INTERNATIONAL STANDARD

ISO 11783-3

Fourth edition
2018-11

Tractors and machinery for agriculture and forestry — Serial control and communications data network —

Part 3: Data link layer

Tracteurs et matériels agricoles et forestiers — Réseaux de commande et de communication de données en série —

Partie 3: Couche liaison de données

Reference number
ISO 11783-3:2018(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

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

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 documents 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).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

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 23, *Tractors and machinery for agriculture and forestry*, Subcommittee SC 19, *Agricultural electronics*.

This fourth edition cancels and replaces the third edition (ISO 11783-3:2014), which has been technically revised. The main changes compared to the previous edition are as follows:

— updates wording with respect to ISO 11898-1 (exclude the usage of CAN Flexible Data Rate);

— allows BAM.TP to be sent with 10 ms;

— ACKNOWLEDGEMENT PG supports Extended Identifier Type when Request2 utilizes it.

A list of all parts in the ISO 11783 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

ISO 11783 specifies a communications system for agricultural equipment based on the ISO 11898-1 CAN protocol. SAE J 1939 documents[1)], on which parts of ISO 11783 are based, were developed jointly for use in truck and bus applications and for construction and agriculture applications. Joint documents have been completed to allow electronic units that meet the truck and bus SAE J 1939 specifications to be used by agricultural and forestry equipment with minimal changes. General information on ISO 11783 can be found in ISO 11783-1.

The purpose of ISO 11783 is to provide an open, interconnected system for on-board electronic systems. It is intended to enable electronic control units (ECUs) to communicate with each other, providing a standardized system.

1) Society of automotive engineers.

Tractors and machinery for agriculture and forestry — Serial control and communications data network —

Part 3: Data link layer

1 Scope

This document specifies the application, the network layer protocols and the mapping to the controller area network (CAN) data link layer protocol as specified in ISO 11898-1. The application layer specifies protocol data units (PDU), which can be mapped to Classical CAN data frames using the Classical Extended Frame Format (CEFF). For PDUs exceeding the length of the CEFF-formatted data frames, this document specifies transport layer protocols and the mapping to CEFF-formatted data frames.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 11783-1, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 1: General standard for mobile data communication*

ISO 11783-5, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 5: Network management*

ISO 11783-7, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 7: Implement messages application layer*

ISO 11898-1, *Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical signalling*

ISO 15765-2, *Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) — Part 2: Transport protocol and network layer services*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 11783-1 and ISO 11898-1 apply.

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

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

4 General description

The data link layer enables the reliable transfer of data across the physical link. This consists of sending the CAN classical data frame with the necessary synchronization, sequence control, error control and flow control. The flow control is accomplished through a consistent message frame format.

5 Technical requirements

5.1 Message frame format

5.1.1 General

The message frame format shall conform to the CAN requirements. The CAN specification referenced throughout this document is specified in ISO 11898-1. When there are differences between the CAN specification and this document, this document shall be the governing document.

The CAN document specifies, in an information-routing-related discussion, that control function addresses are not used. While this is true for some applications of CAN, it is not true for ISO 11783. The definition of the ISO 11783 network requires that control function addressing be used to prevent multiple control functions from using the same CAN identifier field. Many additional requirements exist in ISO 11783 that are not specified by CAN.

ISO 11898-1 specifies two classical frame formats: Classical Basic Frame Format (CBFF) and CEFF. ISO 11898-1 compatibility implies that messages of both formats can potentially be present on a single network, by using certain bit coding which allows for the recognition of the different formats. Up to this point, ISO 11783 also accommodates both message frame formats. However, ISO 11783 only defines a full strategy for standardized communications using the CEFF. All CBFF messages are for proprietary use following the rules defined in this document. Any FD Frame Format shall not be used on the ISO 11783 network.

ISO 11783 controllers shall therefore use the CEFF. CBFF messages may reside on the network, but only in accordance with this document.

The classical CAN data frame is parsed into different bit fields, as shown in Figure 1. The number and parsing of the bits in the arbitration and control field differs between the CBFF and CEFF messages. CBFF messages, as shown in Figure 1 a), contain 11 identifier bits in the arbitration field, whereas the arbitration field of CEFF messages, as shown in Figure 1 b), contain 29 identifier bits. ISO 11783 has further defined the identifier bits in the arbitration field of the CAN message frame formats. These definitions are given in Table 1.

5.1.2 Message frame format according to ISO 11783 (ISO 11898-1 CEFF)

The CEFF message, illustrated by Figure 1, encompasses a single protocol data unit (PDU). The PDU consists of seven predefined fields, assimilated from information provided by the application layer:

— Priority;

— Extended Data Page (EDP);

— Data Page (DP);

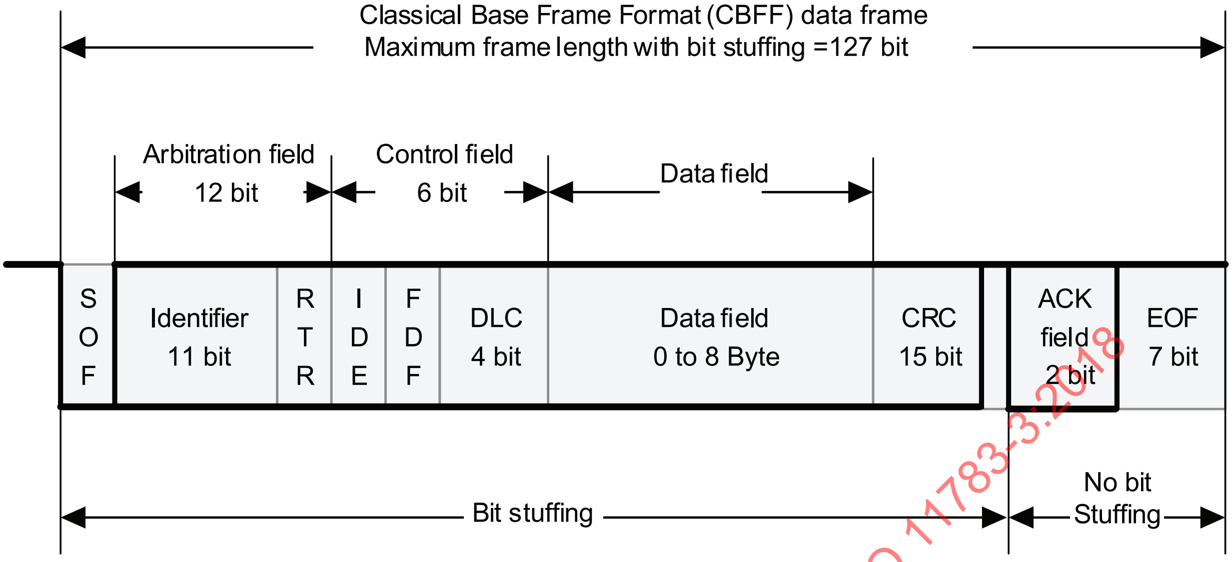
— PDU Format (PF),

— PDU Specific (PS), which can be Destination Address (DA), Group Extension (GE) or proprietary;

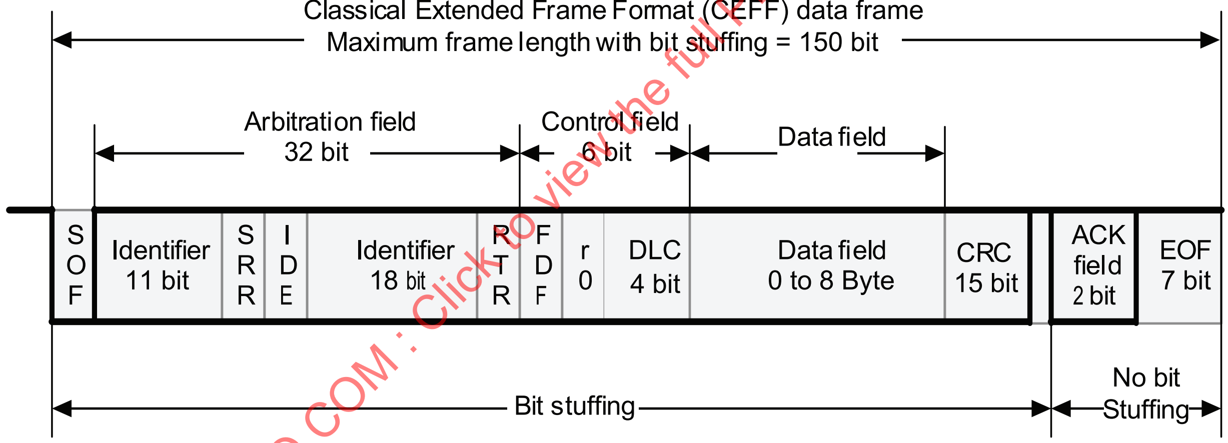
— Source Address (SA);

— Data.

See 5.2 for a detailed description of each field and 5.3 for PDU formats.



a) Classical Base Frame Format (CBFF)



b) Classical Extended Frame Format (CEFF)

Figure 1 — Classical CAN data frames

The fields are then packaged into one classical CAN data frame and sent over the physical media to other network controllers. The layers of the OSI model that ISO 11783 supports are shown in Figure 2. It is possible that some parameter group definitions require more than one classical CAN data frame in order to send their information.

