error active or has been receiver of the previous frame, start transmitting its frame at the next bit with the first bit of its identifier and take part in arbitration, without first transmitting an SOF bit and without becoming receiver.

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

6.6.9 Elements of the MAC frame

On transmission, an LLC frame (see [Table 4](#)) shall be converted into a MAC DF or MAC RF. On reception, a MAC DF or MAC RF shall be converted into an LLC frame.

MAC DFs (see [Figure 8](#)) and MAC RFs (see [Figure 9](#)) shall be composed of the following seven different bit fields as specified in [Figure 5](#):

- SOF;
- arbitration field (contains identifier and part of format control);
- control field (contains DLC and part of format control);
- data field (contains LLC data field);
- CRC field;
- ACK field;
- EOF.

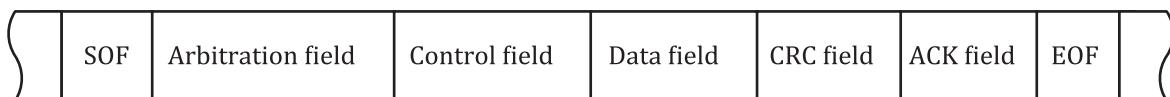


Figure 8 — MAC DF

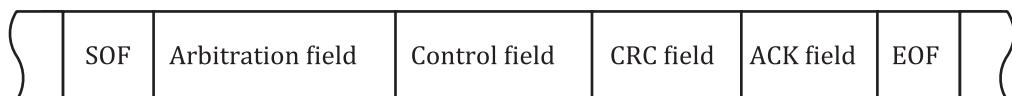


Figure 9 — MAC RF

The arbitration fields of the different frame formats are constructed to enable resolving collisions even between frames of different formats so that only one data frame transmitter continues transmitting. In case of remote frames, several transmitters may continue transmitting, but they transmit the same frame.

NOTE If nodes simultaneously transmit RFs using the same ID with different DLC values, the nodes increment their transmit error counters.

6.6.10 MAC frame in CBFF and CEFF

6.6.10.1 General

The CC frames can be data frames or remote frames, with an 11-bit long identifier or with a 29-bit long identifier. They can contain from zero to eight bytes in the data field.

6.6.10.2 Arbitration field

In CBFF, the arbitration field is composed of 11 identifier bits and the RTR bit. The RTR bit is dominant in DFs and recessive in RFs, as specified in [Figure 10](#) and in [Figure 11](#).

In CEFF, the arbitration field is composed of 29 identifier bits, the recessive SRR and IDE bits, as well as the RTR bit. The RTR bit is dominant in DFs and recessive in RFs, as specified in [Figure 12](#) and in [Figure 13](#).

The values of the identifier bits shall be passed from the ID field of the LLC frame. The values of the SRR and IDE bits shall be passed from the format field of the LLC frame. The value of the RTR bit shall be passed from the FTYP bit of the LLC frame.

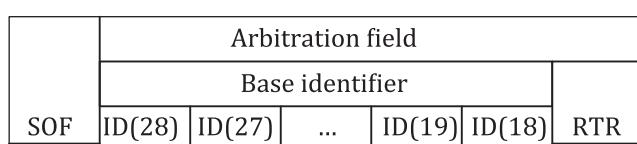


Figure 10 — SOF and arbitration field of CBFF data frame

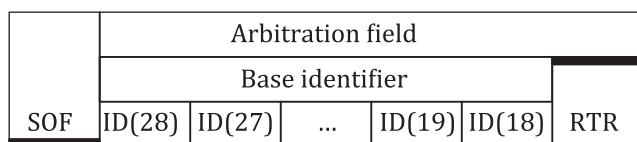


Figure 11 — SOF and arbitration field of CBFF remote frame

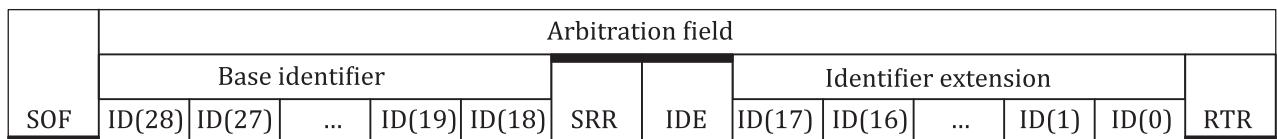


Figure 12 — SOF and arbitration field of CEFF data frame

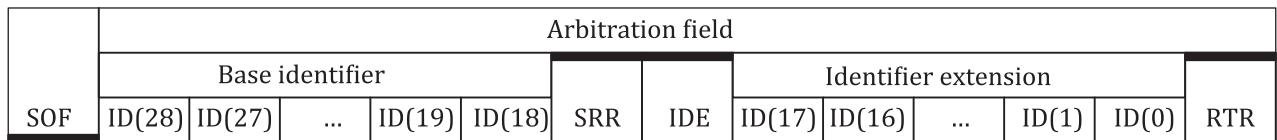


Figure 13 — SOF and arbitration field of CEFF remote frame

Arbitration

Arbitration between CC frames shall start with the ID28 bit and shall end with the RTR bit. When a node transmits a recessive bus state at one of these bits and samples a dominant bus state, it shall lose arbitration and it shall become receiver of that frame. The recessive values of SRR and IDE bit ensure that collisions between a frame in CBFF or FBFF and a frame in CEFF or FEFF, with both frames having the same base identifier, are resolved such that the frame in CBFF or FBFF prevails over the frame in CEFF or FEFF. The

recessive value of the RTR bit in RFs ensure that collisions between a DF and an RF having the same identifier and format are resolved such that the DF prevails over the RF.

SRR bit

The SRR bit shall be transmitted recessive, but receivers shall accept recessive and dominant SRR bits.

NOTE This means that the reception of dominant SRR bit is not regarded as form error.

6.6.10.3 Control field

Control field						
		DLC				
IDE/r1	r0	DLC(3)	DLC(2)	DLC(1)	DLC(0)	

Figure 14 — Control field in CC frames

[Figure 14](#) specifies the control field structure. The first two bits of the control field shall be transmitted dominant, followed by the DLC.

IDE / r1

The first bit of the control field is the IDE bit in CBFF, and it is r1 in CEFF, matching with the FDF bit in FEFF and a CAN FD tolerant CAN node shall detect a protocol exception event when this bit is received recessive.

r0

The second bit of the control field is r0 bit. In CBFF, this bit matches with the FDF bit in FBFF and a CAN FD tolerant CAN node shall detect a protocol exception event when this bit is received recessive. In CEFF, receivers shall accept it with recessive and with dominant state.

DLC

The value of the data length code DLC is passed from the four LSBs of the DLC in the LLC frame. The relationship between the DLC value and the number of data bytes is specified in [Table 5](#).

6.6.10.4 Data field

There is no data field in DFs with DLC=0 and in all RFs. In that case, the control field is immediately followed by the CRC field. The data field shall consist of the data passed from the LLC frame. The number of data field bytes depends on the LLC frame format and on the DLC. The data bytes are transmitted sequentially starting with byte 0, each byte shall contain eight bit. [Figure 15](#) specifies the data field structure.

Data field						
Byte 0			...	Byte f(DLC)		
Bit(7)	...	Bit(0)		Bit(7)	...	Bit(0)

Figure 15 — Data field in data frames

6.6.10.5 CRC field

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

FCRC sequence

The frame check sequence shall be derived from a CRC (BCH code). It shall be calculated with the algorithm shown in [6.6.4.4](#), using the polynomial CRC_15. [Figure 16](#) specifies the CRC field structure.

CRC field					
FCRC sequence					CRC delimiter
Bit(14)	Bit(13)	...	Bit(1)	Bit(0)	CRC delimiter

Figure 16 — CRC field in CC frame

The relevant bit stream for CRC calculation is the bit stream consisting of SOF, arbitration field, control field, and (if present) data field. In CC frames, stuff bits shall not be included in the relevant bit stream for CRC calculation.

CRC delimiter

The CRC delimiter shall follow the FCRC sequence. In CC frames, the CRC delimiter is one single recessive bit.

6.6.10.6 ACK field

The ACK field shall contain the ACK slot and the ACK delimiter as specified in [Figure 17](#). In the ACK field, the transmitter node shall send recessive bits. All receivers check the consistency of the received DF or RF and shall acknowledge a consistent frame and shall flag an inconsistent frame by means of an EF (see [6.6.2](#)). A DF or RF that is not acknowledged shall be regarded as corrupted and shall be flagged by the transmitting node with an EF.

ACK slot

All nodes that receive the matching FCRC sequence shall send an ACK within the ACK slot by overwriting the recessive bit of the transmitter with a dominant bit (they send ACK).

ACK delimiter

The ACK delimiter, being the last bit of the ACK field, shall be a recessive bit. Consequently, the ACK slot is surrounded by recessive bits (CRC delimiter, ACK delimiter).

ACK field	
ACK slot	ACK delimiter

Figure 17 — ACK field in CC or FD frame

6.6.10.7 EOF

DFs and RFs shall be delimited by a flag sequence consisting of seven recessive bits forming the EOF. EOF is followed by inter-frame space, see [6.6.7](#).

6.6.11 MAC frame in FBFF and FEFF

6.6.11.1 General

The FD frames are data frames in two formats, FBFF with an 11-bit long identifier and FEFF with a 29-bit long identifier. They can contain from zero to 64 data bytes. There are no remote FD frames.

6.6.11.2 Arbitration field

In FBFF, the arbitration field is composed of 11 identifier bits and the dominant RRS bit, as specified in [Figure 18](#).

In FEFF, the arbitration field is composed of 29 identifier bits, the recessive SRR and IDE bits, as well as the dominant RRS bit, as specified in [Figure 19](#).

The values of the identifier bits shall be passed from the ID field of the LLC frame. The values of the SRR, IDE and RRS bits shall be defined by the format field of the LLC frame.

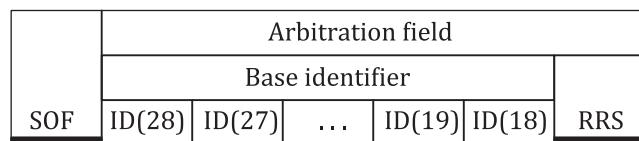


Figure 18 — SOF and arbitration field of FBFF data frame

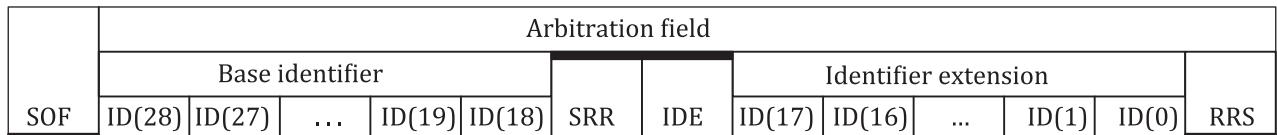


Figure 19 — SOF and arbitration field of FEFF data frame

Arbitration

Arbitration between FD frames shall start with the ID28 bit and shall end with the FDF bit in the control field. When a node transmits a recessive bus state at one of these bits and samples a dominant bus state, it shall lose arbitration and it shall become receiver of that frame. The recessive value of FDF bit in FD frames and the dominant value of the reserved bits in CC frames ensure that collisions between a CC frame and an FD frame, with both frames having the same identifier, are resolved such that the CC frame prevails over the FD frame.

Nodes that do not support the XL frame format, that sample a dominant bus state when transmitting a recessive FDF bit may treat this as a bit error instead of losing arbitration.

SRR bit

The SRR bit shall be transmitted recessive, but receivers shall accept recessive and dominant SRR bits.

NOTE 1 This means that the reception of dominant SRR bit is not regarded as form error.

RRS bit

The RRS bit shall be transmitted dominant, but receivers shall accept recessive and dominant RRS bits.

NOTE 2 This means that the reception of recessive RRS bit is not regarded as form error.

6.6.11.3 Control field

The control field has different formats for FBFF and FEFF, shown in [Figures 20](#) and [21](#).

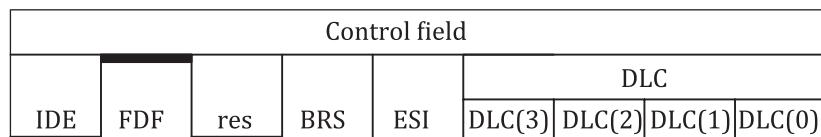


Figure 20 — Control field in FBFF

[Figure 20](#) specifies the control field structure for FBFF, [Figure 21](#) specifies the control field structure for FEFF. The only difference is the IDE bit, which distinguishes between the identifier formats and is part of the control field in FBFF and part of the arbitration field in FEFF. The value of the FDF bit is passed from the format field of the LLC frame.

Control field							
				DLC			
FDF	res	BRS	ESI	DLC(3)	DLC(2)	DLC(1)	DLC(0)

Figure 21 — Control field in FEFF**FDF**

This bit distinguishes between CC frames and FD or XL frames. It is recessive in FD and XL frames, and it is dominant in CC frames. In frames with 11-bit identifiers, FDF comes after the IDE bit. In frames with 29-bit identifiers, it comes as the first bit of the Control Field.

The FDF bit corresponds to the r0 bit in CC frames with 11-bit identifiers and to the r1 bit in CC frames with 29-bit identifiers.

res

In FD frames, the FDF bit shall be followed by the dominant res bit. This bit distinguishes between FD frames and the XL frame. When an FD enabled receiver detects the res bit to be recessive instead of the expected dominant it shall, when protocol exception handling is enabled, detect a protocol exception event; when protocol exception handling is disabled, it shall treat this as a form error.

BRS

This bit indicates whether the bit rate is switched inside the FD frame. If the bit is detected recessive, the bit rate shall be switched from the nominal bit rate of the arbitration phase to the preconfigured FD data bit rate of the data phase. It is not required for the BRS bit to be the same in all FD frames in a single network. The value of the BRS bit is passed from the BRS bit of the LLC frame.

ESI

When the ESI bit of the LLC frame is set, then this flag shall be transmitted recessive, else this flag shall be transmitted dominant by error active nodes and shall be transmitted recessive by error passive nodes.

DLC

The value of the data length code DLC is passed from the four LSBs of the DLC in the LLC frame. The relationship between the DLC value and the number of data bytes is specified in [Table 5](#).

6.6.11.4 Data field

There is no data field in DFs with DLC=0. In that case, the control field is immediately followed by the CRC field. The data field shall consist of the data passed from the LLC frame, the number of bytes given by frame format and DLC. The data bytes are transmitted sequentially starting with byte 0, each byte shall contain eight bit. [Figure 15](#) specifies the data field structure.

6.6.11.5 CRC field

The CRC field shall contain the stuff count and the FCRC sequence followed by a recessive CRC delimiter.

Stuff count

In FD frames, the stuff count shall be at the beginning of the CRC field. It shall represent a number in the range from 0 to 7_d (Gray-coded in SBC3, SBC2, and SBC1; followed by the parity bit SBC0) as specified in [Table 8](#).

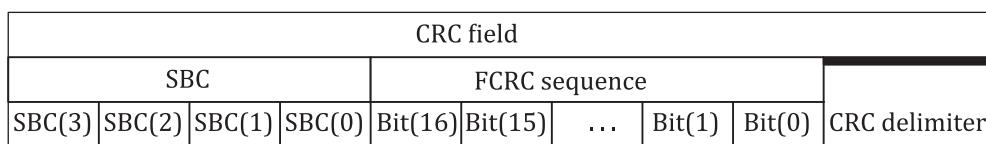
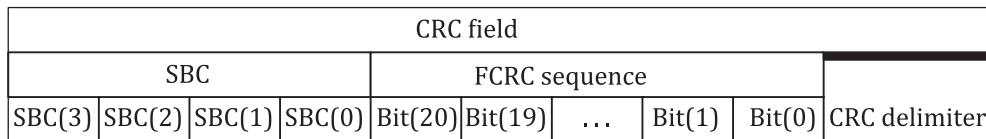
Table 8 — Coding of the stuff count

	Value							
Number of stuff bits modulo 8	0_d	1_d	2_d	3_d	4_d	5_d	6_d	7_d
SBC (Gray-coded with parity bit)	0000_b	0011_b	0110_b	0101_b	1100_b	1111_b	1010_b	1001_b

Both transmitter and receivers of a frame shall count the number of stuff bits before the first fixed stuff bit in the frame. The transmitter shall transmit its stuff bit count, coded as stuff count, at the beginning of the CRC field, before the FCRC sequence. Receivers shall check whether the received stuff count matches with the value calculated from their own stuff bit count.

FCRC sequence

The frame check sequence shall be derived from a CRC (BCH code). It shall be calculated with the algorithm shown in [6.6.4.4](#), using the polynomial CRC_17 for frames with up to and including 16 data bytes or CRC_21 for longer frames. [Figure 22](#) specifies the CRC field structure when CRC_17 is used. [Figure 23](#) specifies the CRC field structure when CRC_21 is used.

**Figure 22 — CRC field in FD frame using CRC_17****Figure 23 — CRC field in FD frame using CRC_21**

The relevant bit stream for the CRC calculation shall be the bit stream consisting of SOF, arbitration field, control field, and (if present) data field. The stuff count and the dynamic stuff bits, but not the fixed stuff bits, shall be included in the relevant bit stream for CRC calculation of FD frames.

CRC delimiter

The CRC delimiter shall follow the FCRC sequence. In FD frames, the CRC delimiter may consist of one or two recessive bits. A transmitter shall send one recessive bit as CRC delimiter, but it shall accept two recessive bits before the edge from recessive to dominant that starts the acknowledge slot. A receiver sends its acknowledge bit after the first CRC delimiter bit.