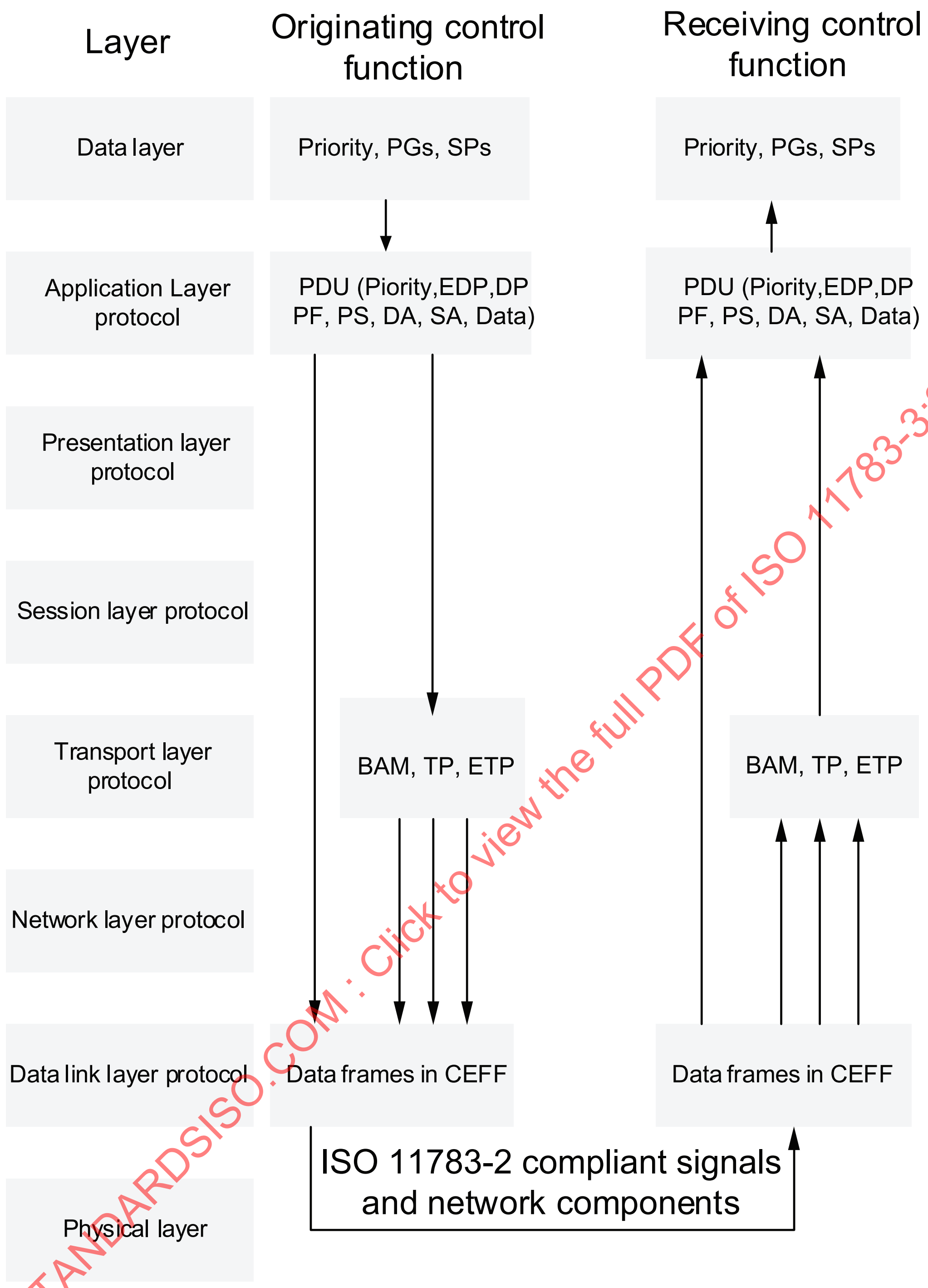


Figure 2 — Application of OSI model according to ISO 11783

Table 1 shows the arbitration and control fields of the 29 bit identifier for CAN, 29 bit identifier for ISO 11783 and 11 bit identifier for CAN, and the use of the 11 bit identifier on an ISO 11783 network. A complete definition for each of the bit field assignments according to ISO 11783 is given in 5.3. In ISO 11783, the CAN data frame data field is described as Bytes 1 to 8. Byte 1's MSB (most significant bit), Bit 8, is the first bit sent closest to the data length code (DLC). Byte 8's LSB (least significant bit), Bit 1, is the last of the data bits to be sent and is closest to the cyclic redundancy check (CRC) field. See Figure 3.

When the extended data page (EDP) is equal to 1 and the data page (DP) is equal to 1, the CAN frame is identified as an ISO 15765-2 formatted frame. ISO 15765-2 specifies diagnostic communication over CAN (DoCAN). Therefore, the processing of this specific CAN frame format does not follow the definitions specified in ISO 11783 and shall be in accordance with ISO 15765-2 (see 5.2.4).

Table 1 — Mapping of ISO 11783 into CAN arbitration and control fields

Bit number	29 bit identifier		11 bit identifier	
	CAN	ISO 11783	CAN	ISO 11783[b]
1	SOF	SOF[a]	SOF	SOF[a]
2	ID28	P3	ID28	P3
3	ID27	P2	ID27	P2
4	ID26	P1	ID26	P1
5	ID25	EDP	ID25	ID8[a]
6	ID24	DP	ID24	ID7[a]
7	ID23	PF8	ID23	ID6[a]
8	ID22	PF7	ID22	ID5[a]
9	ID21	PF6	ID21	ID4[a]
10	ID20	PF5	ID20	ID3[a]
11	ID19	PF4	ID19	ID2[a]
12	ID18	PF3	ID18	ID1[a]
13	SRR (r)	SRR[a]	RTR (x)	RTR[a] (d)
14	IDE (r)	IDE[a]	IDE (d)	IDE[a]
15	ID17	PF2	FDF (d)	FDF[a]
16	ID16	PF1	DLC4	DLC4
17	ID15	PS8	DLC3	DLC3
18	ID14	PS7	DLC2	DLC2
19	ID13	PS6	DLC1	DLC1
20	ID12	PS5		
21	ID11	PS4		
22	ID10	PS3		
23	ID9	PS2		
24	ID8	PS1		
25	ID7	SA8		
26	ID6	SA7		
27	ID5	SA6		
28	ID4	SA5		
29	ID3	SA4		
30	ID2	SA3		
31	ID1	SA2		
32	ID0	SA1		
33	RTR (x)	RTR[a] (d)		
34	FDF (x)	FDF[a] (d)		
35	r0 (d)	r0[a]		

Table 1 *(continued)*

Bit number	29 bit identifier		11 bit identifier	
	CAN	ISO 11783	CAN	ISO 11783[b]
36	DLC4	DLC4		
37	DLC3	DLC3		
38	DLC2	DLC2		
39	DLC1	DLC1		

SOF	Start of Frame bit	EDP	Extended Data Page according to ISO 11783
ID##	Identifier bit number (#)	SA#	Source Address bit number (#) according to ISO 11783
SRR	Substitute Remote Request	DP	Data Page according to ISO 11783
RTR	Remote Transmission Request bit	PF#	PDU Format bit number (#) according to ISO 11783
IDE	Identifier Extension bit	PS#	PDU Specific bit number (#) according to ISO 11783
FDF	FD Format Indicator	(d)	dominant bit
r#	CAN reserved bit number (#)	(r)	recessive bit
DLC#	Data Length Code bit number (#)	(x)	bit state dependent on message
P#	Priority bit number (#) according to ISO 11783		

[a] CAN-defined bit, unchanged in ISO 11783.

[b] Required format of proprietary 11 bit identifiers.

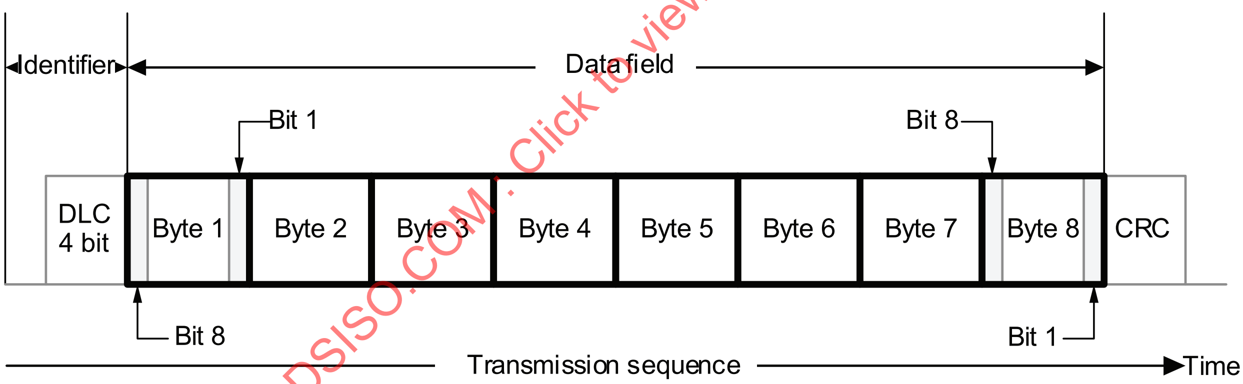


Figure 3 — Classical CAN data field

5.1.3 Parameter group numbers (PGN)

Whenever it is necessary to identify a parameter group in the data field of a classical CAN data frame, this is expressed in 24 bit. The 24 bit value is sent the least significant byte (LSB) first (see Table 2), also according to which the most significant byte (MSB) is sent third and the middle byte second and the LSB first. The 24 bit PGN is determined from the following constituent components: 6 bit set to zero, Extended Data Page bit, Data Page bit, PDU Format field (8 bit), and PDU Specific field (8 bit).

The procedure for the bit fields to be converted to PGN is as follows. The six MSB of the PGN are set to zero. Then the Extended Data Page bit, Data Page bit and PDU Format field are copied into the next 10 bit. If the PF value is less than 240 ($F0_{16}$) then the LSB of the PGN is set to zero. Otherwise, it is set to the value of the PS field. See Table 2 for an illustration of the PGN, their corresponding bits and their conversion to a decimal number.

NOTE Not all 131 072 combinations (2^{17}) are available to be assigned as PGN. Only a total of 8 672 combinations are available for assignment {calculated as: 2 pages × [240 + (16 × 256)] = 8 672}, using the conventions specified in this document. See ISO 11783-1 for the latest PGN assignments.

Table 2 — Parameter group number (PGN) examples

PGN constituent components					PGN		Numbers of assignable PGs	Cumulative numbers of PGs	ISO- or manufacturer-assigned
PGN (MSB) Byte 1 sent third in CAN data frame			PGN Byte 2 sent second in CAN data frame	PGN (LSB) Byte 3 sent first in CAN data frame	Dec_{10}	Hex_{16}			
Bit 8–3	EDP Bit 2	DP Bit 1	PF Bit 8–1	PS Bit 8–1					
0	0	0	0	0	0	000000_{16}			ISO
							239	239	
0	0	0	238	0	60 928	$00EE00_{16}$			
0	0	0	239	0	61 184	$00EF00_{16}$	1	240	MF
0	0	0	240	0	61 440	$00F000_{16}$			ISO
							3 840		
0	0	0	254	255	65 279	$00FEFF_{16}$		4 080	
0	0	0	255	0	65 280	$00FF00_{16}$			
							256		MF
0	0	0	255	255	65 535	$00FFFF_{16}$		4 336	
0	0	1	0	0	65 536	010000_{16}			
0	0	1	238	0	126 464	$01EE00_{16}$	239		ISO
0	0	1	239	0	126 720	$01EF00_{16}$	240	4576	MF
0	0	1	240	0	126 976	$01F000_{16}$			
							4 096		ISO
0	0	1	255	255	131 071	$01FFFF_{16}$		8 672	

5.1.4 ISO 11783 support of ISO 11898-1 CBFF messages

Controllers on the ISO 11783 network may support the CBFF (11 bit identifier) message format. Though these are not compatible with the ISO 11783 message structure, to accommodate the co-existence of the two formats, a minimum level of definition is given. This minimum definition allows controllers that use this format to not interfere with other controllers. CBFF messages are defined as being proprietary. In reference to Table 1, the 11 bit identifier field is parsed as follows: the three most significant bits are used as priority bits; the eight least significant bits identify the SA of the PDU. Priority bits are described in 5.2.2. The SA is described in 5.2.7.

Incorrect bus arbitration can occur when two messages, one base frame and one extended frame, access the bus at the same time. The source address (SA) is a higher relative priority in the base frame messages than in the extended frame messages. The message with an 11 bit identifier (base frame) can have an SA indicating a higher priority than that of the Extended Data Page bit, Data Page bit and PDU Format of the 29 bit identifier (extended frame) message. The three priority bits should be used to achieve the correct bus arbitration.

IMPORTANT — ISO 11783 defines a full strategy for standardized communications using the CEFF. Hardware that does not conform to ISO 11898-1 shall not be used on the network, since these versions of hardware do not allow the CEFF messages to be communicated.

5.2 Protocol data unit (PDU)

5.2.1 General

The applications and/or network layer provide a string of information that is assimilated into a protocol data unit. The protocol data unit provides a framework for organizing the information that is essential to each classical CAN data frame sent. The protocol data unit (PDU) of the ISO 11783 network shall consist of the seven fields listed in 5.1.2 and specified below. These fields shall then be packaged into one or more classical CAN data frames and sent over the physical media to other network controllers. There is only one PDU per classical CAN data frame.

NOTE Some PGN definitions require more than one classical CAN data frame for sending the corresponding data.

Certain bits of the classical CAN data frame fields are left out of the PDU definition because they are controlled entirely by the CAN specification and are invisible to all of the OSI layers above the data link layer. These include the SOF, SRR, IDE, RTR, FDF, CRC, ACK and EOF fields, and parts of the control field. They are defined by the CAN protocol definition and remain unmodified by ISO 11783.

The PDU fields (see Figure 4) are specified in 5.2.2 to 5.2.8.

	Priority,	EDP	DP	PF	PS	SA	Data
No. of bits	...3...,	...1...,	...1...,	...8...,	...8...,	...8...,	...64...

Figure 4 — PDU fields

5.2.2 Priority (P)

Priority bits are used to optimize message latency for transmission onto the bus only. They should be globally masked off by the receiving controller (ignored). The priority of any message can be set from highest, 0 (000_2), to lowest, 7 (111_2). The default for all control oriented messages is 3 (011_2). The default for all other informational, proprietary, request and NACK messages is 6 (110_2). This permits the priority to be raised or lowered in the future as new PGN values are assigned and bus traffic changes. A recommended priority is assigned to each PGN when it is added to the application layer standards. However, the priority field should be reprogrammable to allow for network tuning by the manufacturers if the need arises.

5.2.3 Extended data page (EDP)

The extended data page (EDP) bit is used in conjunction with the data page bit to determine the structure of the CAN identifier of the classical CAN data frame. All ISO 11783 messages shall set the extended data page bit to ZERO on transmit. (See Table 3 for the defined uses of the EDP and DP fields.) It is possible that future definitions will expand the PDU Format field, defining new PDU formats, expanding the priority field, or increasing the address space.

5.2.4 Data page (DP)

The data page (DP) bit is used in conjunction with the EDP bit to determine the structure of the CAN identifier of the classical CAN data frame. With the EDP set to 0, the DP bit selects between page 0 and page 1 of the PGN descriptions. See Table 3.

Table 3 — Definition of extended data page (EDP) and data page (DP) use

EDP Bit 25	DP Bit 24	Description
CAN ID Bit 25	**CAN ID Bit 24**	
0	0	ISO 11783 page 0 PGN
0	1	ISO 11783 page 1 PGN
1	0	ISO 11783 reserved
1	1	ISO 15765-2 defined PGN

NOTE The EDP and DP of the CAN 29 bit identifier being set to “11_2” identifies it as an ISO 15765-2 message. This means that the remaining bits of the CAN identifier is *not* set up as specified by ISO 11783; CAN frames following this format are not described in ISO 11783.

5.2.5 PDU format (PF)

PDU format (PF) is an 8 bit field that determines the PDU format and is one of the fields used to determine the PGN assigned to the classical CAN data field. PGN are used to identify or label commands, data, some requests, acknowledgements and negative acknowledgements, as well as for identifying or labelling information that requires one or more classical CAN data frames to communicate the information. If there is more information than can fit in eight data bytes, a multi-packet message is required to be sent. If there are eight or less data bytes, then a single classical CAN data frame is used. A PGN can represent one or more parameters, where a parameter is a piece of data such as engine rotations per minute. Even though a PGN label can be used for one parameter, it is recommended that multiple parameters be grouped so that all 8 byte of the data field are used.

The definition of two proprietary PGN allows both PDU1 and PDU2 formats to be used. The interpretation of the proprietary information varies between manufacturers.

EXAMPLE Even though two different engines can use a common set of standard PGNs, it is probable that one manufacturer’s proprietary communications will be different from another’s.

5.2.6 PDU Specific (PS)

The PDU specific (PS) field is an 8 bit field whose definition depends on its PDU format, which determines whether it will be a DA or GE field. See Table 4.

Table 4 — Definition of PDU Specific (PS) field

PDU format	PF	PS
PDU1	0–239	Destination Address (DA)
PDU2	240–255	Group Extension (GE)

The DA field defines the specific address to which the message is being sent. Any other controller should ignore this message. The global destination address (255) requires all controllers to listen and respond accordingly as message recipients.

The GE field, in conjunction with the four least significant bits of the PF field, provides for 4 096 parameter groups per data page. These are only available using the GE format PDU (PDU2).

NOTE When the four most significant bits of the PDU format field are set, it indicates that the PS field is a GE field.

In addition, 240 parameter groups are provided in each data page for use only in the destination-specific format PDU (PDU1 format). In total, 8 672 parameter groups are available to be defined using the two data pages currently available.

This total is calculated using Formula (1):

$$[240 + (16 \times 256)] \times 2 = 8\,672 \quad (1)$$

where

240 represents the number of PDU format field values available per data page (i.e. PDU1 format, PS field = DA);

16 is the number of PDU format values per GE value (i.e. PDU2 format only);

256 is the number of possible GE values (i.e. PDU2 format only);

2 is the number of data page states (both PDU formats).

See also 5.3.

5.2.7 Source address (SA)

The source address (SA) field is 8 bit long. There shall only be one control function on the network with a given source address.

NOTE For address management and allocation, and procedures to prevent duplication of SA, see ISO 11783-5.

5.2.8 Data field

5.2.8.1 Data from 0 to 8 byte

When eight or less bytes of data are required for expressing a given parameter group, then all eight data bytes of the classical CAN data frame can be used. It is recommended that 8 byte be allocated or reserved for all PGN assignments likely to expand in the future. This provides a means of adding parameters easily and avoiding incompatibility with previous revisions that only define part of the data field. Once the number of bytes of data associated with a PGN is specified, it cannot be changed (and cannot become multi-packet either, unless originally defined as such). The CAN data length code (DLC) is set to the defined parameter group "data length" value when it is 8 byte or less; otherwise, when the PG data length is 9 byte or greater, the CAN DLC is set to 8. For example, the REQUEST PGN, 59 904, has a PG data length of 3 byte, so the CAN DLC is set to 3. An individual group function (see 5.4.6) shall use the same data field length because the CAN identifier is always identical; while the classical CAN data field is used to convey the specific group subfunctions. These group functions require many different interpretations based on the classical CAN data field.

5.2.8.2 Data greater than 8 byte

When more than 8 data bytes are needed to express a given parameter group, the communication of these data are done in multiple classical CAN data frames. The term multi-packet is used to describe this type of parameter group. A parameter group defined as being multi-packet capable, having less than nine data bytes to transfer in a specific instance, shall be sent in a single classical CAN data frame with the DLC set to 8. When a particular parameter group has nine or more data bytes to transfer, then one of the transport protocol functions are used. The transport function connection management capability is used to set up and close out the communication of the multi-packet parameter groups. The transport protocol data transfer capability is used to communicate the data itself in a series of classical CAN data frames (packets) containing the "packetized" data. Additionally, the transport protocol function provides flow control and handshaking capabilities for destination-specific transfers (see 5.10).

All classical CAN data frames associated with a particular multi-packet response shall have a DLC of 8. All unused data bytes are set to "Not Available". The number of bytes per packet is fixed; however, ISO 11783 defines multi-packet messages that have a variable and or fixed number of packets. The

PGN for active diagnostic codes is an example of a multi-packet message that has a variable number of packets. Parameter groups that are defined as multi-packet only use the transport protocol when the number of data bytes to be sent exceeds eight in number.

5.3 Protocol data unit (PDU) formats

5.3.1 General

The available PDU formats, illustrated in Figure 5, are defined as PDU1 (PS = DA) and PDU2 (PS = GE). PDU1 allows for direction of the classical CAN data frame to a specific destination address (control function); PDU2 only communicates classical CAN data frames that are not destination-specific. Two separate PDU formats are created to provide more possible parameter group number combinations while still providing for destination-specific communications. Proprietary parameter group definitions are assigned so that both PDU formats can be used for proprietary communications. A standardized method for proprietary communications is defined to prevent possible conflicts in identifier usage.

The definition of proprietary Parameter Group Numbers has been established allowing both PDU1 and PDU2 Formats to be used. The interpretation of the proprietary information varies by manufacturer.

	Priority,	EDP,	DP,	PF,	PS(DA),	SA,	Data
No. of bits	...3...,	...1...,	...1...,	...8...,	...8...,	...8...,	...64...

a) PDU1

	Priority,	EDP,	DP,	PF,	PS(GE),	SA,	Data
No. of bits	...3...,	...1...,	...1...,	...8...,	...8...,	...8...,	...64...,

b) PDU2

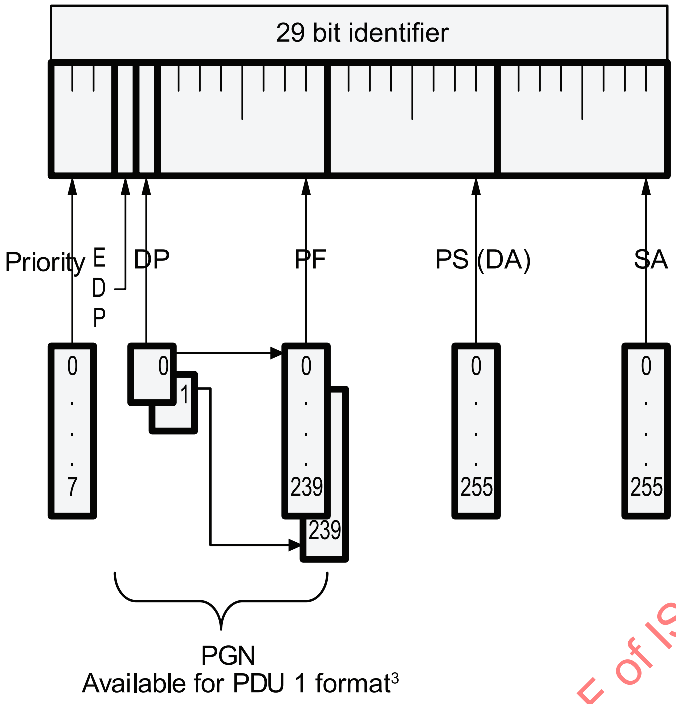
Figure 5 — Available PDU formats

5.3.2 PDU1 format

The PDU1 format provides for applicable parameter groups to be sent to either specific or global destination(s). The PS field contains a DA.

PDU1 format messages can be requested or sent as unsolicited messages.

PDU1 format messages are determined by the PF field. When the value of that field is 0 to 239, the message is in the PDU1 format. The format of the PDU1 message is illustrated by Figure 5. See also Figure 6.



a Currently, 2 × 240 = 480.

Figure 6 — PDU1 format

Parameter groups requiring a destination (PDU1) and minimal latency start at PF = 0 and increment towards x (or x1).

Parameter groups requiring a destination where latency is not critical start at PF = 239 and decrement towards x (or x1).