NOTE CAN nodes switch back from the data phase to the arbitration phase of FD frames when they reach the sample point of the (first bit of the) CRC delimiter.

6.6.11.6 ACK field

The ACK field shall contain the ACK slot and the ACK delimiter as specified in [Figure 17](#). In the ACK field, the transmitter node shall send recessive bits. All receivers check the consistency of the received DF and shall acknowledge a consistent frame and shall flag an inconsistent frame by means of an EF (see [6.6.2](#)). A DF that is not acknowledged shall be regarded as corrupted and shall be flagged by the transmitting node with an EF.

ACK slot

Nodes that receive the matching FCRC sequence and the matching stuff count shall send an ACK within the ACK slot by overwriting the recessive bit of the transmitter with a dominant bit (they send ACK). In FD

frames, all nodes shall accept an up to two bit long dominant phase of overlapping ACK slot bits as a valid ACK, to compensate for phase shifts between the receivers.

ACK delimiter

The ACK delimiter, being the last bit of the ACK field, shall be a recessive bit. Consequently, the ACK slot is surrounded by recessive bits (CRC delimiter, ACK delimiter).

6.6.11.7 EOF

DFs shall be delimited by a flag sequence consisting of seven recessive bits forming the EOF. The inter-frame space follows the EOF (see [6.6.7](#)).

6.6.12 MAC frame in XLFF

6.6.12.1 General

XL frames are DFs in XLFF with an 11-bit bit priority ID. The data field has a length of one byte to 2048 byte. There is no RF specified.

6.6.12.2 Arbitration field

The arbitration field of the XLFF as specified in [Figure 24](#) is composed of the priority ID, the RRS bit (passed from the LLC sub-layer priority identifier and the FTYP bit), the IDE bit, the FDF bit, and the XLF bit.

The values of the identifier bits shall be passed from the ID field of the LLC frame. The values of the IDE, FDF and XLF bits shall be passed from the format field of the LLC frame. The value of the RRS bit shall be passed from the FTYP bit of the LLC frame.

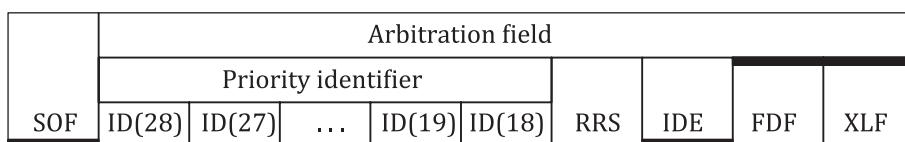


Figure 24 — SOF and arbitration field of XLFF data frame

Arbitration

Arbitration between XL frames shall start with the ID28 bit and shall end with the XLF bit. When a node transmits a recessive bus state at one of these bits and samples a dominant bus state, it shall lose arbitration and it shall become receiver of that frame. The recessive value of FDF and XLF bits in XL frames and the dominant value of the reserved bits in CC frames and FD frames ensure that collisions between a CC frame and an XL frame or between an FD frame and an XL frame, with both frames having the same identifier, are resolved such that the CC frame or the FD frame prevails over the XL frame.

RRS bit

The position of the RRS bit is the next bit after the last identifier bit, like the RTR bit in CBFF and CEFF and like the RRS bit in FBFF and FEFF. When a node transmits a recessive RRS bit and it samples a dominant RRS bit, it shall interpret this as an arbitration lost situation. Receivers shall accept dominant as well as recessive RRS bits.

NOTE 1 This means that the reception of recessive RRS bit is not regarded as form error.

IDE bit

The IDE bit shall be dominant.

FDF bit

This bit distinguishes between CC frames and FD frames as well as XL frames. The FDF bit shall be recessive. When a node transmits a recessive FDF bit and it samples a dominant FDF bit, it shall interpret this as an arbitration lost situation.

NOTE 2 On arbitration lost, the node becomes receiver of a CC frame.

XLF bit

This bit distinguishes between FD frames and XL frames. The XLF bit shall be recessive. When a node transmits a recessive XLF bit and it samples a dominant XLF bit, it shall interpret this as an arbitration lost situation.

NOTE 3 On arbitration lost, the node becomes receiver of an FD frame.

6.6.12.3 Control field

Control field																						
resXL	ADS			SDT		SEC	DLC		SBC		PCRC		VCID		AF							
	ADH	DH1	DH2	DL1	Bit(7)	•	Bit(0)	Bit(10)	•	Bit(0)	SBC(2)	SBC(1)	SBC(0)	PCRC(12)	•	PCRC(0)	VCID(7)	•	VCID(0)	AF(31)	•	AF(0)

Figure 25 — Control field in XL frames

Figure 25 specifies the control field structure. The SDT, the SEC bit, the DLC, the VCID and the AF shall be passed from the LLC frame. The other elements of the control field are resXL bit, ADS, SBC and PCRC.

resXL bit

The resXL bit shall be dominant. It is reserved for future expansion of the protocol.

Node configuration shall determine how a receiver reacts when it detects the resXL bit to be recessive instead of the expected dominant value. When protocol exception event detection for resXL bit is disabled, CAN XL nodes shall treat this as a form error. When protocol exception event detection for resXL bit is enabled, CAN XL nodes shall detect a protocol-exception event (see [6.6.17.2](#)).

ADS

The arbitration to data sequence (ADS) has two purposes:

- switching the bit rate from nominal bit rate to the XL data bit rate;
- switching the CAN transceiver mode from arbitration mode to data TX mode or to data RX mode.

Enabling and disabling of the CAN transceiver mode switching shall be configurable. It can be enabled when an appropriate CAN transceiver supporting mode switching is connected.

The ADS consists of the bits ADH, DH1, DH2, and DL1 as specified in Figure 25. The ADH bit shall be the last bit with nominal bit time before the beginning of the XL data phase with XL data bit rate. The ADH bit shall be transmitted as a recessive bit. The subsequent bits DH1 and DH2 shall be the first bits of the XL data phase and shall be transmitted recessive.

NOTE ADS, together with resXL bit, is designed as a pattern to provide synchronizing edges before and after the transition.

Independent of the CAN transceiver mode switching, both transmitter and receiver ignore the actual sampled ADH bit value on the CAN bus (see [6.6.21.2 d](#)), Exception 3).

Bit rate switching

The bit rate shall be switched from the nominal bit rate to the XL data bit rate at the boundary between the bits ADH and DH1.

CAN transceiver mode change

[Subclause 7.5](#) specifies how mode switching is signalled to the CAN transceiver. When an appropriate CAN transceiver supporting mode switching is connected and the CAN transceiver mode switching is enabled, the time duration of the ADH bit shall be used to signal the CAN transceiver to switch its operating mode from the (default) arbitration mode:

- into the data TX mode if the node is transmitter,
- into the data RX mode if the node is receiver.

When the CAN transceiver is not able to switch between operating modes, or when the CAN transceiver mode switching is not used in the network, the signalling shall be disabled by configuration and the CAN transceiver remains in the arbitration mode during the whole frame.

Synchronization

A receiver performs a hard synchronization at the edges from the XLF bit to the resXL bit preceding ADS and from the DH1 and DH2 bits to the DL1 bit (see [7.3.5](#)). Between the two synchronizations, a phase error accumulates, caused by clock speed differences between the transmitter and the receiver, by the changed transceiver operating modes, and by the introduction of the PWM coding delay. At the bit rate switch, the absolute value of the phase error in relation to the length of the bit time increases by the amount of the bit rate ratio. A tolerance range shall compensate this for the receiver when detecting the DL1 bit. The receiver shall tolerate from one to six consecutively sampled recessive bits beginning with the position of the DH1 bit. The first bit sampled dominant shall be accepted as DL1. The receiver shall tolerate a missing DH2 bit. The receiver shall detect a form error when it detects seven consecutively sampled recessive bits beginning with the position of the DH1 bit. The receiver shall also detect a form error when it samples DH1 as dominant, what corresponds to an amount of 0 consecutive recessive bits.

SDT

The SDT shall be an 8-bit value, passed from the LLC frame.

SEC bit

The SEC bit shall be passed from the LLC frame.

DLC

The data length code shall be an 11-bit value in the range of 0 to 2047_d; it shall be passed from the LLC frame. The length of the data field shall be in the range from 1 byte to 2048 bytes. The actual length of the data field shall be one byte more than the value of the DLC.

EXAMPLE A 128-byte MAC data field is indicated by a DLC of 127.

SBC

The stuff bit count shall be a 3-bit value, giving information on the number of dynamic stuff bits in the arbitration field. The SBC represents a number in the range from 0 to 3_d (gray-coded in SBC2 and SBC1 followed by the parity bit SBC0) as specified in [Table 9](#).

Table 9 — Coding of the SBC

	Value			
	0 _d	1 _d	2 _d	3 _d
Number of stuff bits				
SBC (Gray-coded with parity bit)	001 _b	010 _b	111 _b	100 _b

PCRC

The preface PCRC sequence shall be derived from a CRC (BCH code). It shall be calculated with the algorithm shown in [6.6.4.4](#), using the polynomial CRC_13.

The relevant bit stream for CRC calculation shall be the bit stream consisting of arbitration field, SDT, SEC bit, DLC and SBC. Dynamic stuff bits (of which up to three can occur before the FDF bit) shall be included in, while the static bits SOF, IDE, FDF, XLF, resXL and ADS as well as the fixed stuff bits shall be excluded from CRC calculation.

VCID

The VCID shall be an eight-bit value, passed from the LLC frame.

AF

The AF shall be a 32-bit value, passed from the LLC frame.

6.6.12.4 Data field

The data field shall consist of the data passed from the LLC frame, the number of bytes given by frame format and DLC. The data bytes are transmitted sequentially starting with byte 0, each byte shall contain eight bit. [Figure 15](#) specifies the data field structure.

6.6.12.5 CRC field

The CRC field contains the frame CRC (FCRC) sequence and the format check pattern (FCP) as specified in [Figure 26](#). The FCRC sequence shall be derived from a CRC (BCH code). It shall be calculated with the algorithm shown in [6.6.4.4](#), using the polynomial CRC_32.

CRC field								
FCRC sequence					FCP			
Bit(31)	Bit(30)	...	Bit(1)	Bit(0)	FCP(3)	FCP(2)	FCP(1)	FCP(0)

Figure 26 — CRC field in XL frame

The relevant bit stream for the CRC calculation shall be the bit stream consisting of arbitration field, control field, and data field. It shall exclude the same static bits as the PCRC. Both dynamic and fixed stuff bits shall be excluded from the CRC calculation.

FCP sequence

The FCP sequence is the part of the frame directly before the point where the bit rate is switched back from the data bit rate to the nominal bit rate and where the PMA is signalled to switch back into the arbitration mode. It provides a synchronizing edge before the transition, from FCP2 to FCP1. The FCP3 and FCP2 bits are transmitted as recessive bits. The FCP1 and FCP0 bits are transmitted as dominant bits. The FCP0 bit shall be the last bit of the XL data phase.

6.6.12.6 ACK field

The ACK field contains the DAS, the ACK slot, and the ACK delimiter as specified in [Figure 27](#). The DAH, AH1, AH2, ACK slot, and ACK delimiter bits shall be transmitted as recessive bits. The AL1 bit shall be transmitted as a dominant bit. All receivers check the consistency of the received DF and shall acknowledge a consistent frame and shall flag an inconsistent frame by means of an EF (see [6.6.2](#)). A DF that is not acknowledged shall be regarded as corrupted and shall be flagged by the transmitting node with an EF.

ACK field					
DAS			ACK		
DAH	AH1	AL1	AH2	ACK slot	ACK delimiter

Figure 27 — ACK field in XL frame

DAS

The data to arbitration sequence (DAS) has two purposes:

- switching the bit rate from the XL data bit rate to the nominal bit rate;
- switching the CAN transceiver mode from the data TX mode or the data RX mode to the arbitration mode, if the mode is switched in the preceding ADS.

The DAS consists of the DAH, AH1, AL1 and AH2 bits. The DAH bit shall be the first bit with the nominal bit time after the end of the XL data phase with the XL data bit rate. DAS is designed as a pattern to provide, together with the FCP, synchronizing edges before and after the bit rate switch and (if enabled) the CAN transceiver mode switch. A receiver performs a synchronization at the edge from the FCP2 bit to the FCP1 bit in the CRC field. A receiver performs a hard synchronization from the DAH bit or the AH1 bit to the AL1 bit (see [7.3.5](#)).

Between the two synchronizations, a phase error accumulates, caused by clock speed differences between transmitter and receiver, by the changed transceiver operating modes, and by the elimination of the PWM coding delay. The tolerance range specified in the following compensates this for the receiver. A receiver shall interpret the bit following the DAH bit as the AH1 bit if it is sampled recessive and it shall interpret the bit as AL1 bit if it is sampled dominant. A receiver shall tolerate a missing AH1 bit. The transmitter shall detect a bit error, when it samples any bit in the ACK field, except the ACK slot, different from its transmitted value.

Bit rate switching

The bit rate shall be switched from the XL data bit rate to the nominal bit rate at the boundary between the bits FCP0 and DAH.

CAN transceiver mode change to arbitration mode

[Subclause 7.5](#) specifies how mode switching is signalled to the transceiver. When the operating mode is switched in the preceding ADS, the time duration of the DAH bit is used to signal the transceiver to switch its operating mode from its actual operating mode back to the (default) arbitration mode.

ACK slot

Receivers check the consistency of the received XL frame and shall acknowledge a consistent frame by sending a dominant bit in the ACK slot.

ACK delimiter

The ACK delimiter shall be transmitted as a recessive bit. This means, the ACK slot is surrounded by recessive bits (AH2 and ACK delimiter).

6.6.12.7 EOF

DFs shall be delimited by a flag sequence consisting of seven recessive bits forming the EOF. EOF is followed by inter-frame space, see [6.6.7](#).

6.6.13 MAC frame coding

6.6.13.1 Coding requirements

To limit the maximum distance between edges available for synchronization, the frame segments from SOF to the frame FCRC sequence shall be coded by the method of the bit stuffing. The remaining bit fields of the frame (ACK field, EOF, and, for XL Frames, FCP) shall not be stuffed.

Two methods of the bit stuffing are used in DFs and RFs: dynamic bit stuffing and fixed 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 whole bit time.

6.6.13.2 Dynamic bit stuffing

Dynamic bit stuffing shall be used in CC frames from the SOF up to and including the FCRC sequence. Dynamic bit stuffing shall be used in FD frames from the SOF up to and including the data field. Dynamic bit stuffing shall be used in XL frames from the SOF up to and including the arbitration field. The last position where such a dynamic stuff bit is possible in XL frames is before the FDF bit. The maximum number of dynamic stuff bits in XL frames is three.

Whenever a transmitter detects five consecutive bits (including dynamic stuff bits) of the identical value in the bit stream to be transmitted, it shall insert a complementary bit (called dynamic stuff bit) into the actual transmitted bit stream (see [Table 10](#)). The receiver shall recognize a sequence of five consecutive bits of identical value and shall discard the following dynamic stuff bit.

Table 10 — Dynamic bit stuffing

Destuffed bit stream	01011111010 _b	10100000101b	01011111000010b	10100000111101b
Stuffed bit stream	01011111o010 _b	10100000i101b	01011111o0000i10b	10100000i1111o01b

"0", "o" = dominant (stuff) bit; "1", "i"= recessive (stuff) bit

6.6.13.3 Fixed bit stuffing

6.6.13.3.1 CAN FD CRC field

In the CRC field of FD frames, the stuff bits shall be inserted at fixed positions; they are called fixed stuff bits. There shall be a fixed stuff bit before the first bit of the stuff count, even if the last bits of the preceding field are not a sequence of five consecutive bits of identical value. If the last bits of the preceding field are a sequence of five consecutive bits of identical value, there shall be only the fixed stuff bit, there shall not be two consecutive stuff bits. A further fixed stuff bit shall be inserted after each fourth bit of the CRC field. The value of such a fixed stuff bit shall be the inverse value of the bit preceding the fixed stuff bit. A receiver shall discard the fixed stuff bits from the bit stream for the CRC check. It shall detect a form error if the fixed stuff bit has the same value as its preceding bit. The number of fixed stuff bits in the CRC field of FD frames is equal to the maximum number of stuff bits that would result from applying the dynamic bit stuffing method of the CC frames.

6.6.13.3.2 CAN XL data phase

Fixed bit stuffing shall be applied in the control field, data field and FCRC sequence of XL frames. Here, the stuff bits are inserted at fixed positions; they are called fixed stuff bits. The stuff rate shall be 11, a fixed stuff bit shall be inserted after each 10th frame bit, counted from and including the DL1 bit in the ADS. The value of a fixed stuff bit shall be the inverse value of the bit preceding the fixed stuff bit. A receiver shall discard the fixed stuff bits from the bit stream for the CRC check; it shall indicate a form error if the fixed stuff bit has the same value as its preceding bit. The last fixed stuff bit shall be in the FCRC sequence of the CRC field.

NOTE The position of the last fixed stuff depends on the value of the DLC. The latest possible position of a fixed stuff bit is directly after bit (0) of the FCRC sequence.

6.6.14 Data frame acknowledgement

All receivers shall check the consistency of all received DFs and RFs and they shall acknowledge all consistent DFs and RFs. Acknowledgement shall not depend on the frame's identifier.

6.6.15 Frame validation

6.6.15.1 General

The behaviour is different if error signalling is enabled or is disabled. See [6.6.21](#) for details.

6.6.15.2 Behaviour when error signalling is enabled

The point in time at which DFs and RFs are taken to be valid is different for receivers and transmitters.

Receiver

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 not inhibit frame validation 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 OF (see [6.6.21](#)).

Transmitter

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 [6.6.21.3.1](#).

NOTE If the transmitter samples a dominant bit at the last bit of EOF (a global or a local error at the sender), then the frame is valid for the receiver, but not for the transmitter. The transmitter places an EF and restarts the transmission of the frame and the frame is received twice. The receiver treats the dominant bits as an OF and receives the next frame as an independent second frame.

6.6.15.3 Behaviour when error signalling is disabled

The point in time at which a DF is taken to be valid is the same for receivers and transmitters.

A frame shall be valid for receivers, if there is no error detected until the ACK slot and the ACK slot is sampled dominant.

The frame shall be valid for a transmitter, if the ACK slot is sampled dominant. If the transmitter considers the frame as invalid it shall perform a retransmission as specified in [6.5.3](#).

The bits of the frames sampled after the ACK slot shall have no impact on the validation of the frame for a receiver or a transmitter.

6.6.16 Order of bit transmission

[Figure 2](#) shows the order of the bit transmission of the XL data frames. Within a field, the MSB is transmitted first. Within the data field, the bytes are transferred from byte 0 to the last byte (indicated by the DLC). Within each byte, the bits are transferred from bit(7) down to bit(0).

A broad line at the bottom of a bit indicates that the bit shall be transmitted with the logical value 0; a broad line at the top of a bit indicates that the bit shall be transmitted with the logical value 1. Stuff bits are not shown in [Figure 2](#).

6.6.17 Medium access method

6.6.17.1 Bus access

An error-active node may access the bus as soon as the bus is idle (see [6.6.7.3](#)). An error-passive node, which is the receiver of the current or previous frame, may access the bus as soon as the bus is idle. An error-

passive node, which is transmitter of the current frame or has been transmitter of the previous frame, may access the bus as soon as its suspend transmission time 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 identifier-based arbitration.

6.6.17.2 Protocol exception event

CAN nodes detect a protocol exception event as it is specified in [6.6.10.3](#), [6.6.11.3](#), and [6.6.12.3](#). CAN XL nodes shall also detect a protocol exception event when error signalling is disabled as specified in [6.6.21.4](#). As a reaction to the protocol exception event, the error counters shall not be changed, the hard synchronization shall be enabled, the node shall send recessive bits and shall enter the bus re-integration state (see [6.6.7.5](#)).

6.6.17.3 Transmission of MAC frames

MAC frames can be started when the node is allowed to access the bus according to [6.6.17.1](#).

MAC frames that lose arbitration shall be re-arbitrated. A MAC frame that is re-arbitrated shall be handled like not re-arbitrated ones. It participates in the arbitration process to gain bus access (see [6.5.2](#)). MAC frames that are not acknowledged or that are disturbed by errors during the transmission are retransmitted. A retransmitted MAC frame participates in the arbitration process to gain bus access, like a non-retransmitted MAC frame (see [6.5.3](#)). The automatic retransmission may be disabled by configuration.

6.6.17.4 Identifier 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 transmitter may continue to send. When a CAN node sends a recessive level and monitors a dominant level, the CAN node loses arbitration and shall withdraw from transmitting any more bits, and it shall become receiver of this frame. When a dominant level is sent and a recessive level is monitored, the node shall detect a bit error.

Identifier based arbitration shall be performed in CBFF from the first bit of the identifier up to the IDE bit, in CEFF from the first bit of the identifier up to the RTR bit, and in FBFF, FEFF, as well as in XLFF from the first bit of the identifier up to the bit following the FDF bit.

6.6.17.5 Frame priority

Among two or more frames with different identifiers, the identifier-based arbitration assigns the higher priority to the identifier containing the lower binary value.

If one DF in CBFF, one DF in FBFF, and one DF in XLFF with the same identifier are initiated at the same time, the DF in FBFF has a lower priority than the DF in CBFF, and the DF in XLFF has a lower priority than the DF in FBFF and the DF in CBFF. If one DF in CEFF and one DF in FEFF with the same identifier are initiated at the same time, the DF in FEFF has a lower priority than the DF in CEFF. This is achieved by losing arbitration at the FDF bit or at the XLF bit.

6.6.17.6 Collision resolution

Additional to the principle that transmissions are initiated only when the network segment is idle, there are further principles for the collision resolution of frames in XLFF. The correct function of the collision resolution requires that within one network segment, each MAC frame uses an arbitration field with a unique value. This enables the resolution of all simultaneously transmitted MAC frames independent of their frame formats.

NOTE The assignment of unique arbitration fields to the PDU is a task of higher OSI layers provided by means of the SDU.

6.6.18 MAC data consistency

MAC frames to be transmitted are prepared by the LLC sub-layer entity user. They are transferred via the node's controller host interface (see NOTE in this subclause) and the LLC sub-layer entity to the MAC entity. MAC frames may be stored in a shared memory. Data consistency of transmitted MAC frames from a shared memory shall be ensured by at least one of two methods.

- The MAC entity shall store the whole frame content to be transmitted in a temporary buffer that is filled before the transmission is started.
- The LLC entity shall check for data errors while the frame content to be transmitted is transferred to the MAC entity. If a data error is detected, the transmission shall not be started. If it is already started when the data error is detected, the node shall invalidate the frame by, e.g. switching to the restricted operation mode, see [6.6.19](#), or into bus monitoring mode, see [6.6.20](#). Receiving nodes do not see a valid MAC frame.

NOTE Data errors are, e.g. parity errors in a RAM word, data not provided in time, or data partially updated by the LLC user while a transmission is in progress. Switching to the operation modes restricted operation mode and bus monitoring mode is optional if the first method to ensure data consistency is used.

6.6.19 Restricted operation

CAN nodes shall support the restricted operation mode, in which they shall be able to receive DFs and RFs and shall acknowledge valid frames by means of transmitting a dominant bit in the ACK slot. But CAN nodes in restricted operation mode shall not transmit DFs and RFs. In case of an error condition or an overload condition, CAN nodes shall not send dominant bits. Instead they shall treat this as a protocol exception event and shall enter the bus re-integration state (see [6.6.7.5](#)). The error counters shall neither be incremented nor decremented while the CAN node is in restricted operation mode. If the CAN node in restricted operation mode is a potential primary time provider in a network, it shall be able to transmit a DF containing time reference messages to start up the network. Other frames shall not be transmitted. Nodes that do not support the XL frame format are not required to implement the restricted operation mode.

6.6.20 Bus monitoring

CAN nodes may provide the bus monitoring mode, where they shall be able to receive valid DFs and valid RFs, but the bus monitoring mode sends only recessive bits on the CAN bus and does not start a transmission. If the MAC sub-layer is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit shall be rerouted internally so that the MAC sub-layer monitors this dominant bit, although the CAN bus can remain in recessive state.

6.6.21 Error handling and overload handling

6.6.21.1 Control of error signalling

There shall be a configuration parameter to decide whether error signalling is enabled or is disabled. [Subclause 6.6.21.3](#) specifies the behaviour for the case when error signalling is enabled, [6.6.21.4](#) specifies the case when error signalling is disabled.

6.6.21.2 Error detection

The MAC sub-layer provides the following mechanisms for the error detection:

- bit monitoring;
- stuff rule check;
- frame format check;
- stuff count check in FD and in XL frames;

- 13-bit, 15-bit, 17-bit, 21-bit, or 32-bit CRC;
- ACK check.

There are five different error types, which are not mutually exclusive. For example, the transmitter sees both, bit error and stuff error, when a correctly transmitted stuff bit is disturbed.

a) Bit error

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

Exception 1: A monitored dominant bit shall not lead to a bit error when a recessive bit, which is not a dynamic stuff bit, is sent during arbitration, or a recessive bit is sent during ACK slot.

Exception 2: A node sending a passive error flag and detecting a dominant bit shall not interpret this as a bit error.

Exception 3: A transmitter shall not detect a bit error at the ADH bit, independent of the actual sampled bit value (see [6.6.12.3](#)).

b) Stuff error

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

NOTE When a fixed stuff bit in an FD frame or in an XL frame is not at its expected value, it is regarded as a form error, not as a stuff error,

c) PCRC error and frame CRC error

The CRC sequence consists of the result of the CRC calculation of the transmitter. The receivers shall calculate the CRC in the same way as the transmitter. There are two types of CRC errors: PCRC error and FCRC error.

A receiver shall detect a PCRC error when the calculated PCRC sequence does not equal the received one, or when the SBC of an XL frame does not match to the number of the received dynamic stuff bits, or when the SBC parity does not match. A receiver shall check for a PCRC error at the sample point of the bit following PCRC(0).

A receiver shall detect an FCRC error when the calculated frame FCRC sequence does not equal the received one, or when it detects an error in the FCP field of an XL frame, or when the SBC of an FD frame does not match to the number of the received dynamic stuff bits, or when the SBC parity does not match. A receiver shall check for an FCRC error condition at the sample point of the DAH bit in XL frames and in other frames at the sample point of the CRC delimiter.

d) Form error

A form error shall be detected in the following cases: when a fixed form bit field contains one or more unexpected bit values, or when a fixed stuff bit in an XL frame or in an FD frame is not at its expected value.

Exception 1: A receiver detecting a dominant bit at the last bit of EOF, or a node detecting a dominant bit at the last bit of the error delimiter or of the overload delimiter, shall not treat this as a form error.

Exception 2: A receiver detecting a different value for FCP shall not treat this as a form error.

Exception 3: A receiver shall not detect a form error at the ADH bit, independent of the actual sampled bit value (see [6.6.12.3](#)).

e) ACK error

A transmitter shall detect an ACK error whenever it does not detect a dominant bit during the ACK slot.

6.6.21.3 Behaviour when error signalling is enabled

6.6.21.3.1 Error handling

When error signalling is enabled, the node shall be able to transmit and to receive all CAN frame formats.

Whenever a bit error, a stuff error, a form error, a PCRC error, or an ACK error is detected by a node, the LLC sub-layer shall be informed, and the node shall start an error flag at the following bit.

NOTE A receiver reacts on a PCRC error at the bit after the detection of the PCRC error.

When a receiver detects an FCRC error, it shall not send a dominant bit in the ACK slot of the received frame.

A receiver receiving a CC frame or an XL frame and detecting an FCRC error shall send an EF starting with the first bit of EOF. A receiver receiving an FD frame and detecting an FCRC error shall send an EF after three bit-times following the CRC delimiter. This is shown in [Figure 28](#) for CC frames and in [Figure 29](#) for FD frames. Dominant bits seen between the CRC delimiter and the start of the EF shall not be treated as errors.

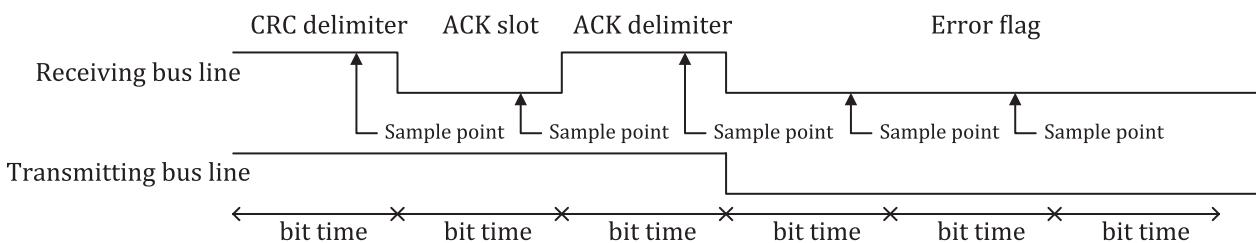


Figure 28 — CRC error in CC frames

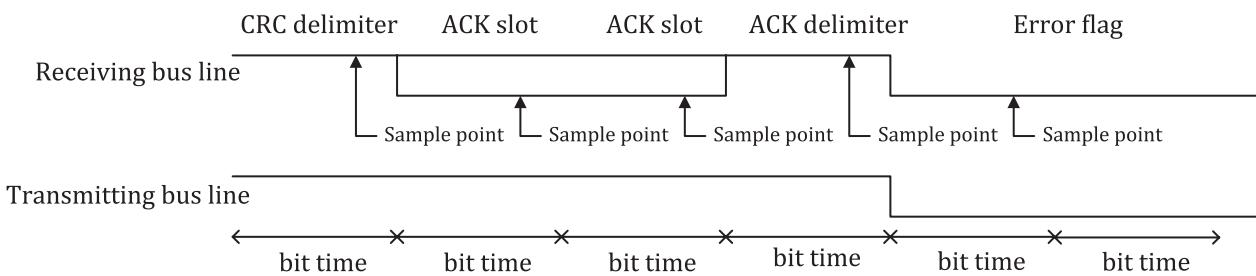


Figure 29 — CRC error in FD frames

When error signalling is enabled and a node detects an error during the arbitration phase, the node shall send an EF. When a node detects an error while the data phase bit rate is used, the node shall switch from the data bit time back into the nominal bit time of the arbitration phase before it starts the error flag. A transmitter using TDC (see [7.3.4](#)) shall notice a potential bit error at an SSP and it shall detect that error at the following SP.

When a node detects an error inside the XL data phase, the node shall finish the XL data phase at the end of the bit in which the error is detected.

When a node detects an error inside the FD data phase, the node shall finish the FD data phase after the SP where the error is detected.

For FD frames, a transmitter that uses TDC shall switch the bit time as specified in [Figure 30](#); a receiver (and a transmitter not using TDC) shall switch the bit time as specified in [Figure 31](#).

The shadings in [Figure 30](#) and [Figure 31](#) represent the parts of the bit time as shown in [Figure 32](#).

After detection of the error at the SSP, then, after the following SP and IPT, the bit timing is set back to nominal bit rate, but the bits during IPT are already counted for the Phase_Seg2, which shall pass before the next bit is started and before the error flag sending is started.

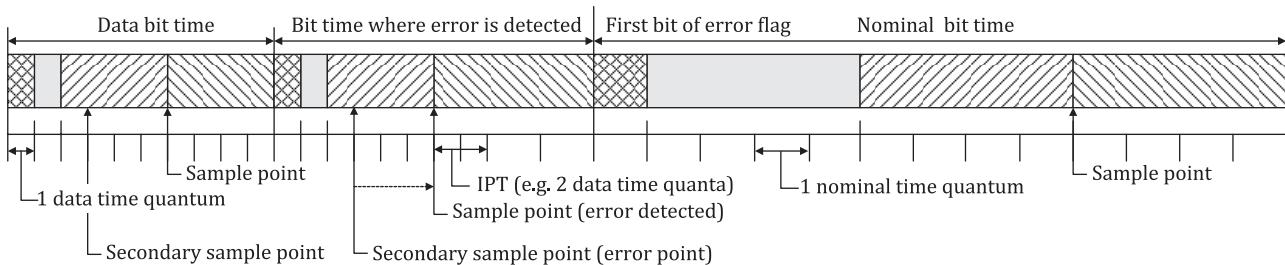


Figure 30 — Transmitter detects bit error at SSP in data phase of an FD frame

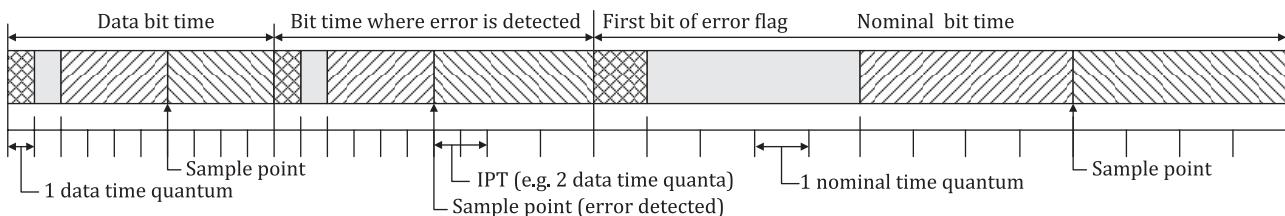


Figure 31 — Receiver detects error in data phase of an FD frame

6.6.21.3.2 Overload signalling

The following conditions shall lead to the transmission of an OF.

- LLC-requested OF (initiated by the LLC sub-layer): An LLC may cause an OF transmission, when internal conditions of the receiver need a delay of the next MAC DF or MAC RF.
- Reactive OF (initiated by the MAC sub-layer): Detection of a dominant bit during either of the first two bits of intermission, detection of one dominant bit at the last bit of EOF by a receiver, or detection of one dominant bit by any node at the last bit of error delimiter or overload delimiter shall cause an OF transmission.

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

At most, two LLC requested OFs shall be generated to delay the next MAC DF or MAC RF.

6.6.21.3.3 Restricted operation

When both, error signalling and restricted operation mode, are enabled, the restricted operation mode shall take precedence.

6.6.21.3.4 Error counting

The error counting rules as specified in [8.1.4](#) shall apply.

6.6.21.4 Behaviour when error signalling is disabled

6.6.21.4.1 General

When error signalling is disabled, the node shall transmit and receive only XLFF frames. It shall not transmit EF, OF and RF, nor DFs in CBFF, CEFF, FBFF and FEFF. A transmitter shall not perform arbitration during the IDE bit, FDF bit and XLF bit. In addition to the errors specified in [6.6.21.2](#), a receiver shall treat a recessive IDE bit, a dominant FDF bit or a dominant XLF bit as a form error.

6.6.21.4.2 Error handling

Transmitter errors

A transmitter shall ignore errors (see [6.6.21.2](#)) that occur in the range from SOF bit up to the AH2 bit, including the SOF bit and the AH2 bit. The transmitter shall calculate the PCRC and FCRC based on the transmitted bits, except for the bit, in which it loses arbitration. For the bit in which it loses arbitration, it shall use the received bit for the PCRC and FCRC calculation.