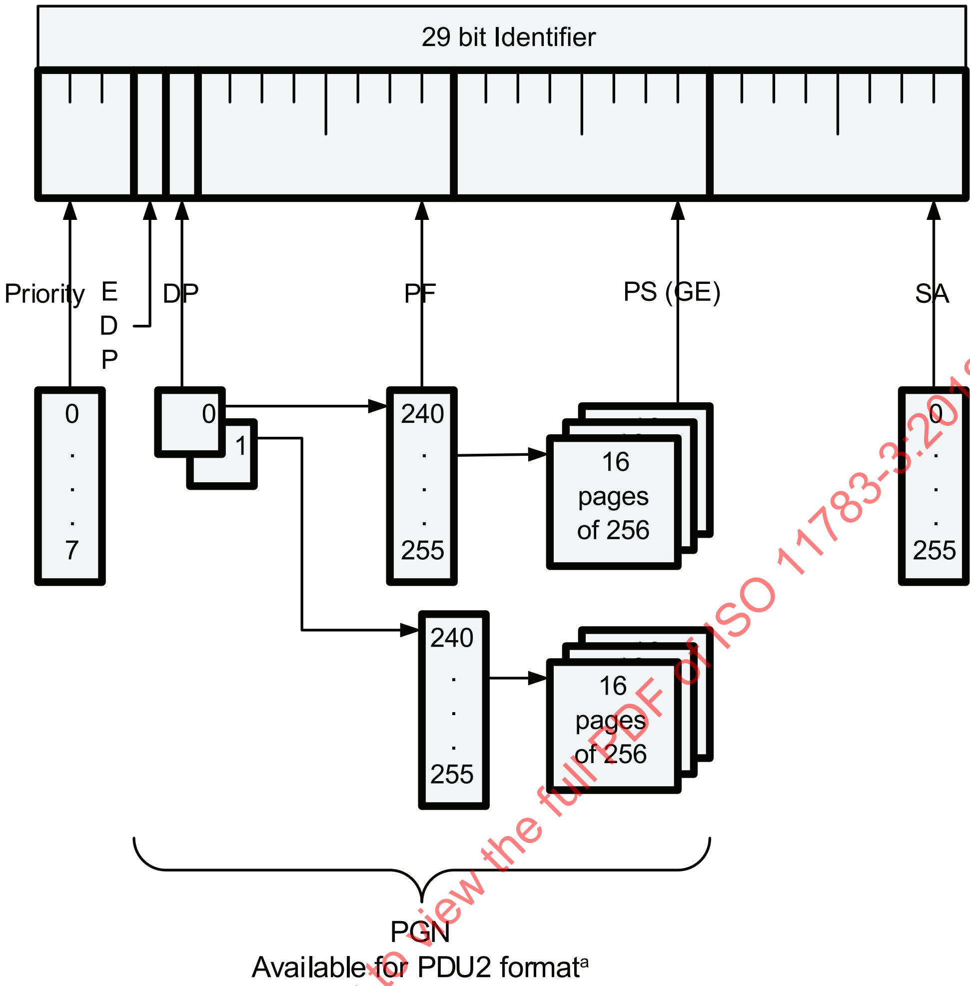
See Table 7.

A PF equal to 239 (Extended Data Page bit = 0 and Data Page bit = 0) is assigned for proprietary use. In this case the PS field is a destination address (see 5.4.6). The PGN for Proprietary A is 61 184.

5.3.3 PDU2 format

The PDU2 format can only be used to communicate parameter groups as global messages. PDU2 format messages can be requested or sent as unsolicited messages. Selection of the PDU2 format at the time a PGN is assigned prevents that PGN from ever being able to be directed to a specific destination. The PS field contains a GE.

PDU2 format messages are defined as being those where the PF value is equal to 240 to 255. The format of the PDU2 message is illustrated by Figure 5. Also see Figure 7.



a Currently, 2 × 16 × 256 = 8 192.

Figure 7 — PDU2 format

The PGN of messages that are sent at fast update rates (generally less than 100 ms) start at PF = 240 and increment towards y (or y1).

The PGN of messages that are only requested, sent on change, or are sent at slow update rates (generally greater than 100 ms) start at PF = 254 and decrement towards y (or y1).

See Table 7.

A PF equal to 255 (Extended Data Page bit = 0 and Data Page bit = 0) is assigned for proprietary use. The PS field is left to be defined and used by each manufacturer (see 5.4.6). The PGN for Proprietary B covers the range 65 280 to 65 535.

5.4 Message types

5.4.1 General

There are five message types currently supported:

— Commands;

— Requests;

— Broadcasts/Responses;

— Acknowledgements;

— Group Functions.

The specific message type is recognized by its assigned PGN. The RTR bit (defined in the CAN protocol for remote frames) is not to be used in the recessive state (logical 1). Therefore, Remote Transmission Request (RTR = 1) is not available for use in the ISO 11783 network.

RTR was meant to "request" a specific CAN object by simply announcing its CAN identifier on the network without any data bytes. Since ISO 11783 uses part of the CAN identifier for message priority and part of it for source address, this mechanism will cause conflicts. A device is not allowed to send a message with another device's SA. A separate request mechanism exists in ISO 11783. See 5.4.3.

Multi-byte parameters that appear in the data field of a classical CAN data frame shall be placed LSB first. Exceptions are noted where applicable (i.e. ASCII data). If a 2 byte parameter were to be placed in Bytes 7 and 8 of the classical CAN data frame, the LSB would be placed in Byte 7 and the MSB in Byte 8.

5.4.2 Command

The command message type categorizes those parameter groups that command a specific or global destination from a source. The destination is then supposed to take specific actions based on the reception of this command message type. Both PDU1 (PS = DA) and PDU2 format (PS = GE) messages can be used for commands. Example command modes include Transmission Control, Address Request and Torque/Speed Control.

5.4.3 Request

The Request message type, identified by the PGN, provides the ability to request information globally or from a specific destination. Requests specific to one destination are known as *destination-specific* requests. The information below assigns a PGN to the Request PGN parameter group. The information is in the same format as parameter group definitions in ISO 11783.

Parameter group name:	REQUEST
Definition:	used to request a parameter group from a network control function or control functions
Transmission repetition rate:	per user requirements, generally recommended that requests occur no more than two or three times per second
Data length:	3 byte (The classical CAN frame for this PG shall set the DLC to 3)
Data page:	0
PF:	234
PS:	DA (global or specific)
Default priority:	6
Parameter group number:	59 904 ($00EA00_{16}$)
Bytes:1, 2, 3	PGN being requested (see 5.1.3 for field definition and byte order)

Table 5 lists the request/response possibilities for PDU1 and PDU2 format PGN. It clarifies that the originating control function of a message determines whether the destination is specific or global, based on whether the request was to a specific or global DA. Table 5 also illustrates that, for unsolicited messages, the originating control function can transmit to a specific or global DA for PDU1 and PDU2

PGN with more than 8 byte. For PDU2 PGN with 8 byte or less, the originating control function can only send the data globally.

Table 5 — PDU1 and PDU2 transmit, request and response requirements

PDU Format	Data length byte	Request PGN 59 904	Response	TP used
1	≤ 8	DA specific	DA specific	Not allowed
1	≤ 8	DA global	DA global	Not allowed
1	≤ 8	None	DA global	Not allowed
			DA specific	Not allowed
1	> 8	DA specific	DA specific	RTS/CTS
1	> 8	DA global	DA global	BAM
1	> 8	None	DA global	BAM
			DA specific	RTS/CTS
2	≤ 8	DA specific	DA global	Not allowed
2	≤ 8	DA global	DA global	Not allowed
2	≤ 8	None	DA global	Not allowed
2	> 8	DA specific	DA specific	RTS/CTS
2	> 8	DA global	DA global	BAM
2	> 8	None	DA global	BAM
			DA specific	RTS/CTS

The following are general rules of operation for determining whether to send a PGN to a global or specific destination. See Table 5.

a) If the request is sent to a global address, then the response is sent to a global address.

A NACK (see 5.4.5) is not permitted as a response to a global request.

b) If the request is sent to a specific address, then the response is sent to a specific address.

A NACK is required if the PGN is not supported.

If the data length is 8 byte or more, the transport protocol RTS/CTS shall be used for the response to a specific address.

Exceptions:

— PDU2 format PGN with 8 byte or less can only be sent to a global destination because there is no destination address field in the PDU2 format.

— The Address Claim PGN is sent to the global destination address even though the request for it was to a specific destination address (see ISO 11783-5).

c) For periodic broadcasts or unsolicited messages, PDU1 or PDU2 format PGN can be sent to a global or specific destination address.

Exception:

— PDU2 format PGN with 8 byte or less can only be sent to a global destination because there is no destination address field in the PDU2 format.

d) Exceptions to these rules do exist, as can be seen from the above. The exceptions are noted in the applicable document in the section in which the PGN is defined.

Table 6 gives two examples of how the Request PGN is used.

Table 6 — Use of specified fields in ISO 11783 PDU1 format

Message type	PGN	PS (DA)	SA	Data 1	Data 2	Data 3
Global request	59 904	255 (Responder)	SA1 (Requester)	PGN LSB[a]	PGN	PGN MSB[a]
Specific request	59 904	SA2 (Responder)	SA1 (Requester)	PGN LSB[a]	PGN	PGN MSB[a]

[a] The parameter group number (PGN) in the data field is used to identify the information being requested.

A response is always required from a specified destination (not global), even if it is a NACK indicating that the particular PGN value is not supported. A global request shall not be responded to with a NACK when a particular PGN is not supported by a control function.

NOTE 1 Some PGNs are multi-packet, so several classical CAN data frames can occur for a single request.

The Request PGN can be directed to a specific destination address to determine if a specific parameter group is supported (i.e. "does the requested destination address transmit the specific group?"). The response to the request determines whether the PGN is supported. If it is supported, then the receiving control function shall send the requested information. If the Acknowledgement PGN is appropriate, then the control byte shall be set to 0 or 2 or 3. If it is not supported, the receiving control function shall send the Acknowledgement PGN with the control byte set to 1, for Negative Acknowledgement. The remaining portions of the ISO 11783 PDU format and parameter group shall be filled in appropriately (see 5.4.5). It is not possible to determine whether a control function acts upon the PG (when received) by using this method.

NOTE 2 "Not supported" above means that the PG is not transmitted.

If a control function fails to receive a response (either the PG or ACKNOWLEDGEMENT) to a Request within the Response timeout, then the control function can resend or retry the same Request. The number of retries for a specific Request should be limited to two (2) retries, i.e. the Request is issued a total of three (3) times. If the control function fails to receive a response (either the PG or ACKNOWLEDGEMENT) to the Request after the second retry, then the control function should abandon further request attempts for the same information or the control function can wait for an extended period of time (minutes rather than seconds) before attempting to request the same information.

5.4.4 Broadcast/Response

The Broadcast/Response message type can be either an unsolicited broadcast of information from a control function or the response to a command or request.

5.4.5 Acknowledgement

There are two forms of acknowledgement available. The first form is provided for by the CAN protocol. It consists of an "in-frame" acknowledgement confirming that a message is received by at least one controller. In addition, messages are further acknowledged by the absence of CAN error frames. Their absence acknowledges that all other powered and connected controllers received the message correctly.

The second form of acknowledgement is a response of a "normal broadcast" or "ACK" or "NACK" to a specific command or request as provided for by an application layer. The definition of the Acknowledgement parameter group is shown below. The type of acknowledgement required for some parameter groups is defined in the applications layer.

For Group Function parameter groups (see 5.4.6), the Group Function Value parameter allows a control function to identify the specific group function that is being acknowledged. The Group Function Value is unique to each Group Function PG. It is desirable that the Group Function Values only use the range 0 to 250.