NOTE This means that from the SOF bit to the AH2 bit the transmitter ignores stuff errors on dynamic stuff bits and bit errors but can still lose arbitration when a recessive priority ID bit or the RRS bit is overwritten. When the transmitter loses arbitration after ignoring a preceding error, it can, then as a receiver, detect a PCRC error.

If the transmitter detects an ACK error, it shall remain transmitter and it shall transmit a recessive ACK delimiter bit followed by an EOF that shall consist of five recessive bits.

If the transmitter detects an error during the ACK delimiter bit or the EOF, then it treats this error as a protocol exception event (see [6.6.17.2](#)).

In the transmitter the MAC sub-layer shall inform the LLC sub-layer on any error that is not ignored.

Receiver errors

If the receiver detects an error during the following parts of the XLFF, it shall treat this as a protocol exception event: arbitration field, control field, data field, CRC field, ACK field, EOF. When a node detects such a protocol exception event inside the XL data phase, the node shall finish the XL data phase at the end of the bit, in which the event is detected. In case of an FCRC error, the receiver shall detect a protocol exception event at the DAH bit.

In the receiver the MAC sub-layer shall inform the LLC sub-layer on each detected error.

6.6.21.4.3 Overload signalling

A node shall treat any overload condition specified in [6.6.21.3.2](#) as a protocol exception event.

NOTE This means a node treats a dominant bit sampled during the first bit of the intermission and the second bit of the intermission as a protocol exception event. Receivers treat additionally a dominant bit sampled during the last bit of the EOF as a protocol exception event.

6.6.21.4.4 Restricted operation

If the error signalling is disabled in the restricted operation mode, the CAN node shall not transmit any DF.

6.6.21.4.5 Error counting

The error counting rules as specified in [8.1.4](#) shall not apply. The behaviour of the error counters is specified in [8.1](#).

7 PL specifications

7.1 General and functional modelling

The PL connects CAN nodes with a bus.

NOTE The number of nodes is limited by the electric loads on the bus and by the CAN data link layer protocol.

The PL is modelled according to ISO/IEC/IEEE 8802-3. It has three sub-layers.

- The PCS shall encompass functions related to bit encoding/decoding, timing and synchronization as well as bus failure detection. They are specified in the subsequent subclauses.

- b) The PMA sub-layer encompasses functional circuitry for the bus transmission/reception. It is not in the scope of this document.
- c) The PMD sub-layer encompasses the mechanical and electrical interfaces between the physical media and the PMA sub-layer. It is not in the scope of this document.

7.2 Services of the PCS interface

7.2.1 Requirements

The services of the PL shall allow the local MAC sub-layer entity to exchange bits with peer MAC sub-layer entities.

The PCS shall provide the following services to the MAC sub-layer:

- PCS_Data.Request;
- PCS_Data.Indicate;
- PCS_Mode.Request;
- PCS_Status.Transmitter;
- PCS_Status.Receiver.

7.2.2 PCS_Data.Request

The PCS_Data.Request service is passed from the MAC sub-layer to the PCS to request the transmission of a bit. The service shall provide the following parameter:

```
PCS_Data.Request(  
    Output_Unit  
)
```

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

The interpretation of the two values dominant or recessive depends on the operation mode of the CAN transceiver.

7.2.3 PCS_Data.Indicate

The PCS_Data.Indicate service is passed from the PL to the MAC sub-layer to indicate the arrival of a bit. The service shall provide the following parameter:

```
PCS_Data.Indicate(  
    Input_Unit  
)
```

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

7.2.4 PCS_Mode.Request

The PCS_Mode.Request service is passed from the MAC sub-layer to the PCS to indicate that the MAC sub-layer transmits or receives an XL frame. The service shall provide the following parameter:

```
PCS_Mode.Request(  
    XL_Mode  
)
```

The XL_Mode parameter shall take on one of the two values: active and passive.

When the MAC sub-layer transmits or receives an XL frame and when the CAN transceiver mode switching is enabled, XL_Mode shall be set to active from the XLF bit of a XL frame until the end of that frame, else it shall be set to passive.

7.2.5 PCS_Status.Transmitter

The PCS_Status.Transmitter service is passed from the MAC sub-layer to the PCS to indicate that the MAC sub-layer transmits the FD data phase of an FD frame or that part of an XL frame in which the CAN transceiver is switched into the data TX mode when the CAN transceiver mode switching is enabled. The service shall provide the following parameter.

```
PCS_Status.Transmitter(  
    D_Transmit  
)
```

The D_Transmit parameter shall take on one of the two values: active and passive.

In transmitted XL frames, D_Transmit shall be set to active at the beginning of the ADH bit and it shall be set to passive at the beginning of the DAH bit or when that frame is disturbed. In transmitted FD frames, D_Transmit shall be set to active at the beginning of the FD data phase and it shall be set to passive at the end of the FD data phase or when that frame is disturbed.

NOTE When the following expression is true, it means that the transceiver operates in data TX mode: (PCS_Mode.Request(XL_Mode) = active) AND (PCS_Status.Transmitter(D_Transmit) = active).

7.2.6 PCS_Status.Receiver

The PCS_Status.Receiver service is passed from the MAC sub-layer to the PCS to indicate that the MAC sub-layer receives the FD data phase of an FD frame or that part of an XL frame in which the transceiver is switched into the data RX mode when the CAN transceiver mode switching is enabled. The service shall provide the following parameter.

```
PCS_Status.Receiver(  
    D_Receive  
)
```

The D_Receive parameter shall take on one of the two values: active and passive.

In received XL frames, D_Receive shall be set to active at the beginning of the ADH bit and it shall be set to passive at the beginning of the DAH bit or when that frame is disturbed. In received FD frames, D_Receive shall be set to active at the beginning of the FD data phase and it shall be set to passive at the end of the FD data phase or when that frame is disturbed.

NOTE When the following expression is true, it means that the transceiver has to operate in data RX mode: (PCS_Mode.Request(XL_Mode) = active) AND (PCS_Status.Receiver(D_Receive) = active).

7.3 PCS

7.3.1 Bit encoding/decoding

7.3.2 Bit time

Bus management functions executed within the PCS, such as the CAN node synchronization behaviour, the compensation of the network transmission delay and the sample point positioning, are realized by the programmable bit timing logic of the CAN node.

CAN nodes shall support three different bit rates, the nominal bit rate, the FD data bit rate and the XL data bit rate. There shall be separate configuration register sets for the definition of these bit rates.

The XL data bit time shall only be used inside the XL data phase of an XL frame. The XL data phase shall start at the beginning of the DH1 bit inside the ADS. This XL data phase shall be finished at the beginning of the DAH bit inside the DAS, or when the CAN node detects an error condition that results in the starting of an EF (when error signalling is enabled) or is treated as a protocol exception event (when error signalling is disabled). Nodes that do not support the XL frame format do not include configuration registers for the XL data bit rate.

The FD data bit time shall only be used inside the FD data phase of an FD frame. Nodes that do not support the FD frame format do not include configuration registers for the FD data bit rate.

NOTE The FD data phase starts at the sample point of the BRS bit if that bit is detected to be recessive. This FD data phase is finished when the first sample point of the CRC delimiter is reached or when the CAN node sees an error condition that results in the starting of an EF. When the bit rate is switched because an error condition is detected, the switching time is shifted after the sample point, by less than or equal to two-time quanta.

The phase outside the FD data phase or the XL data phase is the arbitration phase. All CC frames, EFs, OFs, idle time, and all parts of FD frames with a dominant BRS bit shall belong to the arbitration phase. The nominal bit time shall be used inside the arbitration phase.

Time quantum

The time quantum (t_q) shall be a fixed unit of time derived from the node clock period. There shall exist a programmable prescaler, with integer values. The minimum time quantum ($t_{q,\min}$) shall be one node clock period long. The length of t_q shall be an integer multiple m of the minimum time quantum $t_{q,\min}$ with m as the value of the prescaler.

The length of a bit time depends on the length of the t_q and on the number of the t_q in the bit. If different parameter combinations come to the same length of the bit time, the combination with the shorter t_q results in a better synchronization of the nodes. Harmonization of bit time configurations in all nodes of the system is facilitated using harmonized node clocks. It is recommended to use node clock frequencies of 20 MHz, 40 MHz, or 80 MHz for CAN FD applications. For CAN XL applications with a data bit rate of up to 20 Mbit/s, 160 MHz are recommended.

If the following three properties are met, then the bit timing configuration is optimal with respect to bit rate switching for networks using FD frames:

- the same time quantum length is used in the nominal bit time and in the data bit time;
- the positions of the sample points in the nominal bit time are the same in all CAN nodes of a network;
- the positions of the sample points in the FD data bit time are the same in all CAN nodes of a network.

The frequencies of the node clock oscillators in the different CAN nodes shall be coordinated to provide a network-wide specified time quantum for the nominal bit time and, for FD or XL enabled nodes, for the data bit time. The acceptable oscillator tolerances of the CAN nodes (see [7.3.6](#)) and the potential for incorrect synchronization are determined by Phase_Seg1, Phase_Seg2, and SJW.

Bit rates and bit times

The bit rate gives the number of bits per second transmitted by a node in the absence of the resynchronization. There are relationships between bit rates and bit times: the nominal bit time is the multiplicative inverse of the nominal bit rate; the FD data bit time is the multiplicative inverse of the FD data bit rate; and the XL data bit time is the multiplicative inverse of the XL data bit rate.

All three bit times shall consist of separate non-overlapping time segments. The segments shall form the bit times as shown in [Figure 32](#).

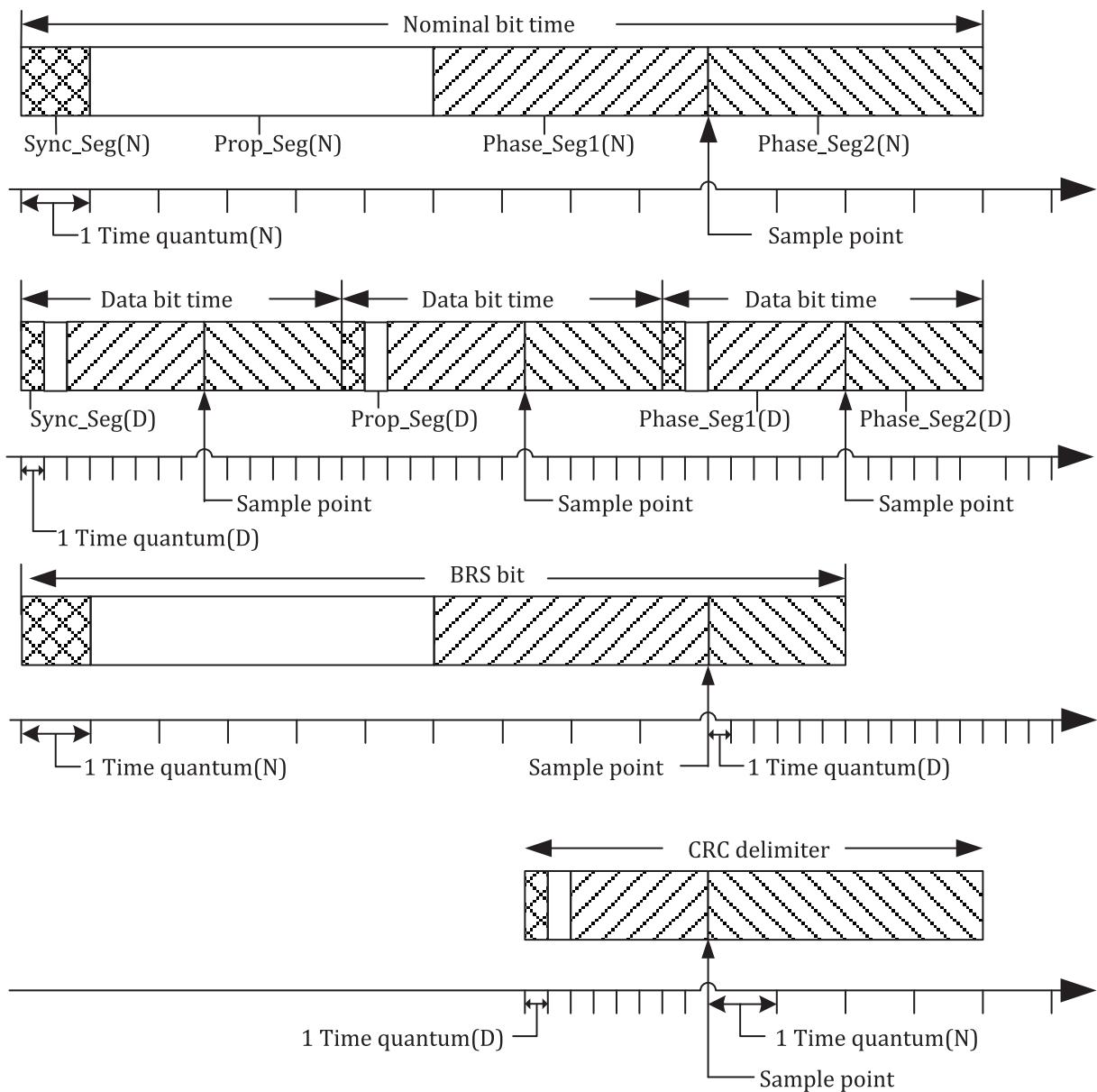


Figure 32 — Segments of nominal bit time and of data bit time

Sync_Seg

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

Prop_Seg

This part of the bit time, the propagation time segment, is used to compensate for 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, see [Figure 33](#).

Phase_Seg1, Phase_Seg2

These phase buffer segments 1 and 2 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 in 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 number of time quanta required for the calculation of the subsequent bit level. This calculation begins at the sample point and shall be less than or equal to Phase_Seg2 (see [7.3.5](#) for additional restrictions).

Synchronization 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 synchronization 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 CAN node. For more details see [Figure 33](#).

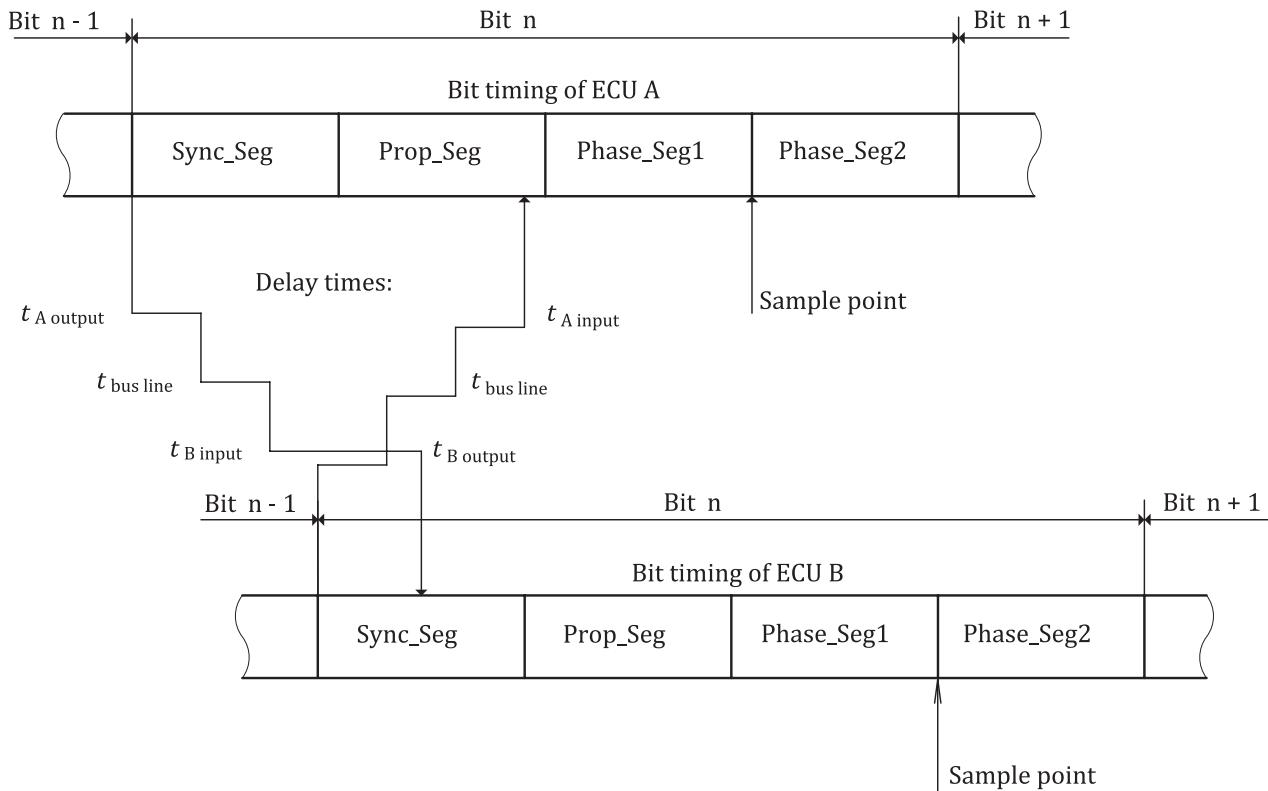


Figure 33 — Time relationship between delay times and bit time phases of CAN nodes A and B during arbitration phase

In connection with [Figure 33](#),

- the sum of output and input CAN node delays is critical relative to the configuration of the nominal bit time. The important characteristic parameter of a CAN node is given by [Formula \(1\)](#):

$$t_{node} = t_{output} + t_{input} \quad (1)$$

- for proper arbitration, the following conditions given by [Formula \(2\)](#) shall be met:

$$t_{\text{prop_seg}} \geq t_{\text{node_A}} + t_{\text{node_B}} + 2 \times t_{\text{busline}} \quad (2)$$

- 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.

7.3.3 Configuration of the bit time parameters

Configuration of the bit time shall be performed using the following periods of time: Sync_Seg, Prop_Seg, Phase_Seg1, Phase_Seg2, and SJW. All parameters are representing integer number of the t_q based on the $t_{q,\min}$ and the prescaler m. For nodes that support FD or XL frame formats, the configuration range of the SSP offset, representing integer number of the $t_{q,\min}$, is also specified.

The different ranges for the configuration of bit time segments are specified in [Table 11](#) to [Table 13](#), they depend on the chosen implementation option.

There shall be one prescaler m for the three sets of bit time configurations. There may be separate prescalers for the bit time configurations, but the prescalers should be configured to the same values. Except for the synchronization segment, which shall be exactly one t_q long, CAN nodes may allow time segments that exceed the minimum required configuration ranges specified in the tables.

When the bit time configuration of a CAN node is not programmable, its fixed bit time configuration shall meet the restrictions for this configuration as specified in this subclause.

Table 11 — Time segments minimum configuration ranges when FD frame format is not supported

Parameter	Nominal bit time
Prescaler m	1 to 32
Sync_Seg	1 t_q
Prop_Seg	1 to 8 t_q
Phase_Seg1	1 to 8 t_q
Phase_Seg2	2 to 8 t_q
SJW	1 to 4 t_q

Table 12 — Time segments minimum configuration ranges when XL frame format is not supported

Parameter	Separate prescalers	Shared prescalers	Separate or shared
	Nominal bit time	Nominal bit time	FD data bit time
Prescaler m	1 to 32		1 to 32
Sync_Seg	1 t_q	1 t_q	1 t_q
Prop_Seg	1 to 48 t_q	0 to 96 t_q	0 to 8 t_q
Phase_Seg1	1 to 16 t_q	1 to 32 t_q	1 to 8 t_q
Phase_Seg2	2 to 16 t_q	2 to 32 t_q	2 to 8 t_q
SJW	1 to 16 t_q	1 to 32 t_q	1 to 8 t_q
SSP offset	not applicable	not applicable	1 to 63 $t_{q,\min}$

For nodes that support the FD frame format but not the XL frame format, the values given in [Table 13](#) should be used. [Table 12](#) is provided to enable implementations of FD nodes that comply with ISO 11898-1:2015 to be compliant with this document.

Table 13 — Time segments minimum configuration ranges when all frame formats are supported

Parameter	Nominal bit time	FD data bit time	XL data bit time
Prescaler m	1 to 32		
Sync_Seg	$1 t_q$	$1 t_q$	$1 t_q$
Prop_Seg	1 to 384 t_q	0 to 128 t_q	0 to 128 t_q
Phase_Seg1	1 to 128 t_q	1 to 128 t_q	1 to 128 t_q
Phase_Seg2	2 to 128 t_q	2 to 128 t_q	2 to 128 t_q
SJW	1 to 128 t_q	1 to 128 t_q	1 to 128 t_q
SSP offset	not applicable	1 to 160 $t_{q,min}$	1 to 160 $t_{q,min}$

The following restrictions apply for the configuration of the bit time segments.

- The information processing time shall be less than or equal to $2 t_q$ long.
- In data bit time, Phase_Seg2 shall be greater than or equal to these two items: SJW and the maximum information processing time.
- In nominal bit time, Phase_Seg2 shall be greater than or equal to the maximum of these two items: SJW and the information processing time.
- In nominal bit time and in data bit time, SJW shall be less than or equal to the minimum of these two items: Phase_Seg1 and Phase_Seg2.

In case of synchronization, Phase_Seg1 can be longer and Phase_Seg2 can be shorter than its programmed value. The position of the sample point can differ in the three bit timing configurations; the length of the Prop_Seg can be zero in the configuration for the data bit rates.

An example for the nominal bit time configuration based on a prescaler of $m(N) = 3$ and the time segments Prop_Seg(N) = 5, Phase_Seg1(N) = 4, Phase_Seg2(N) = 4 combined with the data bit time configuration based on a prescaler of $m(D) = 1$ and the time segments Prop_Seg(D) = 1, Phase_Seg1(D) = 6, Phase_Seg2(D) = 6, as well as the resulting bits of intermediate length BRS and CRC delimiter, is shown in [Figure 32](#).

In a CAN node, Prop_Seg and Phase_Seg1 do not need to be programmable separately; it shall be sufficient to program the sum of Prop_Seg and Phase_Seg1. The total number of time quanta in a nominal bit time shall be programmable at least from 8 to 25 for nodes that are not FD enabled. For nodes that are FD enabled, the total number of time quanta in a data bit time shall be programmable at least from 5 to 25 and in a nominal bit time at least from 8 to 80. For nodes that are XL enabled, the total number of time quanta in a data bit time shall be programmable at least from 5 to 385 ($1 + 128 + 128 + 128$) and in a nominal bit time at least from 5 to 641 ($1 + 384 + 128 + 128$).

The XL data bit rate shall be at least two times the nominal bit rate.

7.3.4 Transmitter delay compensation

CAN nodes are connected to the CAN bus via an electrical circuit, the CAN transceiver, that unavoidably imposes a time delay between when a bit is transmitted at the protocol controller's output signal and when the same bit signal appears at the protocol controller's input signal. Without transmitter-delay compensation, the bit rates in the data phase of FD frames or XL frames are limited by the fact that the transmitter detects a bit error if it cannot receive its own transmitted bit latest at the sample point of that bit.

Nodes that support the FD frame format and nodes that support the XL frame format shall support a transmitter delay compensation mechanism. Transmitter delay compensation should be used in applications where the length of the bit time in the FD data phase or the XL data phase is shorter than or equal to 1 000 ns. Transmitter delay compensation needs to be used in applications where the sum of Sync_Seg, Prop_Seg, Phase_Seg1 in the FD data phase or the XL data phase is shorter than the propagation delay from the PCS to PMA output notification (see [7.4.2.1](#)) to the PMA to PCS input notification, as specified in other parts of the ISO 11898 series. Transmitters shall only use this mechanism in the XL data phase of XL frames when error

signalling is enabled or in the FD data phase of FD frames if the BRS bit is recessive. It shall be programmable whether the mechanism is used. When this mechanism is used, the value of the prescaler m shall be one or two.

The node shall be able to compensate transmitter delays of at least $95 t_{q,\min}$. The position of the SSP shall be configurable separately for XL frames and for FD frames.

Nodes that do not support the XL frame format may limit the range for transmitter delay compensation to 63 minimum time quanta.

Bit error detection shall be disabled for those bits at the end of the XL data phase or FD data phase in which the SSPs of the bits are in the following arbitration phase.

NOTE The latest SSP is $160 + 95 = 255 t_{q,\min}$ after the start of the transmitted bit, when the minimum configuration ranges are used. With these minimum configuration ranges, assuming an $t_{q,\min}$ duration of 6,25 ns (160 MHz), a transmitter delay of $95 t_{q,\min} \times 6,25 \text{ ns} / t_{q,\min} = 593,75 \text{ ns}$ can be compensated and the SSP offset can be set to $160 t_{q,\min} \times 6,25 \text{ ns} / t_{q,\min} = 1\,000 \text{ ns}$.

The transmitter delay compensation mechanism defines a secondary sample point SSP. When it is used, the transmitter shall ignore bit errors detected at the sample point. The received bit value shall be compared, at the SSP, with the (delayed) transmitted bit value. If a bit error is detected at the SSP, the transmitter shall react to this bit error at the subsequent following sample point. Bit error detection shall be disabled for those bits at the end of the data phase in which the SSPs of the bits would be in the following arbitration phase.

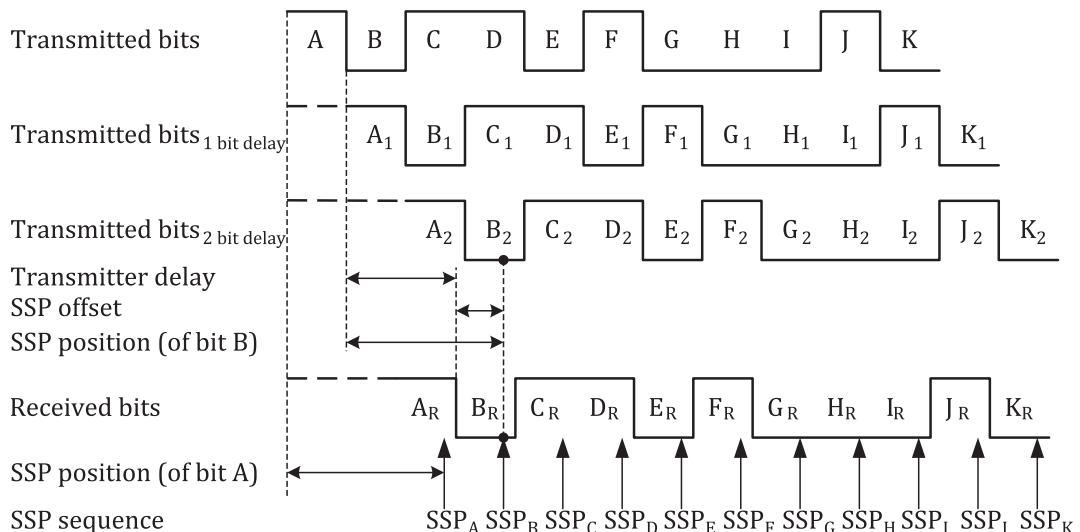


Figure 34 — Position of the secondary sample point

The position of the SSP is specified by its distance from the start of the bit time. The SSP position shall be set to a value derived from a measurement of the actual transmitter delay. The measurement of the transmitter delay shall be done in each transmitted frame at the recessive to dominant edge from the FDF bit to the res bit in FD frames and at the recessive to dominant edge from the XLF bit to the resXL bit in XL frames. Application of the measured value for the SSP position is done at the data phase of the same frame. A counter shall be started when the transmitter starts to transmit the dominant res bit in FD frames or the resXL bit in XL frames at the transmit output. The counter shall be incremented by one each minimum time quantum until the dominant signal is detected at the receive input; then the counter shall be stopped. The counter value is the measured transmitter delay time. The SSP position shall be the sum of the measured transmitter delay value and the configured SSP offset.

The example bit stream [A to K] in [Figure 34](#) shows the relation between the transmitter delay and the SSP position. In that example, the transmitter delay is almost two data bit times long. Therefore, the SSP is placed in the range from the transmitter delay up to three data bit times after the start of the bit; at a point where the input signal is expected to have stabilized. In this case, the bits from the received bit stream shall be compared to the bits from the transmitted bit stream that are delayed by two data bit times: At SSP_A, the received bit A_R is compared to the delayed transmitted bit A₂ and so on.

If SSP position value is an odd number and the value of the prescaler for the data time quantum is 2, then SSP position may be determined by dividing the SSP position value by 2 and rounding down the result.

[Figure 35](#) shows the end of the FD data phase with transmitter-delay compensation. The SSP sequence stops at the transmitter's sample point of the (first) CRC delimiter bit, where the data phase ends. The sample point of the CRC delimiter is still part of the data phase. In that example, the last CRC sequence bit before the CRC delimiter is locally disturbed (received bit E_R is seen as recessive but was transmitted as dominant). The transmitter tolerates a second recessive CRC delimiter bit before it sees the dominant ACK. There would be an EF instead of the ACK if the disturbance of bit E were not locally limited.

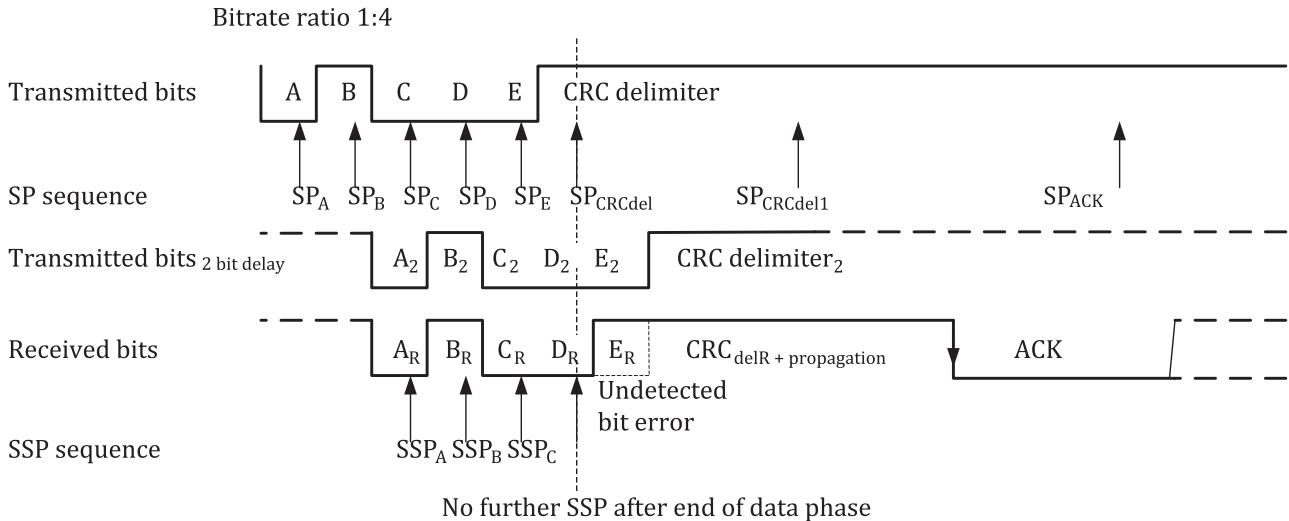


Figure 35 — End of transmitter delay compensation phase

CAN nodes may choose to continue the SSP sequence beyond the limit of the data phase, checking for local errors at the transmitting node as shown in [Figure 36](#).

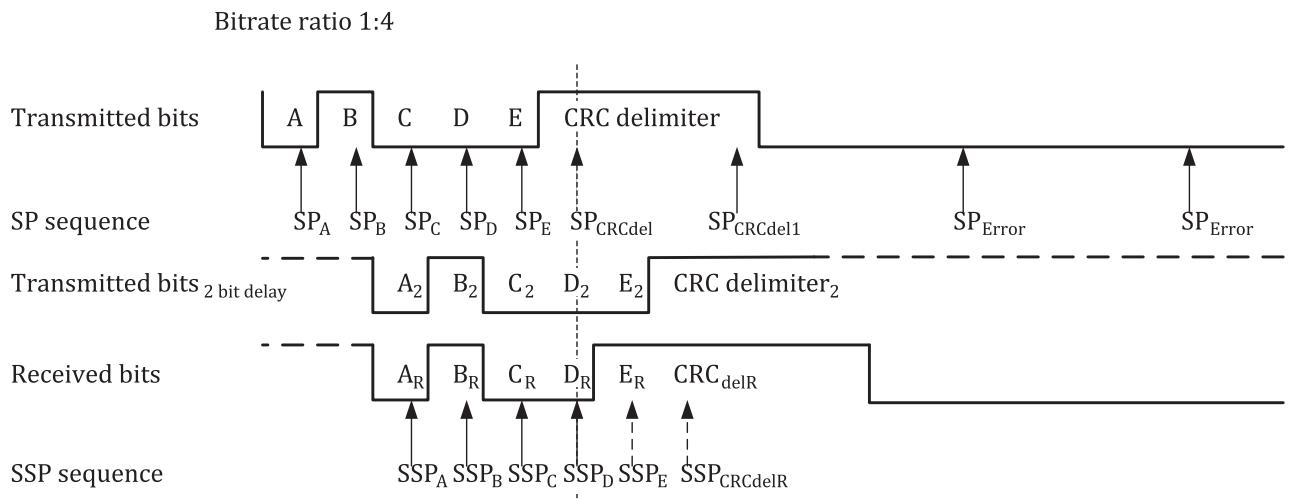


Figure 36 — Optional continuation of SSP sequence

7.3.5 Synchronization

7.3.5.1 Description

The state machine synchronizing the operation of the CAN node to the PCS_Data.Indicate shall operate in time steps of one t_q . At each time quantum, it shall be analysed whether the bus state is recessive or dominant. The bus state detected at the sample point of a bit shall be regarded as the value of that bit. A difference in

bus states between two consecutive time quanta is an edge. A receiving node that is synchronized to an undisturbed frame detects edges only in the Sync_Seg of a bit time. Edges detected outside the Sync_Seg may cause the CAN node to synchronize its operation to that edge.

NOTE 1 The bus comparator converting the physical signals into data signals can operate independently from the node clock. The time needed to synchronize the data signals to the node clock before they can be regarded by the clocked state machines is part of the CAN node's input delay time (see [Figure 33](#)).

Hard synchronization and resynchronization are 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. After an edge is detected, synchronizations shall be disabled until the next time the bus state detected at the sample point is recessive.
- b) An edge shall cause synchronization only if the bus state detected at the previous sample point (previous read bus state) was recessive. If a transmitter uses the transmitter delay compensation (see [7.3.4](#)), also the first detected edge from recessive to dominant after the sample point of the CRC delimiter shall cause synchronization. In this case, CAN nodes may choose alternatively to synchronize on the first detected edge from recessive to dominant seen by an amount of at least the transmitter delay plus one time quantum after the sample point of the CRC delimiter. An edge with a positive phase error (see [7.3.5.2](#)) shall not cause synchronization in a node sending a dominant bit.
- c) Hard synchronization shall be performed:
 - on edges during inter-frame space (with the exception of the first bit of intermission);
 - when a node is in bus-integration state;
 - at the edge between the FDF bit and the following dominant res bit inside an FD frame;
 - when a node is a receiver of an XL frame, at the edge between the XLF bit and the following dominant resXL bit;
 - at the edge between the DH1 and DH2 bits and the DL1 bit;
 - at the edge between the DAH or AH1 bit and the AL1 bit.