Each form of Acknowledgement includes an address acknowledge byte that contains the source address of the originator of the request that the Acknowledgement is directed towards. For maximum compatibility with SAE J1939, the address acknowledge field is a duplicate of the PS field. The parameters, in byte 5, are Address Acknowledged, Address Negative Acknowledgement, Address Access Denied and Address Busy.

Parameter group name:	ACKNOWLEDGEMENT
Definition:	parameter group used to provide a handshake mechanism between transmitting and receiving control functions
Transmission repetition rate:	upon reception of a PGN that requires this form of acknowledgement
Data length:	8 byte
Data page:	0
PF:	232
PS:	DA
Default priority:	6
Parameter group number:	59 392 ($00E800_{16}$)

Data ranges for parameters used by this message type:

Control byte:	0 to 3	See definitions below
	4 to 127	Reserved for assignment in a future International Standard.
	128 to 131	See definitions below
	132 to 143	Reserved for assignment in a future International Standard.
	144 to 147	See definitions below
	148 to 159	Reserved for assignment in a future International Standard.
	160 to 163	See definitions below
	164 to 255	Reserved for assignment in a future International Standard.
Group Function Value	0 to 250	Definition is specific to the individual PGN, when applicable. Most often it is located as the first byte in the data field of the applicable Group Function parameter group.
	251 to 255	Follows the conventions defined in ISO 11783-7.

Positive Acknowledgement: control byte = 0

Byte:	1	Control byte = 0, Positive Acknowledgement (ACK)
Byte:	2	Group Function Value (if applicable) (see 5.4.6) otherwise FF_{16}
Byte:	3 to 4	Reserved for assignment in a future International Standard; send each of these bytes as FF_{16}
Byte:	5	Address Acknowledged

Byte:	6	PGN of requested information (8 LSB of parameter group number, Bit 8 MSB)
Byte:	7	PGN of requested information (second byte of parameter group number, Bit 8 MSB)
Byte:	8	PGN of requested information (8 MSB of parameter group number, Bit 8 MSB)

Negative Acknowledgement: control byte = 1

Byte:	1	Control byte = 1, Negative Acknowledgement (NACK)
Byte:	2	Group Function Value (if applicable) (see 5.4.6) otherwise FF_{16}
Byte:	3 to 4	Reserved for assignment in a future International Standard, send each of these bytes as FF_{16}
Byte:	5	Address Negative Acknowledgement
Byte:	6 to 8	PGN of requested information (see above)

Access Denied: control byte = 2

Byte:	1	Control byte = 2, Access Denied (PGN supported but security denied access)
Byte:	2	Group Function Value (if applicable) (see 5.4.6) otherwise FF_{16}
Byte:	3 to 4	Reserved for assignment in a future International Standard, send each of these bytes as FF_{16}
Byte:	5	Address Access Denied
Byte:	6 to 8	PGN of requested information (see above)

Cannot Respond: control byte = 3

Byte:	1	Control byte = 3; Cannot Respond (PGN supported but ECU is busy and cannot respond now, re-request the data at a later time)
Byte:	2	Group Function Value (if applicable) (see 5.4.6) otherwise FF_{16}
Byte:	3 to 4	Reserved for assignment in a future International Standard, send each of these bytes as FF_{16}
Byte:	5	Address Busy
Byte:	6 to 8	PGN of requested information (see above)

Special ACKNOWLEDGEMENT cases only for when the Request2 utilizes the Extended Identifier Type (see 5.4.7).

Positive Acknowledgement: control byte = 128 and Extended Identifier Type “One Byte Extended Identifier”

Byte:	1	Control byte = 128, Positive Acknowledgement (ACK)
Byte:	2	Group Function Value = Extended Identifier
Byte:	3 to 4	Reserved for assignment in a future International Standard, send each of these bytes as FF_{16}
Byte:	5	Address Acknowledged
Byte:	6 to 8	PGN of requested information (see above)

Negative Acknowledgement: control byte = 129 and Extended Identifier Type “One Byte Extended Identifier”

Byte:	1	Control byte = 129, Negative Acknowledgement (NACK)
Byte:	2	Group Function Value = Extended Identifier
Byte:	3 to 4	Reserved for assignment in a future International Standard, send each of these bytes as FF_{16}
Byte:	5	Address Negative Acknowledged
Byte:	6 to 8	PGN of requested information (see above)

Access Denied: control byte = 130 and Extended Identifier Type “One Byte Extended Identifier”

Byte:	1	Control byte = 130, Access Denied (PGN supported but security denied access)
Byte:	2	Group Function Value = Extended Identifier
Byte:	3 to 4	Reserved for assignment in a future International Standard, send each of these bytes as FF_{16}
Byte:	5	Address Access Denied
Byte:	6 to 8	PGN of requested information (see above)

Cannot Respond: control byte = 131 and Extended Identifier Type “One Byte Extended Identifier”

Byte:	1	Control byte = 131, Cannot Respond (PGN supported but CA is busy and cannot respond now. Re-request the data at a later time)
Byte:	2	Group Function Value = Extended Identifier
Byte:	3 to 4	Reserved for assignment in a future International Standard, send each of these bytes as FF_{16}
Byte:	5	Address Busy
Byte:	6 to 8	PGN of requested information (see above)

Positive Acknowledgement: control byte = 144 and Extended Identifier Type “Two Byte Extended Identifier”

Byte:	1	Control byte = 144, Positive Acknowledgement (ACK)
Byte:	2	Group Function Value = Extended Identifier (8 LSB of Extended Identifier, Bit 8 MSB)
Byte:	3	Group Function Value = Extended Identifier (8 MSB of Extended Identifier, Bit 8 MSB)
Byte:	4	Reserved for assignment in a future International Standard, send this byte as FF_{16}
Byte:	5	Address Acknowledged
Byte:	6 to 8	PGN of requested information (see above)

Negative Acknowledgement: control byte = 145 and Extended Identifier Type “Two Byte Extended Identifier”

Byte:	1	Control byte = 145, Negative Acknowledgement (NACK)
Byte:	2	Group Function Value = Extended Identifier (8 LSB of Extended Identifier, Bit 8 MSB)
Byte:	3	Group Function Value = Extended Identifier (8 MSB of Extended Identifier, Bit 8 MSB)
Byte:	4	Reserved for assignment in a future International Standard, send this byte as FF_{16}
Byte:	5	Address Negative Acknowledged
Byte:	6 to 8	PGN of requested information (see above)

Access Denied: control byte = 146 and Extended Identifier Type “Two Byte Extended Identifier”

Byte:	1	Control byte = 146, Access Denied (PGN supported but security denied access)
Byte:	2	Group Function Value = Extended Identifier (8 LSB of Extended Identifier, Bit 8 MSB)
Byte:	3	Group Function Value = Extended Identifier (8 MSB of Extended Identifier, Bit 8 MSB)
Byte:	4	Reserved for assignment in a future International Standard, send this byte as FF_{16}
Byte:	5	Address Access Denied
Byte:	6 to 8	PGN of requested information (see above)

Cannot Respond: control byte = 147 and Extended Identifier Type “Two Byte Extended Identifier”

Byte:	1	Control byte = 147, Cannot Respond (PGN supported but CA is busy and cannot respond now. Re-request the data at a later time)
Byte:	2	Group Function Value = Extended Identifier (8 LSB of Extended Identifier, Bit 8 MSB)
Byte:	3	Group Function Value = Extended Identifier (8 MSB of Extended Identifier, Bit 8 MSB)
Byte:	4	Reserved for assignment in a future International Standard, send this byte as FF_{16}
Byte:	5	Address Busy
Byte:	6 to 8	PGN of requested information (see above)

Positive Acknowledgement: control byte = 160 and Extended Identifier Type "Three Byte Extended Identifier"

Byte:	1	Control byte = 160, Positive Acknowledgement (ACK)
Byte:	2	Group Function Value = Extended Identifier (8 LSB of Extended Identifier, Bit 8 MSB)
Byte:	3	Group Function Value = Extended Identifier (second byte of Extended Identifier, Bit 8 MSB)
Byte:	4	Group Function Value = Extended Identifier (8 MSB of Extended Identifier, Bit 8 MSB)
Byte:	5	Address Acknowledged
Byte:	6 to 8	PGN of requested information (see above)

Negative Acknowledgement: control byte = 161 and Extended Identifier Type "Three Byte Extended Identifier"

Byte:	1	Control byte = 161, Negative Acknowledgement (NACK)
Byte:	2	Group Function Value = Extended Identifier (8 LSB of Extended Identifier, Bit 8 MSB)
Byte:	3	Group Function Value = Extended Identifier (second byte of Extended Identifier, Bit 8 MSB)
Byte:	4	Group Function Value = Extended Identifier (8 MSB of Extended Identifier, Bit 8 MSB)
Byte:	5	Address Negative Acknowledged
Byte:	6 to 8	PGN of requested information (see above)

Access Denied: control byte = 162 and Extended Identifier Type "Three Byte Extended Identifier"

Byte:	1	Control byte = 162, Access Denied (PGN supported but security denied access)
Byte:	2	Group Function Value = Extended Identifier (8 LSB of Extended Identifier, Bit 8 MSB)
Byte:	3	Group Function Value = Extended Identifier (second byte of Extended Identifier, Bit 8 MSB)
Byte:	4	Group Function Value = Extended Identifier (8 MSB of Extended Identifier, Bit 8 MSB)
Byte:	5	Address Access Denied
Byte:	6 to 8	PGN of requested information (see above)

Cannot Respond: control byte = 163 and Extended Identifier Type "Three Byte Extended Identifier"

Byte:	1	Control byte = 163, Cannot Respond (PGN supported but CA is busy and cannot respond now. Re-request the data at a later time)
Byte:	2	Group Function Value = Extended Identifier (8 LSB of Extended Identifier, Bit 8 MSB)
Byte:	3	Group Function Value = Extended Identifier (second byte of Extended Identifier, Bit 8 MSB)
Byte:	4	Group Function Value = Extended Identifier (8 MSB of Extended Identifier, Bit 8 MSB)
Byte:	5	Address Busy
Byte:	6 to 8	PGN of requested information (see above)

5.4.6 Group function

5.4.6.1 General

The Group Function message type is used for groups of functions. Proprietary functions, multi-packet transport functions and network management functions (ISO 11783-5) Virtual terminals (ISO 11783-6), task controllers (ISO 11783-10), file servers (ISO 11783-13) and sequence controllers (ISO 11783-14) use Group Function messages.

Each group of functions is recognized by its assigned PGN. The function itself is defined within the data structure (typically the first byte of the data field). More detailed explanation of the group function's proprietary and transport protocol is given in 5.4.6.2 to 5.4.6.4. The proprietary group function provides a means of transmitting proprietary messages in a way that eliminates CAN identifier usage conflicts between different manufacturers. It also provides a means for receiving and distinguishing proprietary messages for use when desired. Group functions can provide their own request, ACK and or NACK mechanisms if the messages defined in this document are not sufficient.

A request using PGN 59 904 (see 5.4.3) can be used to determine if a specific parameter group of the Group Function message type is supported. If supported, then the receiving control function sends the Acknowledgement PGN with the control byte equal to zero for Positive Acknowledgement, or equal to two for Access Denied or equal to 3 for Cannot Respond. If not supported, the receiving control function sends the Acknowledgement PGN with the control byte set to one, for Negative Acknowledgement. The remaining portions of the ISO 11783-specified PF and parameter group shall be filled in appropriately (see 5.4.5).

Per the definitions in this paragraph, the phrase “not supported” means that the PG is not transmitted. It is not possible to determine whether a control function acts upon the received PG by using this method.

The data length of Proprietary A, A2 and B messages may be set by each manufacturer. Therefore, two manufacturers may use the same GE value and it can have a different data length code. Receiving control functions of this information need to differentiate between the two manufacturers. How the data field of this message is used is at the option of the manufacturer, as is the use of proprietary messages. Use of proprietary messages during normal operation should be minimized; where the sum of Proprietary A, A2, and B per CF should not exceed 2 % of the network capacity (see Annex D).

5.4.6.2 Proprietary A

Parameter group name:		PROPRIETARY A
Definition:		Proprietary PG that uses destination-specific PDU Format to allow manufacturers to direct their proprietary communications to a specific destination control function.
Transmission repetition rate:		per user requirements
Data length:		0 to 1 785 byte (multi-packet supported)
Data page:		0
PF:		239
PS:		DA
Default priority:		6
Parameter group number:		61 184 ($00EF00_{16}$)
Bytes:	1 to 8	manufacturer-specific use (see 5.1.3)

Data ranges for parameters used by this group function: none defined by ISO.

5.4.6.3 Proprietary A2

Parameter group name:		PROPRIETARY A2
Definition:		Proprietary PG that uses destination-specific PDU Format to allow manufacturers to direct their proprietary communications to a specific destination control function.
Transmission repetition rate:		per user requirements
Data length:		0 to 1 785 byte (multi-packet supported)
Data page:		1
PF:		239
PS:		DA
Default priority:		6
Parameter group number:		126 720 ($01EF00_{16}$)
Bytes:	1 to 8	manufacturer-specific use (see 5.1.3)

Data ranges for parameters used by this group function: none defined by ISO.

5.4.6.4 Proprietary B

Parameter group name:	PROPRIETARY B
Definition:	Proprietary PG that uses the PDU2 Format message to allow manufacturers to define the PS (GE) field content as they desire.
Transmission repetition rate:	per user requirements
Data length:	0 to 1 785 byte (multi-packet supported)
Data page:	0
PF:	255
PS:	GE (manufacturer-assigned)
Default priority:	6
Parameter group number:	65 280 to 65 535 ($00FF00_{16}$ to $00FFFF_{16}$)
Bytes: 1 to 8	manufacturer-defined usage (see 5.1.3)

Data ranges for parameters used by this group function:

Manufacturer-defined usage results with the data length code being unique per component supplier and source address. Caution should be taken when using the Proprietary B parameter group (PGN = 65 280) because multiple source addresses can use the same Proprietary B PGN value for different purposes.

5.4.7 Request2

The Request2 PG has the added capability of specifying whether the receiving control function uses the Transfer PGN 51 712. By specifying that the receiving control function use the Transfer PGN, it provides the ability for the receiving control function to report the data set for all control functions it is tasked with reporting (see 5.4.8), including that which the receiving control function would normally report upon receiving the same PGN requested in PGN 59 904 (properly formatted for TRANSFER PGN) and the data set for each control function it is tasked to report. For instance, if the Use Transfer PGN parameter is yes (01_2), the response shall include all known data relative to the requested PGN. When Use Transfer PGN is 00_2, the effect of the Request2 PGN is the same as if the Request PGN (59 904) was used. The response to the Request2 when the Use Transfer PGN equals 00_2 is not sent using the Transfer PGN, and it is sent exactly the same as it would be had the request been made using the Request PGN (i.e. PGN 59 904). The information below assigns the PGN to the Request2 parameter group.

The Request2 and Transfer PGN are required in cases where a given control function is tasked with reporting a PGN and data about more than one control function.

EXAMPLE ECU identification information, Component ID and Software Identification PGN. See also example of 5.4.8. If a control function fails to receive a response (either the PGN or ACKNOWLEDGEMENT) to a Request within the Response timeout, then the control function can resend or retry the same Request. The number of retries for a specific Request should be limited to two (2) retries, i.e. the Request is issued a total of three (3) times. If the control function fails to receive a response (either the PGN or ACKNOWLEDGEMENT) to the Request after the second retry, then the control function should abandon further request attempts for the same information or the control function can wait for an extended period of time (minutes rather than seconds) before attempting to request the same information.

The support of REQUEST2 is optional.

Parameter group name:	REQUEST2
Definition:	Used to request a PGN from network control function or control functions and to specify whether the response uses or does not use the Transfer PGN.
Transmission repetition rate:	Per user requirements, generally recommended that requests occur no more than 2 or 3 times per second. When a control function supports Request2, a NACK is required if the PG being asked for with a destination-specific address is not supported, see PGN 59 392.
Data length:	8 byte (multi-packet supported)
Data page:	0
PF:	201
PS:	destination-specific (global or specific)
Default priority:	6
Parameter group number:	51 456 ($00C900_{16}$)
Bytes 1 to 3:	Requested PGN
Byte 4:	Special instructions
Bits 6 to 8:	Reserved for assignment in a future International Standard
Bits 3 to 5:	Control Indicating Extended Identifier Type (Extended Identifier conveyed in data bytes 5 through 7)
Bits 1 to 2:	Use Transfer PGN for response (00_2 = No, 01_2 = Yes, 10_2 = Undefined, 11_2 = Not Allowed)
Byte 5:	Extended Identifier Byte 1 (least significant byte)
Byte 6:	Extended Identifier Byte 2
Byte 7:	Extended Identifier Byte 3 (most significant byte)
Byte 8:	Reserved for assignment in a future International Standard

Data ranges for parameters used by this message type:

Control Indicating Extended Identifier Type:

000_2 - No Extended Identifier. None of the data bytes are used for identifier/control values. Used when requesting a PGN that does not have Group Function/Extended Identifier bytes. Indicator that the device supports the new Request2 special instruction functionality. Data bytes 5 to 7 of the Request2 PGN are set to 255 (FF_{16}).

001_2 - One Byte Extended Identifier. Data byte 5 of the Request2 PGN contains the one byte identifier/control value that would match the requested PGN's data byte 1. Data bytes 6 to 7 of the Request2 PGN are set to 255 (FF_{16}).

010_2 - Two Byte Extended Identifier. Data byte 5 of the Request2 PGN contains the byte identifier/control value that would match the requested PGN's data byte 1 and data byte 6 of the Request2 PGN contains the identifier/control value that would match the requested PGN's data byte 2. Data byte 7 of the Request2 PGN is set to 255 (FF_{16}).

011_2 - Three Byte Extended Identifier. Data byte 5 of the Request2 PGN contains the byte identifier/control value that would match the requested PGN's data byte 1, data byte 6 of the Request2 PGN contains the identifier/control value that would match the requested PGN's data byte 2, and data byte 7 of the Request2 PGN contains the identifier/control value that would match the requested PGN's data byte 3.

100_2 to 110_2 - Reserved for assignment in a future International Standard

111_2 - Take no action. Not Applicable.

5.4.8 Transfer

The Transfer PGN provides a mechanism for reporting multiple data sets for a given PGN in response to Request2 (see 5.4.7) These multiple sets of data for a given PGN require that each data set have a length and be labelled with four bytes from the ISO 11783-5 NAME. The four bytes of the NAME identify each control function. The control function responding to the request shall report the same information it would with PGN 59 904 as the first data set in this response. If a control function only has one data set then it shall respond with the one data set utilizing the Transfer PG.

The Request2 and Transfer PGN are useful in cases where a given control function is tasked with reporting a PGN and data about more than one control function. Examples include PGN such as Implement Identification, Component ID and Software Identification. The information below assigns the PGN to the Transfer parameter group.

Parameter group name:		TRANSFER
Definition:		Used for transfer of data in response to a Request2 when “Use Transfer PGN for Response” is set to Yes.
Transmission repetition rate:		in response to a Request2 PGN with “Use Transfer PGN” = 01
Data length:		9 to 1 785 byte (multi-packet supported)
Data page:		0
PF:		202
PS:		DA (specific or global)
Default priority:		6
Parameter group number:		51 712 ($00CA00_{16}$)
Bytes 1 to 3:		a) PGN requested by Request2 (see Table 2 for PGN ordering)
Byte 4:		b) Length of data for the reported PGN associated to the control function identified (e.g. Control function in bytes 5 to 8). Length value is the total of this byte, length of identity bytes (i.e. bytes 5 to 8), and the associated PGN data. So the length is b + c + d.
Bytes 5 to 8:		c) Identity of control function associated to the PGN and data (i.e. Control function identity)
Byte 5:	Bits 4 to 8	Function Instance (most significant at Bit 8)
	Bits 1 to 3	ECU Instance (most significant at Bit 3)
Byte 6:	Bits 1 to 8	Function (most significant at Bit 8)
Byte 7:	Bits 2 to 8	Device Class (most significant at Bit 8)

	Bit 1	Reserved
Byte 8:	Bit 8	Self Configurable Capable
	Bits 5 to 7	Industry Group (most significant at Bit 7)
	Bits 1 to 4	Device Class Instance (MSB Bit 4)
Bytes 9 to x:		d) Repeating information for 2nd and following shall contain: "Control function Identity," Length, and "PGN requested by Request2's data" (See format definitions below.)

Format a, b, c, d, b, c, d, b, c, d ...:

a PGN requested by Request2 when transfer mode is set to Yes;

b first data set — length of concatenated control function identity and associated PGN data (length = b + c + d);

c identity of control function to which field d is associated;

d requested PGN data for specific control function;

b second data set — length of concatenated control function identity and associated PGN data;

c identity of control function to which field d is associated;

d requested PGN data for second specific control function.

EXAMPLE For a given vehicle, the engine control function knows the VIN numbers for the tractor and the trailer. Another control function sends the Request2 directed to the global destination, requesting the VIN with "Use Transfer PGN" set to 01_2. The response from the Engine might be

— BAM transfer of the Transfer PGN reporting the VIN for the tractor and VIN for the trailer, or

— if the request had the "Use Transfer PGN" set to 00_2, a response of BAM transfer of the VIN for the tractor but not utilizing the Transfer PGN.

5.5 Message priority

The CAN data frame priority shall be in accordance with ISO 11898-1. The value within the CAN identifier field determines the message priority. A low value (the 29 bit identifier set to all zeros) is the highest priority, while the largest CAN identifier is the lowest priority (the 29 bit identifier set to all ones). The assignments are identified in the application layer following the guidelines given in 5.9.

5.6 Bus access

When the bus is free, any controller can start to transmit a classical CAN data frame. If two or more controllers start to transmit frames at the same time, the bus access conflict is resolved by contention-based arbitration using the CAN data frame identifier. The mechanism of arbitration ensures that neither information nor time is lost. The originating controller with the frame of the highest priority gains bus access.

5.7 Contention-based arbitration

During arbitration, every originating controller compares the level of the bit transmitted with the level that is monitored on the bus. If these levels are equal, the controller can continue to send. When a recessive level is sent and a dominant level is monitored, that controller loses arbitration and shall withdraw without sending another bit. When a dominant level is sent and a recessive level monitored, that controller detects a bit error.