A transmitter of an XL frame shall perform a hard synchronization at an edge after the sample point of the FDF bit and before the sample point of the XLF bit when error signalling is enabled.

- d) All other recessive to dominant edges fulfilling rules a) and b) shall be used for resynchronization with one exception: A node transmitting an FD frame or an XL frame shall not synchronize while it transmits the data phase of that frame. Clocking information shall be derived from transitions from one bit value to the other. The property that (due to the bit stuffing) only a fixed maximum number of successive bits have the same value shall provide the possibility of resynchronizing a CAN node to the bit stream during a frame.

NOTE 2 A transmitter that loses arbitration on the XLF bit eliminates all phase shift to the winning transmitter by the hard synchronization to this dominant bit.

7.3.5.2 Phase error of an 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 the following:

- $e = 0$ if the edge lies within Sync_Seg;
- $e > 0$ if the edge lies between the Sync_Seg and the sample point of the current bit;
- $e < 0$ if the edge lies between the sample point of the current bit and the Sync_Seg of the following bit.

7.3.5.3 Hard synchronization

After a hard synchronization, the bit time shall be restarted by each bit-timing logic unit with Sync_Seg completed. Thus, hard synchronization shall force the edge which has caused the hard synchronization to lie within the synchronization segment of the restarted bit time. The function of hard synchronization shall not be limited by the synchronization jump width.

CAN nodes may perform edge filtering during the bus re-integration state. Two consecutive nominal time quanta with dominant bus state is required to detect an edge that causes the bit counter for re-integration to be reset when this option is enabled by configuration. When edge filtering is performed, dominant bus-states shorter than two nominal time quanta shall not cause a reset of that counter.

Nodes that do not support the XL frame format may implement this low-pass edge filter also in the signal path to the hard-synchronization.

7.3.5.4 Bit resynchronization

Resynchronization leads to a shortening or lengthening of the bit time to correct the position of the sample point in relation to the position of the detected edge. When the magnitude of the phase error of the edge which causes resynchronization is less than or equal to the programmed value of the synchronization jump width, the effect of a resynchronization shall be the same as a hard synchronization.

When the magnitude of the phase error is larger than the programmed value of the synchronization jump width,

- and if the phase error e is positive, Phase_Seg1 shall be lengthened by an amount equal to the synchronization jump width;
- and if the phase error e is negative, then Phase_Seg2 shall be shortened by an amount equal to the synchronization 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.

NOTE Since synchronization is only enabled when the bus state is read recessive at the sample point and at most only one synchronization is allowed between two sample points, only edges from recessive to dominant are relevant for synchronization. The first edge is used for synchronization, after that, synchronization is disabled, even if that edge does not cause synchronization, e.g. because it is detected inside the Sync_Seg.

Neither receiver nor transmitter shall perform a synchronization at an edge after the sample point of the ADH bit and before the sample point of the DH1 bit.

7.3.6 Tolerance range of the oscillator frequencies

The tolerance of a node clock oscillator frequency f_{osc} around the nominal frequency f_{nom} shall be given by the range $(1 - df) \times f_{nom} \leq f_{osc} \leq (1 + df) \times f_{nom}$. The tolerance df depends on the length of the t_q , the segments of the bit time t_{bit} , and on the SJW. The maximum difference between the node clock oscillators of any two nodes is expected to be $2 \times df \times f_{nom}$.

The maximum tolerance df of f_{osc} shall meet the following conditions:

$$df < \frac{SJW(N)}{2 \times 10 \times t_{bit}(N)} \quad (3)$$

$$df < \frac{\min[Phase_Seg1(N), Phase_Seg2(N)]}{2 \times [13 \times t_{bit}(N) - Phase_Seg2(N)]} \quad (4)$$

$$df < \frac{SJW(D)}{2 \times 10 \times t_{bit}(D)} \quad (5)$$

$$df < \frac{\min[Phase_Seg1(N), Phase_Seg2(N)]}{2 \times \left[[6 \times t_{bit}(D) - Phase_Seg2(D)] \times \frac{m(D)}{m(N)} + 7 \times t_{bit}(N) \right]} \quad (6)$$

$$df < \frac{SJW(D) - \max\left(0, \frac{m(N)}{m(D)} - 1\right)}{2 \times \left[[2 \times t_{bit}(N) - Phase_Seg2(N)] \times \frac{m(N)}{m(D)} + Phase_Seg2(D) + 4 \times t_{bit}(D) \right]} \quad (7)$$

where

(N) indicates the values for the arbitration phase;

(D) indicates the values for the data phase.

The conditions given in [Formulae \(3\)](#) and [\(4\)](#) shall be met for CC frames.

All conditions of [Formulae \(3\)](#) to [\(7\)](#) shall be met for FD frames.

The [Formulae \(3\)](#) and [\(4\)](#) shall be met for XL frames, together with the [Formula \(8\)](#):

$$df < \frac{SJW(X)}{2 \times (2S + 1) \times t_{bit}(X)} \quad (8)$$

whereby (X) indicates the values for the XL data phase and S is the stuff rate (see [6.6.13.3.2](#)).

The configured SJW shall not be larger than the smaller one of the Phase_Seg1 and Phase_Seg2.

NOTE The configured Prop_Seg limits the part of the bit time that can be used for the Phase_Seg1 and Phase_Seg2.

7.4 Attachment unit interface

7.4.1 General

This interface is between the PCS as specified in [7.3](#) and the PMA sub-layer specified in other ISO standards (e.g. in ISO 11898-2 and ISO 11992-1).

7.4.2 PCS to PMA symbols

7.4.2.1 Output symbol

The PCS shall send an output symbol to the PMA sub-layer when it receives an Output_Unit from the MAC sub-layer. The output symbol causes the PMA sub-layer to transmit a dominant (level_0) or recessive (level_1) voltage level to the medium.

7.4.2.2 Bus_off symbol

The PCS shall send a bus_off symbol to the PMA sub-layer, when it receives a bus_off_request from the supervisor (see [8.1.3.4](#)).

7.4.2.3 Bus_off_release symbol

The PCS shall send a bus_off_release symbol to the PMA sub-layer when it receives a bus_off_release_request from the supervisor (see [8.1.3.4](#)).

7.4.2.4 Mode symbol

The PCS shall send a mode symbol to the PMA sub-layer when it receives an XL_Mode from the MAC sub-layer. The mode symbol causes the PMA sub-layer to select on one of the two values:

- CAN transceiver mode switching enabled for XL frame, and
- CAN transceiver mode switching disabled.

7.4.2.5 D_Transmit symbol

The PCS shall send a D_Transmit symbol to the PMA sub-layer when it receives a D_Transmit from the MAC sub-layer.

NOTE This symbol is not mandatory for CAN CC and CAN FD nodes.

7.4.2.6 D_Receive symbol

The PCS shall send a D_Receive symbol to the PMA sub-layer when it receives a D_Receive from the MAC sub-layer.

NOTE This symbol is not mandatory for CAN CC and CAN FD nodes.

7.4.3 PMA to PCS symbol

The PMA shall send an input symbol to the PCS when it receives a dominant (level_0) or recessive (level_1) voltage level from the medium. The input signal indicates to the PCS the arrival of a dominant (level_0) or recessive (level_1) bit.

7.5 PWM encoding

7.5.1 General function and definitions

The PWM encoding function encodes the PCS signals towards the AUI allowing to control the physical behaviour of the PMA sub-layer implicitly through the output bit stream. The CAN SIC XL transceiver, as specified in ISO 11898-2, supports two different bus driver and bus comparator characteristics depending on the phases of the protocol as specified in [5.16](#).

During the XL data phase, the PMA sub-layer can activate a push-pull PMA bus driver characteristic (level_0 and level_1) instead of the dominant and recessive characteristic. The receiver thresholds are changed accordingly. This mechanism is called CAN transceiver mode switching and can be enabled or disabled as specified in [6.6.12.3](#).

The CAN transceiver mode switching is initiated and maintained through the PWM encoding function. A PWM symbol encoding with a certain period time that lies within the minimum and maximum of the PWM detection time $t_{SymbolNom}$ as specified in ISO 11898-2:2024, Table 12, is used to switch the PMA sub-layer from the arbitration mode with the dominant-and-recessive characteristic towards the data TX mode or the data RX mode with the level_0 and level_1 bit driven through a push-pull characteristic.

The PWM symbol consists of a short phase and a long phase. The nominal PWM symbol length is defined between two similar edges accumulating a short phase and a long phase.

PWM symbols for the transmitter start and end with rising signal edges on the AUI, as defined in [Figure 37](#). PWM symbols for the receiver start and end with falling signal edges on the AUI, as defined in [Figure 38](#).

The bit level driven by the transmitter is defined through the ratio between the length of the high and the length of the low phase of the PWM symbol, as specified in [Figure 37](#) and [Figure 38](#). A physical level_1 equals a logical 1 and corresponds to a longer high phase compared to the low phase within one PWM symbol time (PWM_1 symbol). A physical level_0 equals a logical 0 and corresponds to a longer low phase compared to the high phase within one PWM symbol time (PWM_0 symbol).

The PWM symbol time is an integer multiple of the $t_{q,\min}$ and is independent of a configured bit rate prescaler. PWM coding guidelines are given in the CiA 612-2 document.

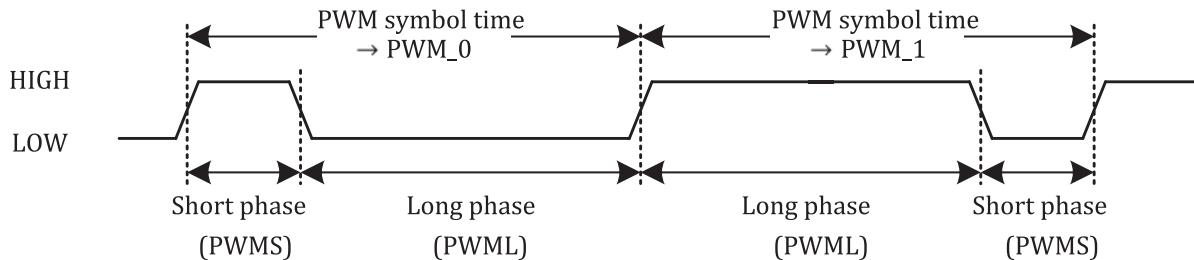


Figure 37 — PWM symbol coding definitions, example for the transmitter

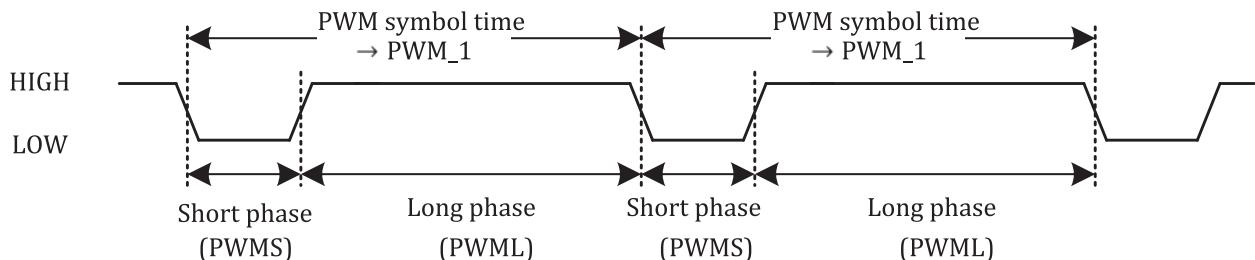


Figure 38 — PWM symbol coding definitions, example for the receiver

7.5.2 PCS without PWM encoding

In case the CAN transceiver mode switching is disabled, or the PCS does not request a mode change via the service PCS_Mode.Request, PCS_Status.Transmitter and PCS_Status.Receiver, PCS output symbols shall be directly forwarded to the PMA sub-layer.

7.5.3 PCS sub-layer with PWM encoding

7.5.3.1 PWM encoding activation and deactivation

In case the CAN transceiver mode switching is enabled and the service parameter PCS_Mode.Request (XL_Mode) is active, the signalling from the PCS to the PMA sub-layer shall be PWM-encoded for a transmitter while the service parameter PCS_Status.Transmitter (D_Transmit) is passed as active as specified in [7.2.5](#) and for a receiver while the service parameter PCS_Status.Receiver (D_Receive) is passed as active as specified in [7.2.6](#).

7.5.3.2 Internal processing delay

The PWM-encoded signal towards the PMA sub-layer shall have an internal processing delay of 0 $t_{q,\min}$ or 1 $t_{q,\min}$ relative to the DLL bit boundaries. The internal processing delay shall be of constant length and shall impact identically both, the activation and the deactivation of the PWM encoding. This delay is compensated by the receiving nodes synchronizing on the edge to the DL1 bit.

If a processing delay of 1 $t_{q,\min}$ is used, then the signal towards the PMA sub-layer shall remain at its value for this 1 $t_{q,\min}$ between the signalling of the activation and the actual activation of the PWM encoding. The processing delay of 1 $t_{q,\min}$ after deactivation of the PWM encoding shortens DAH bit by 1 $t_{q,\min}$.

7.5.3.3 PWM configuration

The PWM short and long phases are integer multiples of the one $t_{q,\min}$. PWM symbol lengths of 200 ns (nominal length) or shorter than shall be supported. The required minimum PWM configuration depth n depends on the length of one $t_{q,\min}$ and shall follow [Formula \(9\)](#):

$$n > t_{\text{PWM_nom_max}} \times \frac{1}{t_{q,\min}} \quad (9)$$

The length of $t_{\text{PWM_nom_max}}$ is 200 ns.

The PWM short phase length (PWMS) shall be configurable supporting a range of at least 1 to $n-1$ $t_{q,\min}$. The PWM long phase length (PWML) shall be configurable supporting a range of at least 1 to $n-1$ $t_{q,\min}$. The PWM symbol length is the sum of PWMS and PWML.

The parameter PWM time offset (PWMO) shall be configurable supporting a range of at least 0 $t_{q,\min}$ to $n-1$ $t_{q,\min}$ used to delay the start of the PWM encoding at the beginning of the ADH bit compensating for odd bit length relations between the arbitration phase and the XL data phase for transmitter. PWMO shall always be smaller than the sum of PWMS and PWML.

For the transmitter, PWMO shall be filling the time ahead of the first PWM symbol within the ADH bit in case the ADH bit length is not able to be assembled out of an integer number of nominal PWM symbols; PWMO shall be zero in case the ADH bit time is able to be assembled out of an integer number of nominal PWM symbols. [Figure 39](#) specifies the details. The receivers shall not make use of PWMO and shall ignore this parameter, as specified in [Figure 40](#).

NOTE 1 The CiA 612-2 document provides recommendations for the depth n for each PWM parameter (PWMS, PWML, PWMO) and the guidance for the configuration.

NOTE 2 The XL date bit rate of 5 Mbit/s corresponds to the XL date bit time of 200 ns. For multiple PWM symbols per bit the short and long phases of the PWM symbol are configured accordingly. [Figure 41](#) shows an example where multiple PWM symbols are used per XL data bit time. Further configuration and timing examples are given in the CiA 612-2 document.