5.8 Error detection

The following measures are provided for detecting errors:

— originating controllers compare the bit levels to be transmitted with the bit level detected on the bus;

— bit cyclic redundancy check (CRC);

— variable bit stuffing with a stuff width of 5;

— frame format check.

NOTE For a more detailed explanation of these error detection techniques, see ISO 11898-1.

5.9 Assignment process for SA and PGN

5.9.1 General

The protocol data units that are available provide for the two formats, PDU1 and PDU2. Parameter groups are assigned specifically to use either the PDU1 or PDU2 format. Once a format is assigned, the other format is not available for that parameter group. The PDU1 format shall be used whenever it is necessary to direct a parameter group to a specific destination. The assignment of a parameter group shall be done using the following characteristics:

— priority;

— update rate;

— importance of the data in the packet to other network control functions;

— length of the data associated with the parameter group.

In order to help with this assignment process a request form is available for use when requesting new SA or PGN assignments.

Table 7 provides a template for assigning PGN. Note that the priority column is used to assign a default priority value for each PGN. The priority field may be programmable for each PGN value so that network tuning can be performed by an OEM (original equipment manufacturer), if necessary. Although any PGN can be requested, requests are strongly discouraged for messages that are already periodically broadcasted.

Messages shall be assigned a PGN that requires a destination only if it is a parameter intended to directly control (command) one of several specific control functions. Otherwise, a PGN shall be selected without a destination so that any control function can access the parameters within the message.

Preferred SA are assigned in a linear fashion without concern for message priority, update rate, or importance.

PGN are assigned linearly to the various sections in Table 7 based on the criteria provided in the PGN and SA Request form. Note that multi-packet messages are not permitted when the repetition rate is greater than or equal to 10 Hz.

5.9.2 Address assignment criteria

The number of unassigned addresses in ISO 11783 is limited and new address assignments shall be made efficiently. The maximum number of addresses assigned within the system shall not exceed 256. Additions to the address definitions shall therefore be limited to units that provide specific functions within the tractor or implement. Examples of specific functions include the currently defined addresses for engine, transmission, brakes and fuel system. Functions proposed for new address assignments within this document should have a scope similar to currently defined addresses and be useful to most ISO 11783 users.

ISO 11783 controllers shall support address self-configuring in accordance with ISO 11783-5.

5.9.3 Parameter group assignment criteria

The number of unassigned parameter groups available in ISO 11783 is limited when compared to the large number that might be proposed for forestry or agricultural applications. The need for large numbers of parameter groups is alleviated by features incorporated into ISO 11783. Three primary communication methods exist within ISO 11783, the appropriate use of each of which allows effective use of the available parameter groups:

— PDU1 format (PS = DA, allowing destination-specific communications);

— PDU2 format (PS = GE);

— proprietary communications using two predefined proprietary PGN.

Each of these methods has an appropriate use. Destination-specific parameter groups are needed where the same message can be directed to one or another destination. A torque control message is defined in ISO 11783 which can be sent to an engine. In the case of more than one engine, this message is to be sent only to the desired engine and a destination-specific parameter group is required and assigned.

PDU2 format communications apply in several situations, including messages sent from single or multiple sources to a single destination, and those sent from single or multiple sources to multiple destinations. PDU2 communications cannot be used where a message is to be sent to one or another destination and not to both.

Proprietary communications are provided by the use of the proprietary PGN. Different PGN have been assigned for non-destination-specific proprietary communications and for destination-specific proprietary communications. This allows for two alternative functions:

a) a specific source can send its proprietary message in a PDU2 Format (non-destination-specific) with the PS field identified as desired by the user;

b) use of PDU1 format (destination-specific) allowing for situations where a diagnostic tool directs its communication to a specific destination out of a possible group of controllers.

   EXAMPLE An engine uses more than one controller but wants to be able to perform diagnostics while all of its controllers are connected to the same network. In this case, the proprietary protocol needs to be able to be destination-specific.

Proprietary communications are useful in two situations:

— when it is unnecessary for standardized communications;

— when it is important to communicate proprietary information.

Much of the communication between controllers constructed by a single manufacturer does not require standardization. The information that is communicated is not generally useful to other controllers on the network. In this situation, the proprietary parameter group can be used.

When parameter group assignment is contemplated, proprietary and PDU2 format communications methods should be considered. If proprietary information is being communicated, or if the information to be communicated is not of general interest, the proprietary method should be used. If the information is of general interest and does not require direction of the message to a particular control function, a PDU2 format assignment should be sought. Finally, if the information is of general interest but requires direction to one or another control function, then destination-specific addressing is needed and a destination PDU1 format parameter group assignment should be sought.

5.9.4 Data field definition

Minimizing message overhead with CAN-based systems requires full use of the data field (all 8 byte). Except in the case of very time critical messages, related parameters should be grouped to fill the eight-byte data field. Following this principal conserves PGN for future assignment. Strong justification is needed to allow definition of parameter groups that result in sparsely used data fields.

Table 7 — ISO 11783 PGN template

Legend:

EDP	= Extended Data Page	(1 bit)			
DP	= Data Page	(1 bit)	GE	= Group Extension	(8 bit)
PF	= PDU Format	(8 bit)	P	= Priority	(1 bit)
PS	= PDU Specific Field	(8 bit)	NA	= Not Allowed	
DA	= Destination Address	(8 bit)	un	= Undefined	
PGN	= Parameter Group Number	(3 byte)			

EDP	DP	PF	PS	Parameter Group Definition	Multipacket	PGN
0	0	0	DA	PDU1 Format - 100 ms or less	NA	000
0	0	1	DA			256
	Boundary x					
0	0	238	DA	PDU1 Format - Greater than 100 ms	Allowed	60928
0	0	239	DA	PDU1 Format - Proprietary A	Allowed	61184
0	0	240	0	PDU2 Format - 100 ms or less	NA	61440
0	0	240	1			61441
	Boundary y					
0	0	254	254			65278
0	0	254	255	PDU2 Format - Greater than 100 ms	Allowed	65279
0	0	255	un	PDU2 Format - Proprietary B	Allowed	65280-65535
0	1	0	DA	PDU1 Format - 100 ms or less	NA	65536
0	1	1	DA			65792
	Boundary x1					
0	1	238	DA	PDU1 Format - Greater than 100 ms	Allowed	126464
0	1	239	DA	PDU1 Format - Proprietary A2	Allowed	126720
0	1	240	0	PDU2 Format - 100 ms or less	NA	126976
0	1	240	1			126977
	Boundary y1					
0	1	254	254			130814
0	1	254	255	PDU2 Format - Greater than 100 ms	Allowed	130815
0	1	255	un	PDU2 Format - Proprietary B	Allowed	130816-131071

5.10 Transport protocol functions

5.10.1 General

Transport protocol functions are described as a part of the data link layer with the recognition that the Transport protocol functionality is subdivided into two major functions: message "packetization" and reassembly, and connection management. These are described in 5.10.2 to 5.11.6, in which the term originating control function corresponds to the control function that transmits the Request to Send message, and receiving control function corresponds to the control function that transmits the Clear to Send message.

There are two kinds of transport protocol functions. For messages greater than 8 byte and up to 1 785 byte in length, transport protocol functions are used. If the message length is greater than 1 785 byte, then extended transport protocol functions are necessary. The transport protocol functions are harmonized with SAE but the extended transport protocol functions are not (See 5.11).

5.10.2 "Packetization" and reassembly

5.10.2.1 General

Messages greater than 8 byte in length are too large to fit into a single classical CAN data frame. Therefore, they are broken into several smaller packets, and those packets are transmitted in separate classical CAN data frames. At the receiving control function, the individual message frames are received and reassembled in order of sequence number of the received packets.

5.10.2.2 Message packets

The classical CAN data frame includes an 8 byte data field. Because the individual packets which comprise a large message have to be identified individually so that they can be reassembled correctly, the first byte of the data field is defined as the sequence number of the packet.

Individual message packets are assigned a sequence number of one to 255. This yields a maximum message size of 255 packet × 7 byte/packet = 1 785 byte when transport protocol is used.

5.10.2.3 Sequence Number

Sequence numbers are assigned to packets for transmission on the network during message "packetization" and then used on reception of packets to reassemble them back into a message.

Sequence numbers shall be assigned to individual packets beginning with one and continuing sequentially until the entire message has been "packetized" and transmitted. The packets shall be sent sequentially in ascending order starting with packet 1.

5.10.2.4 "Packetization"

A multi-packet message is defined as one whose data does not fit into the data field of a single CAN data frame (i.e. messages with a data field greater than 8 byte).

For the purposes of this protocol, the large message is considered to be a parameter group with a string of nine or more bytes. The first data transfer packet contains the sequence number one and the first seven bytes of the string. The second seven bytes are placed into another data frame along with the sequence number 2, the third with sequence number 3, and so on until all the bytes in the original message have been placed into ISO 11783-specified classical CAN data frames and transmitted.

Each data transfer packet (other than the last packet in a transmission sequence) shall include 7 byte of the original large message. The final packet includes a data field of 8 byte, this being the sequence number of the packet and at least 1 byte of data related to the parameter group, and then any remaining, unused, bytes set to FF_{16}.

The time between packets for multi-packet broadcast messages shall be 10 ms to 200 ms (see 5.13.3). For multi-packet messages directed to a specific destination, the transmitting control function maintains a maximum time between packets (where CTS allows more than one) of 200 ms with no minimum time requirement. Receiving control functions shall be aware that the packets containing the data all have the same identifier.

5.10.2.5 Reassembly

Data packets are received sequentially. Each data packet of a multi-packet message shall be assembled, in order of sequence number, into a single string of bytes. This string of bytes is passed to the application entity responsible for the large message.

5.10.3 Transport Protocol — Connection management

5.10.3.1 General

Connection management is concerned with the opening, use and closure of virtual connections between control functions for destination-specific transfers. A virtual connection in the ISO 11783-specified environment is considered a temporary association of two control functions for the purpose of transferring a single large message that is described by a single PGN (see Figures B.1 and B.2). In cases where the connection is from one to many, there is no flow control or closure provided (see, Figure B.3).

5.10.3.2 Multi-packet broadcast

Multi-packet messages can be non-destination-specific, i.e. they can be broadcast messages. To broadcast a multi-packet message, a control function first transmits a Broadcast Announce Message (BAM). This message, which shall be transmitted to the global destination address, constitutes a large message warning to the control functions on the network. The BAM message contains the PGN of the large message to be broadcast, its size and the number of packets into which it has been packeted. Control functions interested in the broadcast data are then required to allocate the resources necessary to receive and reassemble the message. The Data Transfer PGN (PGN = 60 160) is then used to send the associated data.

5.10.3.3 Connection initiation

A connection is initiated when a control function transmits a Request to Send message to a DA. The Request to Send message contains the size of the entire message in bytes, the number of separate messages in which it will be transferred, the maximum number of packets that can be sent in response to one CTS, and the PGN of the message being transported.

Upon receipt of a Request to Send message, a control function can elect to accept the connection or to reject it. To accept the connection, the receiving control function transmits a Clear to Send message. The Clear to Send message contains the number of message packets it can accept and the sequence number of the first packet it is expecting. The receiving control function shall ensure that it has sufficient resources to handle the number of packets of which it is accepting delivery. The sequence number of the packet, in the instance of a freshly opened connection, is 1.

NOTE The Clear to Send message cannot include provision for all the component frames of the message.