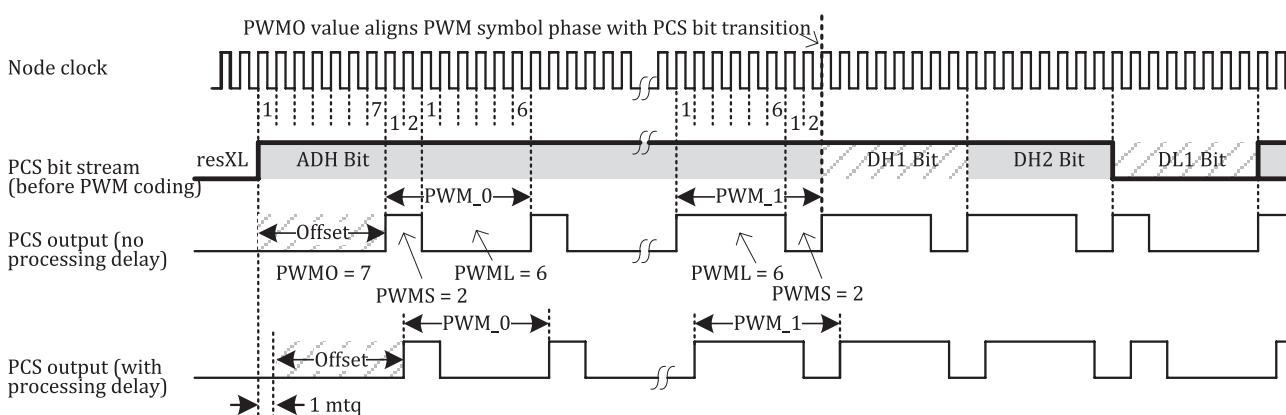


Figure 39 — Example for PWM configuration for transmitters

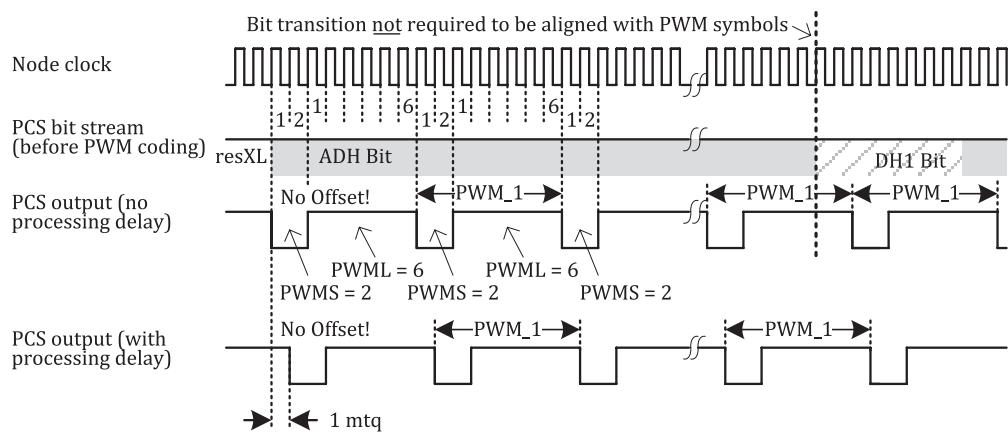


Figure 40 — Example for PWM configuration for receivers

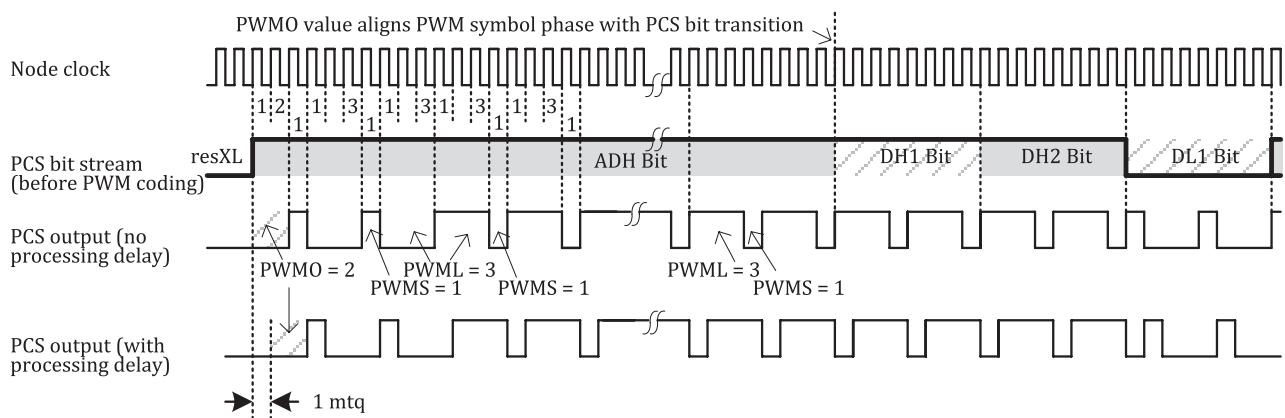


Figure 41 — Example for PWM configuration for transmitters for multiple PWM symbols per data bit

7.5.3.4 Transmitter

With the reception of the active service parameter PCS_Status.Transmitter (D_Transmit) and PCS_Mode.Request (XL_Mode) the PWM encoding shall output a logical 0 towards the PMA sub-layer for one PWM offset time.

With expiration of the PWM offset time the PWM encoding in the PCS sub-layer shall drive two consecutive PWM_0 symbols. From there onwards the PWM symbols shall follow the PCS output symbols. [Figure 39](#) and [Figure 41](#) show the examples.

If the service parameter PCS_Status.Transmitter (D_Transmit) is passive or if the service parameter PCS_Mode.Request (XL_Mode) is passive, the PCS output symbols shall be directly forwarded to the PMA sub-layer regardless of the actual PWM phase.

7.5.3.5 Receiver

With reception of the active service parameter PCS_Status.Receiver (D_Receive) and PCS_Mode.Request (XL_Mode) the PWM encoding shall drive consecutive PWM_1 symbols without a leading PWMO. Resynchronization events in the PCS shall not influence the PWM symbol timing. [Figure 40](#) shows an example.

If the service parameter PCS_Status.Receiver (D_Receive) is passive or if the service parameter PCS_Mode.Request (XL_Mode) is passive the PCS output symbols shall be directly forwarded to the PMA sub-layer regardless of the actual PWM phase.

8 Description of supervisor FCE

8.1 Fault confinement

8.1.1 Objectives

The objective of fault confinement is to preserve a high availability of the data transmission network even in the presence of a defective node. Therefore, the fault confinement strategies shall prove reliable on the following instances:

- a) distinction between temporary errors and permanent failures;
- b) localization and switching-off of faulty nodes.

Fault confinement shall be enabled only when error signalling is enabled.

When error signalling is disabled within a network, the node shall never enter error passive state or bus-off state as specified in [8.1.4](#).

NOTE If error signalling is disabled, error counters are not maintained and fault confinement is not possible in the FCE. Higher-layer functions can, if required, be used to implement fault confinement, for example, by maintaining error counters and reacting on the values of the counters.

8.1.2 Strategies

All nodes shall include a transmit error counter and a receive error counter. The transmit error counter shall register the number of errors during the transmission, and the receive error counter shall register the errors during the reception of frames.

When frames are sent or received correctly, the counters shall be decremented. When frames are sent or received with errors, the counters shall be incremented more than they are decremented in the absence of errors. The ratio in which the counters are incremented/decremented depends on the acceptable ratio of invalid/valid frames on the bus. At any time the levels of the error counters reflect the relative frequency of previous errors.

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

8.1.3 Fault confinement interface specification

8.1.3.1 Description

The interface between the fault confinement entity and the OSI sub-layer entities is shown in [Figure 42](#).

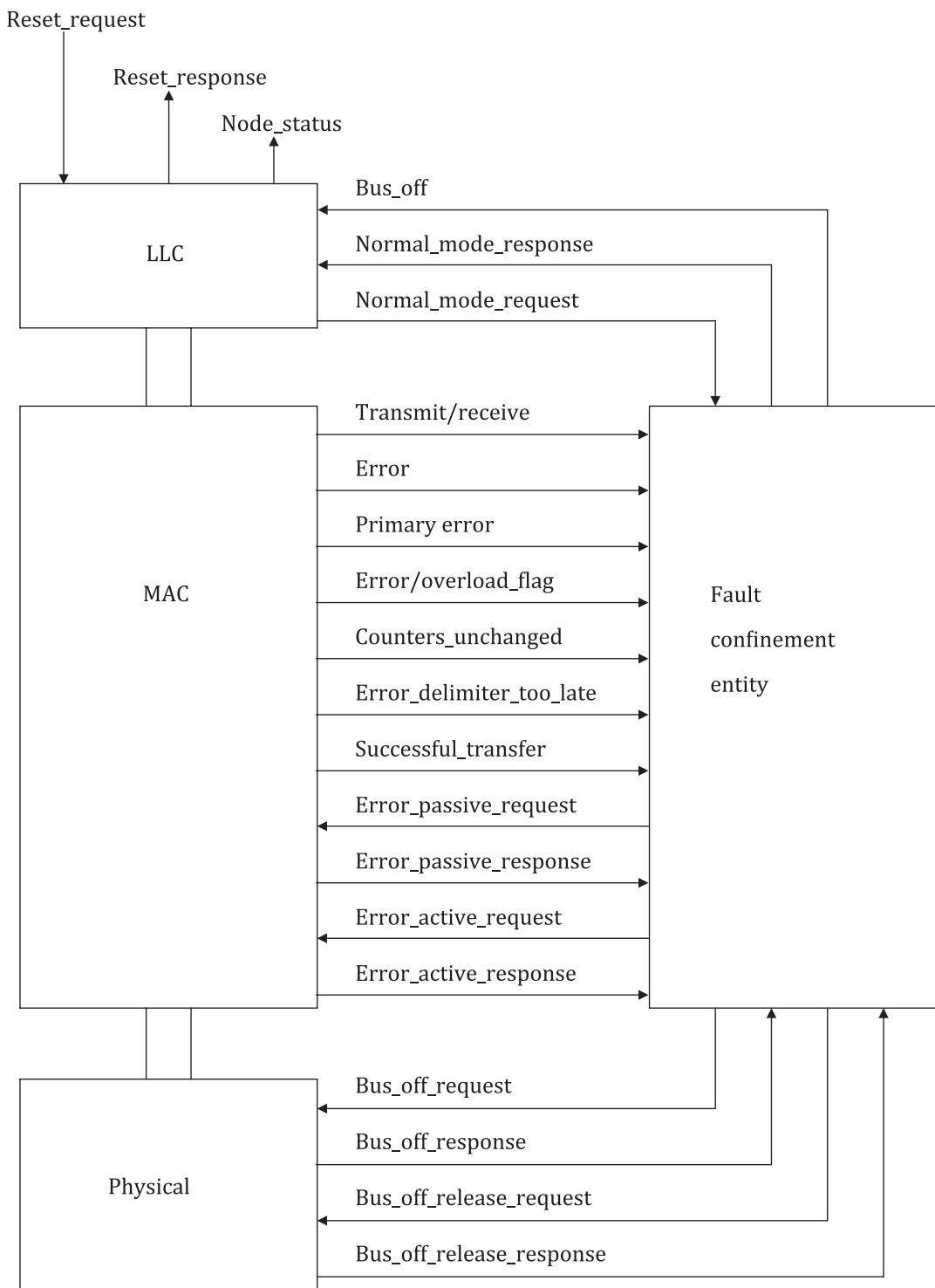


Figure 42 — Fault confinement entity interface to the OSI sub-layer entities

8.1.3.2 LLC sub-layer/FCE interface

[Table 14](#) specifies the LLC-to-FCE service interface and [Table 15](#) specifies the FCE-to-LLC service interface.

Table 14 — LLC-to-FCE service interface

Service	Description
Normal_mode_request	Resets FCE to initial state (the values of the transmit error counter and the receive error counter are set to zero when the FCE passes into its initial state).

Table 15 — FCE-to-LLC service interface

Service	Description
Normal_mode_response	Response to the Normal_mode_request.
Bus_off	Indicates that the node is in the bus-off state.

8.1.3.3 MAC sub-layer/FCE interface

[Table 16](#) specifies the MAC-to-FCE service interface and [Table 17](#) specifies the FCE-to-MAC service interface.

Table 16 — MAC-to-FCE service interface

Service	Description
Transmit/receive	Indicates the node's current transfer mode.
Error	Indicates that the MAC sub-layer has detected an error (bit error, stuff error, CRC error, form error, ACK error).
Primary_error	Signals that the MAC sub-layer has detected a dominant bit after sending an error flag (indicates that the MAC sub-layer 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 sub-layer is sending an error flag or an overload flag.
Counters_unchanged	Indicates that the FCE counters remain unchanged (due to special cases; see rule c) in 8.1.4.2).
Error_delimiter_too_late	Indicates that the MAC sub-layer is waiting too long for error delimiter. This signal is set 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.

Table 17 — FCE-to-MAC service interface

Service	Description
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.

8.1.3.4 PL/FCE interface

[Table 18](#) specified the PCS-to-FCE service interface and [Table 19](#) specifies the FCE-to-PCS service interface.

Table 18 — FCE-to-PL service interface

Service	Description
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.

Table 19 — PL-to-FCE service interface

Service	Description
Bus_off_response	Response to bus_off_request.
Bus_off_release_response	Response to bus_off_release_request.

8.1.4 Rules of fault confinement

8.1.4.1 Description

With respect to fault confinement, a node can be in one of the three states, depending on the error counter levels (see [5.10](#) and [5.12](#)):

- error-active;
- error-passive;
- bus-off.

8.1.4.2 Error counting

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

- a) When a receiver detects an error, the receive error counter shall be incremented by 1, except when the detected error is 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 incremented by 8.
- c) When a transmitter sends an error flag, the transmit error counter shall be incremented by 8.
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.
Exception 2:
If the transmitter sends an error flag because a stuff error occurred before the arbitration is decided, whereby the stuff bit should have been recessive, and after it is sent recessive but is monitored to be dominant. In exception 1 and in exception 2, 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 incremented by 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 incremented by 8.
- f) Any node shall tolerate up to 7 consecutive dominant bits after sending an active error flag, passive error flag, or overload flag. After detecting 14 consecutive dominant bits (in case of an active error flag or an overload flag) or after detecting 8 consecutive dominant bits following a passive error flag, and after each sequence of additional 8 consecutive dominant bits, every transmitter shall increment its transmit error counter by 8 and every receiver shall increment its receive error counter by 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 decremented by 1 unless it was already 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 decremented by 1 if it was between 1 and 127. If the receive error counter was 0, it shall stay at 0, and if it was greater than 127, then it shall be set to a value between 119 and 127.

8.1.4.3 Transition between error-active and error-passive states

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

An error condition causing a node to 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 127 (see [Figure 43](#)).

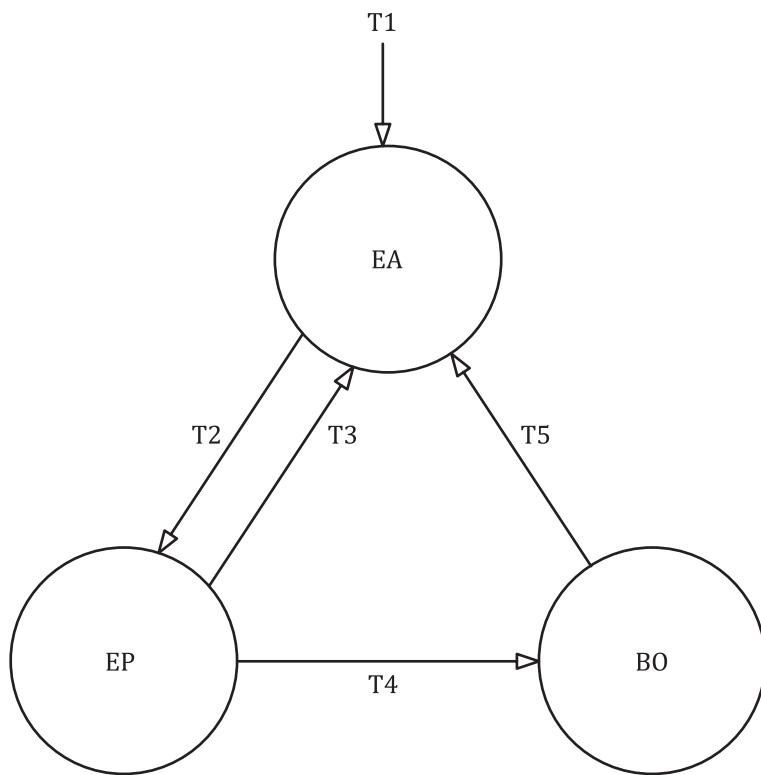
When the receive error counter of a node exceeds the error-passive limit of 127, further increments of this receive error counter shall be limited by the width of the counter. At the next successful reception of a frame (transition to error-active), the receive error counter shall be set to a value below the error-passive limit [see error counting rule h) in [8.1.4.2](#)].

8.1.4.4 Bus-off management

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

A node which is in the bus-off state shall have no influence on the bus. It shall not send any frames nor acknowledge DFs or RFs. Whether or not such a node accepts DFs from the bus depends on the implementation.

Upon a restart request, a node which is in the bus-off state shall be re-integrating to the CAN communication (see [6.6.7.5](#)) and may become error-active (no longer bus-off) with its error counters both set to zero after having monitored 128 occurrences of the idle condition (see [3.34](#)) on the bus. [Figure 43](#) shows the node status transitions.

**Key**

EA	error active
EP	error passive
BO	bus-off
T1	init (Normal_Mode_Request)
T2	receive error counter > 127 OR transmit error counter > 127
T3	receive error counter < 128 AND transmit error counter < 128
T4	transmit error counter > 255
T5	Normal_Mode_Request AND 128 occurrences of 11 consecutive recessive bits

Figure 43 — Node status transition diagram**8.1.5 Node start-up**

If during network start-up only one node is online and if this node transmits any DF or RF, it does not get an ACK. In this case, it shall detect an error and repeat the frame. According to rule c) in [8.1.4.2](#), exception 1, it may become error-passive but shall not go bus-off.

When a switched-off node is switched on again, it shall:

- synchronize with already available nodes to re-integrate into the bus communication (see [6.6.7.5](#)) before starting to transmit. Synchronization is achieved when 11 recessive bits have been detected that are equivalent to ACK delimiter + EOF + intermission or Error/Overload delimiter + intermission;
- wait for other nodes if there is no other node available at the moment without becoming bus-off.

8.2 Bus failure management

During normal operation, several bus failures can occur that influence the bus operation. These failures and the resulting behaviour of the network shall be in accordance with the used PMA.

Annex A

(normative)

CAN FD light — Data link layer and physical coding sub-layer requirements of responder nodes

A.1 Basic concepts of CAN FD light

A.1.1 General

A CAN FD light responder node is an implementation based on a subset of the CAN FD data link layer functionality that provides the data link layer services as well as the attachment unit interface and that implements the data link layer protocol as well as the physical coding sub-layer specified in this document. A commander node is a node that sends data frames to responder nodes to initiate a CAN FD light communication.

NOTE System design recommendations are provided in CiA 604-3.

Responder nodes receive and transmit CAN data frames in FBFF. The responder node has the following properties:

- no arbitration;
- no bit rate switching;
- acceptance filtering;
- no remote frame (RF);
- no error frame (EF);
- no overload frame (OF);
- no automatic retransmission of frames;
- fixed values for bits in the control field.

A.1.2 Bus access method

Responder nodes transmit DFs on request received in DFs.

A.1.3 ACK

Responder nodes send the ACK bit after error-free reception of a DF in the ACK field as specified in [6.6.14](#). The sending of the ACK bit may be disabled by node configuration.

A.1.4 Error detection and signalling

The responder node supports error detection and handling as specified in [A.2.2.11](#), but does not detect overload conditions and does not signal errors. This means that the responder node does not send EFs and OFs. An error counter is not implemented. In case of error conditions, the responder node triggers the protocol exception event (see [A.2.2.9.3](#)).

Network wide data consistency is not supported. Data consistency between commander node and responder node is achieved by the commander/responder communication scheme.

A.2 Data link layer requirements of the responder node

A.2.1 Requirements of LLC sub-layer

A.2.1.1 Services of LLC sub-layer

A.2.1.1.1 General

Responder nodes shall implement the unacknowledged data transfer service of the LLC sub-layer, as specified in [6.4.5](#), and shall not implement the unacknowledged remote data request service.

A.2.1.1.2 Unacknowledged data transfer service

Responder nodes shall support the LLC data frame in FBFF. Other type of frames shall not be supported.

A.2.1.2 Functions of LLC sub-layer

Responder nodes shall support the frame acceptance filtering as specified in [6.5.4](#), shall not support the overload notification and shall not support the recovery management as specified in [6.5.5](#) and [6.5.6](#).

A.2.1.3 Structure of LLC frames

Responder nodes shall implement the LLC frames as shown in [Table A.1](#).

Table A.1 — LLC frame

Field	Size	Function
Identifier	11 bit	base identifier
DLC	4 bit	data length code, as specified in Table 5
LLC data	0 to 64 byte	data content of the DFs

NOTE Since all transmitted and received frames are in FBFF and the bits ESI=0 and BRS=0, all bits in the Format field are fixed. Therefore, the Format field is omitted.

A.2.2 Requirements of MAC sub-layer

A.2.2.1 Services of MAC sub-layer

Responder nodes shall implement the acknowledged data transfer service of the MAC sub-layer as specified in [6.6](#) and shall not implement the acknowledged remote data request service and OF transfer service.

A.2.2.2 Functional model of MAC sub-layer architecture

A.2.2.2.1 Functions

Responder nodes shall implement the following subset of the functions of the MAC sub-layer in [A.2.2.2](#) and shall not support the functions that are not in the subset.

A.2.2.2.2 Frame transmission

- a) Transmitting data encapsulation consists of:
 - acceptance of LLC frames and interface control information;
 - CRC sequence calculation including stuff bit count for FBFF;

- construction of MAC frame by adding SOF, IDE bit, RRS bit, FDF bit, res bit, BRS bit, ESI bit, CRC, ACK and EOF to the LLC frame.
- b) Transmitting medium access management consists of:
 - serialization of the MAC frame;
 - insertion of stuff bits (bit stuffing);
 - presentation of a serial bit stream to the PL for transmission.

A.2.2.2.3 Frame reception

- a) Receiving medium access management consists of:
 - reception of a serial bit stream from the PL;
 - deserialization and recompiling of the MAC frame structure;
 - deletion of stuff bits (bit de-stuffing);
 - error detection (CRC, stuff bit count check, format check, stuff rule check);
 - transmission of ACK.
- b) Receiving data de-capsulation consists of:
 - removing the MAC-specific information from the received frame;
 - presenting the LLC frame and interface control information to the LLC sub-layer.

A.2.2.3 Structure of MAC frames

A.2.2.3.1 Description

Responder nodes shall implement the frame type DF that carries data from a transmitter and shall not support the RFs, EFs and OFs.

Responder nodes shall support the following type of DFs:

- DFs in FBFF.

A.2.2.3.2 Requirements of MAC DF

A.2.2.3.2.1 SOF

Responder nodes shall support SOF as specified in [6.6.8](#). SOF shall mark the beginning of DFs in FBFF.

A.2.2.3.2.2 Arbitration field

Responder nodes shall support the relevant specification of FBFF in [6.6.11.2](#).

A.2.2.3.2.3 Control field

Responder nodes shall support the relevant specification of FBFF in [6.6.11.3](#).

BRS bit

Responder nodes shall transmit the BRS bit as dominant.

ESI bit

Responder nodes shall transmit the ESI bit as dominant.

A.2.2.3.2.4 Data field

The MAC frame data field shall be equivalent to the LLC data field as required in [A.2.1.3](#).

A.2.2.3.2.5 CRC field

Responder nodes shall support the 17-bit CRC for DFs with up to 16 data-field byte and 21-bit CRC for DFs with 20 to 64 data-field byte as specified in [6.6.11.5](#).

A.2.2.3.2.6 ACK field

Responder nodes shall support the ACK field as specified in [6.6.11.6](#).

A.2.2.3.2.7 EOF

DFs in FBFF shall be delimited by the EOF as in [6.6.11.7](#).

A.2.2.4 Specification of inter-frame space

A.2.2.4.1 Definition

The inter-frame space for DFs is as defined in [6.6.7, Figure 7](#) for error passive nodes, which is the transmitter of the previous frame. The interframe space shall consist of the intermission field, the suspend transmission field and bus idle.

NOTE This inter-frame definition applies regardless, whether the responder node is transmitter or receiver previously.

A.2.2.4.2 Intermission field

The intermission field for DFs is as defined in [6.6.7.2](#) with the difference that the intermission field shall consist of zero number of bits.

A.2.2.4.3 Bus idle

Responder nodes shall support the bus idle as defined in [6.6.7.3](#).

The period of bus idle is of arbitrary length. The bus shall be recognized to be idle when the seventh bit of EOF is seen recessive or when the re-integration condition is detected as defined in [A.2.2.9.2](#) during bus integration state.

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

A.2.2.4.4 Suspend transmission

Responder nodes shall suspend the start of a frame transmission for six-bit-times following the intermission field.

If a responder node detects a dominant bit on the bus during suspend transmission it shall interpret it as SOF.

NOTE 1 The responder recognizes the end of the frame when it detects eight consecutive recessive bits, starting with the ACK delimiter. The eight bits are the ACK delimiter and the seven EOF bits. The next dominant bit on the bus is therefore the SOF bit of the next frame. After the recognition of the EOF a responder node with a pending transmission request waits for additional six bits to account for a potential misalignment between the commander node and the responder node caused by clock frequency deviations.

NOTE 2 A commander node that conforms with this document inserts three bits of the intermission field after sending a frame. Therefore, it sends at least eleven recessive bits in a row.

A.2.2.5 Frame coding

Responder nodes shall support the frame coding as specified in [6.6.13](#).

A.2.2.6 Frame ACK

Responder nodes shall check the consistency of all received DFs and shall acknowledge all consistent DFs. ACK shall not depend on the frame's identifier. The ACK may be disabled (see [Clause A.5](#)).

A.2.2.7 Frame validation

DFs shall be valid for a receiver if there is no error until bit(0) of the CRC sequence. Transmitters shall not validate any DF.

A.2.2.8 Order of bit transmission

Responder nodes shall transfer the supported DFs in FBFF as specified in [6.6.16](#).

A.2.2.9 Medium access method

A.2.2.9.1 Bus access

Responder nodes shall not support arbitration and shall transmit DFs on request. Any pending transmission shall be cancelled in case of a reception of an SOF or an error as specified in [A.2.2.11.2](#).

NOTE A responder node transmits after the reception of a request from the commander node. Any interruption by another frame or an error leads to dismiss the transmission of a DF.

A.2.2.9.2 Bus integration state

Responder nodes shall enter the bus integration state after starting the protocol operation, or after detecting the protocol exception event. Responder nodes shall leave the bus integration state when the re-integration condition is detected. The re-integration condition shall be the detection of a sequence of eight consecutive sampled recessive bits.

When synchronization occurs, the search for the re-integration condition shall be restarted.

A.2.2.9.3 Protocol exception event

Responder nodes detect a protocol exception event as specified in [A.2.2.11](#). As reaction to the protocol exception event, the responder node shall send recessive bits and shall enter the bus integration state.

A.2.2.9.4 Collision handling

Responder nodes shall not support collision detection.

NOTE A responder node being the transmitter does not monitor the bus as specified in [A.2.2.11.2](#).

A.2.2.10 MAC data consistency

Responder nodes shall support the MAC data consistency as specified in [6.6.18](#).

A.2.2.11 Error handling

A.2.2.11.1 General

[A.2.2.11](#) specifies the error detection and the error signalling mechanisms of responder nodes.

NOTE Responder nodes do not detect an overload condition.

A.2.2.11.2 Error detection

Responder nodes shall support the following mechanisms for detecting errors as specified in [6.6.21.2](#):

- stuff rule check;
- stuff count check in FD frames;
- 17-bit CRC for FBFF with up to 16 data-field bytes and 21-bit CRC for FBFF with 20 to 64 data-field bytes.

Responder nodes shall also support the following mechanism:

- frame format check.

Responder nodes shall support the following three different error types, which are not mutually exclusive:

- a) stuff error,
- b) CRC error,
- c) form error.

A receiver shall detect a form error when it detects a recessive IDE bit or a dominant FDF bit. A receiver shall detect a form error when it detects a recessive BRS bit.

Exception: Responder nodes shall accept any value of the ESI bit, if the CRC value is correct.

NOTE 1 Responder nodes do not detect bit error and ACK error.

NOTE 2 The additional format error conditions are added because CAN FD light uses only FBFF.

NOTE 3 The format is checked for the entire frame even after a CAN FD light DF is accepted as valid after bit(0) of the CRC sequence. Detected errors are treated according to [A.2.2.11.3](#)

A.2.2.11.3 Error handling and signalling

The responder node shall trigger the protocol exception event (see [A.2.2.9.3](#)) and shall not send a response, in case it detects an error. Responder nodes shall not support error signalling via EFs.

NOTE 1 The CAN FD light commander node recognizes at which node the errors have occurred because of the commander/responder communication scheme.

NOTE 2 Additional error handling is possible on higher-layers.

NOTE 3 The way how the error handling and signalling is specified tolerates other responder node implementations to support other DFs than FBFF.

A.2.2.12 Overload signalling

Responder node shall not support overload signalling by means of OFs.

NOTE According to [A.2.2.4.2](#) no overload condition is possible.

A.3 Physical layer requirements of the responder node

A.3.1 Services of PL

Responder nodes shall implement the services of the PL as specified in [7.2](#).

A.3.2 PCS specification

A.3.2.1 Bit encoding/decoding

Since responder nodes implement the BRS bit as dominant, all parts of the frame belong to the arbitration phase. Only the nominal bit time, and therefore only time quantum (N) is used.

Responder nodes shall support the nominal bit time as specified in [7.3](#) and shall not support requirements based on the need for proper arbitration, because responder nodes do not arbitrate.

Responder nodes shall support the time segments' minimum configuration ranges for the nominal bit time as specified in [Table A.2](#) and may support a fix time segment configuration.

Table A.2 — Time segments' minimum configuration ranges

Parameter	Nominal bit time
<i>Sync_Seg</i>	1 time quantum (N)
<i>Prop_Seg</i>	1 to 8 time quanta (N)
<i>Phase_Seg1</i>	1 to 16 time quanta (N)
<i>Phase_Seg2</i>	2 to 16 time quanta (N)
<i>SJW</i>	1 to 16 time quanta (N)

SJW = min (*Phase_Seg1*, *Phase_Seg2*) should be used. In this case the parameter *SJW* does not need to be configurable.

A.3.2.2 Synchronization

A.3.2.2.1 Requirements

Responder nodes shall support the synchronization mechanisms specified in [7.3](#). Resynchronization may be used. In case resynchronization is not supported, hard synchronization shall be performed instead of resynchronization.

For implementation simplicity only hard synchronization should be applied.

A.3.2.2.2 Tolerance range of the oscillator frequencies

The tolerance of a responder node clock oscillator frequency f_{osc} around the nominal frequency f_{nom} shall be given by the range

$$(1-df)*f_{\text{nom}} \leq f_{\text{osc}} \leq (1+df)*f_{\text{nom}} \quad (\text{A.1})$$

The tolerance df depends on the length of the time quantum, the segments of the bit time and the synchronization jump width.

The maximum difference between the responder node and the commander node clock oscillator shall be

$$(df_{\text{commander}} + df_{\text{responder}})*f_{\text{nom}} \quad (\text{A.2})$$

where $df_{\text{commander}}$ is the frequency tolerance of the commander node and $df_{\text{responder}}$ is the frequency tolerance of the responder node.

The maximum tolerance $df_{\text{responder}}$ of f_{osc} shall meet the following condition:

$$df_{\text{responder}} < \frac{SJW}{10 * t_{\text{bit}}(N)} - df_{\text{commander}} \quad (\text{A.3})$$

or in case only hard synchronization is supported:

$$Phase_Seg_min = \min(Phase_Seg1, Phase_Seg2) \quad (\text{A.4})$$

$$df_{\text{responder}} < \frac{Phase_Seg_min}{10 * t_{\text{bit}}(N)} - df_{\text{commander}} \quad (\text{A.5})$$

SJW shall not be larger than the smaller one of $Phase_Seg1$ and $Phase_Seg2$. $Prop_Seg$ limits the part of the bit time that is used for $Phase_Seg1$ and $Phase_Seg2$.

NOTE Additional effects are possible to reduce the required frequency tolerance further, e.g. ringing effect, bit asymmetry.

A.3.3 AUI specification

A.3.3.1 PCS to PMA symbols

Responder nodes shall support output symbols as specified in [7.4.2](#).

A.3.3.2 PMA to PCS symbol

Responder nodes shall support an input symbol as specified in [7.4.3](#).

A.4 Description of supervisor fault confinement

Responder nodes shall not support the fault confinement as specified in [8.1.3](#).

Responder nodes shall not support the node status transition diagram as specified in [Figure 43](#) and the error counter as specified in [8.1.4](#).

A.5 Disabling ACK and transmitter synchronization

Sending the dominant ACK bit and all re-synchronizations and hard synchronizations for transmitting nodes can be disabled.

NOTE This option allows to support higher bit rates, for example.

Annex B (informative)

Configuration interface

A CAN node, except a CAN FD light responder node, provides configuration interfaces for the following features:

- the bit timing and transmitter delay compensation configuration as described in [7.3.2](#);
- enabling or disabling of the protocol exception handling as described in [6.6.17.2](#);
- enabling or disabling of the restricted operation mode as described in [6.6.19](#) (mandatory only for nodes that support the XL frame format);
- enabling or disabling of the optional bus monitoring operation mode as described in [6.6.20](#);
- enabling or disabling of the optional time stamping function as described in [6.5.7](#);
- enabling or disabling of error signalling as described in [5.8](#) (only for nodes that support the XL frame format);
- enabling or disabling of switchable operating modes of the PMA as described in [5.16](#) (only for nodes that support the XL frame format);
- the PWM configuration as described in [7.5.3.3](#) (only for nodes that support the XL frame format);
- limitation of retransmission attempts as described in [6.5.3](#);
- enabling or disabling of edge filtering for re-integration as described in [7.3.5.3](#).

Annex C (informative)

Additional information

C.1 Differences between CC frames, FD frames and XL frames

There are four implementation options for the support of frame formats:

- a) support for the CC frame format only, not tolerating the FD frame format (CAN CC);
- b) support for the CC frame format and tolerating the FD frame format (CAN FD tolerant);
- c) support for the CC frame format and for the FD frame format (CAN FD enabled);
- d) support for the CC frame format, for the FD frame format, and for the XL frame format (CAN XL enabled).

Option d) is recommended to be implemented for new designs.

The difference between implementation options a) and b) is the protocol exception event that causes the FD tolerant node to enter the bus integration state when a recessive FDF bit is detected.

The differences between implementation options a) and c) concern the frame formats, bit rates, CRC mechanisms, acknowledgement, and the transmitter delay compensation.

The differences between implementation options c) and d) concern the frame formats, bit rates, CRC mechanisms, error signalling and the transmitter operating mode switching.

C.2 Implementation hints

This document specifies the CAN communication on the CAN bus. This document does not specify the interface between an implementation of the CAN protocol and a host controller. This allows integration of CAN nodes into simple sensor or actuator nodes, as well as into micro-controllers or as stand-alone protocol controllers.

If the implementation allows to change the configuration of a node by software, the configuration data (e.g. bit time configuration, operating mode) should be locked against changes while the CAN communication is ongoing.

Nodes that do not support the XL frame format have implementation options for the functions listed in [Table C.1](#).

Table C.1 — Options for nodes that do not support the XL frame format

Function	Described in subclause
Retransmission counter	6.5.3
Restricted operation	6.6.19
Arbitration on FDF bit	6.6.11.2
Time segments minimum configuration ranges	7.3.3
Transmitter delay compensation range	7.3.4
Edge filtering during integration	7.3.5.3

Bibliography

- [1] ISO/IEC 8802-2, *Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part 2: Logical link control*
- [2] ISO/IEC/IEEE 8802-3, *Information technology — Telecommunications and exchange between information technology systems — Requirements for local and metropolitan area networks — Part 3: Standard for Ethernet*
- [3] ISO 11898-2:2024, *Road vehicles — Controller area network (CAN) — Part 2: High-speed physical medium attachment (PMA) sublayer*
- [4] ISO 11992-1:2019, *Road vehicles — Interchange of digital information on electrical connections between towing and towed vehicles — Part 1: Physical and data-link layers*
- [5] CiA 612-1, CAN XL guidelines and application notes – Part 1: System design recommendations
- [6] CiA 612-2, CAN XL guidelines and application notes - Part 2: CAN XL PWM coding guideline
- [7] CiA 604-3, CAN FD light – Part 3: System design recommendations



ICS 43.040.15

Price based on 82 pages