To reject the connection, the control function responds with a Connection Abort message. The connection can be rejected for any reason, although lack of resources, memory, etc. can be the cause.

The connection is considered established for the originating control function (i.e. RTS control function) when the originating control function receives a corresponding CTS from the receiving control function (i.e. CTS control function). The connection is considered established for the receiving control function when it has successfully transmitted its CTS message in response to an RTS. These definitions are used to determine when a Connection Abort is to be sent to close a connection. A receiving control function should send a Connection Abort if it has looked at the RTS message and decided not to establish the connection. This allows the originating control function to move on to a new connection without having to wait for a timeout.

5.10.3.4 Data transfer

5.10.3.4.1 General

Data transfer begins after the originating control function of the connection receives the Clear to Send message. An exception is if the data transfer was the result of the BAM — in this case, the Clear to Send message is not used. The Data Transfer packet uses the Data Transfer PGN (see 5.10.5) but the data bytes contained in this packet apply to the PGN that was announced in bytes 6 to 8 of the Request to Send or Broadcast Announce message. The first byte of the data field contains the sequence number of the packet.

5.10.3.4.2 Connection Flow Control

In the case of destination-specific messages, the receiving control function is responsible for coordinating flow control between the control functions. If the receiving control function wants to stop data flow momentarily while a connection is open, it shall use the Clear to Send message, setting the number of packets to send equal to zero. In the case where the flow is required to be stopped for some number of seconds, the receiving control function shall repeat the transmission of the Clear to Send message once per T_h s (0,5 s) in order to ensure the originating control function the connection is not broken. All remaining bit fields are set to ones ("Don't care").

5.10.3.5 Connection closure

Two connection closure cases exist in the absence of errors: the first for a global destination and the second for a specific destination. In the case of the global destination, no connection closure operation is performed beyond the reception of the data itself (see 5.10.4.1 and 5.10.4.5). In the case of destination-specific transfer and upon receipt of the last packet in the message stream, the receiving control function transmits an End of Message acknowledgement to the originating control function of the message. This is the signal to the originating control function that the connection is considered closed by the receiving control function. The End of Message ACK closure is required to free the connection for subsequent use by other control functions.

The Connection Abort message is not allowed to be used by receiving control functions in the case of a global destination message (5.10.4 and 5.10.4.5). In the case of a destination-specific transfer, either the originating or the receiving control function can, at any time, use Connection Abort to terminate the connection. (See 5.10.3.3 for an explanation of when a connection is considered established for the sending and receiving control function.) If the receiving control function, for example, determines that there are no resources available for processing the message, it can abort the connection by issuing the Connection Abort message with abort reason 2 (see Table 8). Upon receipt of this message, any message packet already passed is abandoned.

A failure of either control function can also cause closure of a connection.

EXAMPLE 1 A time delay of more than T1 s from receipt of the last packet when more were expected (CTS allowed more).

EXAMPLE 2 A time delay of more than T2 s after a CTS was transmitted (originating control function failure).

EXAMPLE 3 A lack of CTS or ACK for more than T3 s after the last packet was transmitted (receiving control function failure).

EXAMPLE 4 A lack of CTS for more than T4 s after CTS (0) message to "hold the connection open".

Any of these examples causes a connection closure to occur.

The timeout values are T_r = 200 ms, T_h = 500 ms, T1 = 750 ms, T2 = 1 250 ms, T3 = 1 250 ms and T4 = 1 050 ms (see also 5.13.3 and Figure B.1 regarding timeouts). When either the originating control function or receiving control function decides to close out a connection for any reason including a timeout, it shall send a Connection Abort message with abort reason 3 from Table 8.

With the definitions in this subclause and those given throughout 5.10, the following observations can be made.

a) Connection closure for a broadcast announce message includes the following. A connection is considered closed

   1) when the originating control function sends the last data transfer packet,

   2) when the receiving control function

      i) receives the last data transfer packet, or

      ii) has a T1 connection timeout.

b) Connection closure for Request to Send/Clear to Send messages includes the following. A connection is considered closed

   1) when the originating control function

      i) receives the TP.CM_EndOfMsgACK at the completion of the data transfer for the entire PGN,

      ii) sends a Connection Abort for any reason (e.g. due to a T3 or T4 timeout), or

      iii) receives a Connection Abort,

   2) when the receiving control function

      i) sends the TP.CM_EndOfMsgACK at the completion of the data transfer for the entire PGN,

      ii) receives a Connection Abort, or

      iii) sends a Connection Abort for any reason (including stopping the session early if desired, for a T1 or T2 connection timeout, etc.).

5.10.4 Transport Protocol — Connection management messages (TP.CM)

5.10.4.1 Transport Protocol Connection management message definition

This type of message is used to initiate and close connections and also to control flow. Transport protocol provides the following five transport protocol connection management messages: the Connection Mode Request to Send, the Connection Mode Clear to Send, the End of Message Acknowledgement, the Connection Abort, and the Broadcast Announce Message. The format of this message is shown below in the Parameter Group definition for Transport Protocol — Connection Management.

Parameter group name:	TRANSPORT PROTOCOL — CONNECTION MANAGEMENT (TP.CM)
Definition:	Used for the transfer of parameter groups with 9 byte or more of data. A definition of each specific message defined as part of the transport protocol is contained in the following clauses.
Transmission repetition rate:	per the PGN to be transferred
Data length:	8 byte
Data page:	0
PF:	236

PS: DA

Default priority: 7

Parameter group number: 60 416 ($00EC00_{16}$)

Data ranges for parameters used by this group function:

Control byte:	0 to 15, 18, 20 to 31, 33 to 254 reserved for assignment in a future International Standard
Total message size, byte:	9 to 1 785 (2 byte), 0 to 8 and 1 786 to 65 535 not allowed
Total number of packets:	2 to 255 (1 byte), zero and 1 not allowed
Maximum number of packets:	2 to 255 (1 byte), zero and 1 not allowed
Number of packets that can be sent:	0 or 1 to 255 (1 byte)
Next packet number to be sent:	1 to 255 (1 byte), zero not allowed
Sequence number:	1 to 255 (1 byte), zero not allowed

NOTE A zero in "Number of packets that can be sent" means put connection on hold (see 5.10.3.4.1).

Connection Mode Request To Send (TP.CM_RTS): destination-specific

Byte:	1	Control byte = 16, Destination Specific Request_To_Send (RTS)
Bytes:	2 to 3	Total message size, number of bytes
Byte:	4	Total number of packets
Byte:	5	Maximum number of packets that can be sent in response to one CTS. FF_{16} indicates that no limit exists for the originating control function
Byte:	6	PGN of requested information (8 LSB of parameter group number, Bit 8 MSB)
Byte:	7	PGN of requested information (second byte of parameter group number, Bit 8 MSB)
Byte:	8	PGN of requested information (8 MSB of parameter group number, Bit 8 MSB)

Connection Mode Clear To Send (TP.CM_CTS): destination-specific

Byte:	1	Control byte = 17, Destination Specific Clear_To_Send (CTS)
Byte:	2	Number of packets that can be sent. This value shall be no larger than the smaller of the two values in Byte 4 and Byte 5 of the RTS message
Byte:	3	Next packet number to be sent
Bytes:	4 to 5	Reserved for assignment in a future International Standard, these bytes shall be sent as FF_{16}
Bytes:	6 to 8	PGN of packeted message

End of Message Acknowledgement (TP.CM_EndofMsgACK): destination-specific

Byte:	1	Control byte = 19, End_of_Message Acknowledge
Bytes:	2 to 3	Total message size, number of bytes
Byte:	4	Total number of packets
Byte:	5	Reserved for assignment in a future International Standard, this byte shall be sent as FF_{16}
Bytes:	6 to 8	PGN of packeted message

Connection Abort (TP.Conn_Abort): destination-specific

Byte:	1	Control byte = 255, Connection Abort
Byte:	2	Connection Abort reason
Bytes:	3 to 5	Reserved for assignment in a future International Standard, these bytes shall be sent as FF_{16}
Bytes:	6 to 8	PGN of packeted message

Broadcast Announce Message (TP.CM_BAM): global destination

Byte:	1	Control byte = 32, Broadcast Announce Message
Bytes:	2 to 3	Total message size, number of bytes
Byte:	4	Total number of packets
Byte:	5	Reserved for assignment in a future International Standard, this byte should be sent as FF_{16}
Bytes:	6 to 8	Parameter Group Number of packeted message

5.10.4.2 Transport Protocol Connection Mode Request To Send (TP.CM_RTS)

The TP.CM_RTS message informs a control function that another control function on the network wishes to open a virtual connection with it. The TP.CM_RTS is a message with the SA field set to that of the originating control function, DA set to that of the intended receiving control function of a large message, and the remaining fields set appropriately for the PGN being sent. Byte 5 of this message allows the originating control function to limit the receiving control function's number of packets specified in the Clear To Send message (see Figures B.4 and B.5). When the receiving control function complies with this limit, it ensures that the originating control function can always retransmit packets that, for whatever reason, the receiving control function has not received. If multiple RTS are received from the same SA for the same PGN, then the most recent RTS shall be acted on and the previous RTS abandoned. No abort message shall be sent for the abandoned RTS in this specific case. TP.CM_RTS is only transmitted by the originating control function.

5.10.4.3 Transport Protocol Connection Mode Clear To Send (TP.CM_CTS)

The TP.CM_CTS message is used to respond to the Request To Send message. It informs the peer control function that it is ready for a certain amount of large message data. The amount of large message data cleared to send shall be no larger than the smaller of the two values in Byte 4 and Byte 5 of the originating control function TP.CM_RTS message. If multiple CTS are received after a connection is already established, then the connection shall be aborted. When the originating control function aborts the connection, it shall send the Connection Abort message with abort reason 4 from Table 8. The receiving control function shall not send the next CTS until it has received the last data packet from the previous CTS or has timed out. In the case of time out the receiving control function has the choice whether to send a connection abort or to send a CTS. The following cases exist when data transfer happens with errors:

— A missing or errant packet(s) is detected and the last packet is successfully received, then the receiving control function sends a CTS requesting retransmission starting from the missing packet.

— Missing packet(s) including the last packet leads to time out T1. In this case, the receiving control function decides to either send a CTS or a connection abort with reason 4 from Table 8. See Figure B.7.

When the CTS is used to request the retransmission of data packet(s), it is recommended not to use more than 2 retransmit requests. When this limit is reached, a connection abort with abort reason 5 from Table 8 shall be sent. If a CTS is received while a connection is not established, it shall be ignored. CTS not only control the flow but also confirm correct receipt of any data packet prior to that CTS packet's number. Therefore, if information for the previous CTS was corrupted, then a CTS for the corrupted information shall be sent before continuing on to the next sequential packets to be sent. Because of this requirement, the originating control function of a large message transmission can use Byte 5 of the TP.CM_RTS message as a way to ensure the possibility of retransmission of a packet within the last set of packets cleared to send. TP.CM_CTS is only transmitted by the receiving control function.

5.10.4.4 Transport Protocol End of Message Acknowledgement (TP.CM_EndofMsgACK)

The TP.CM_EndofMsgACK message is passed from the receiving control function of a large message to its originating control function, indicating that the entire message was received and reassembled correctly. The receiving control function can keep the connection open after the last Data Transfer of the session by not immediately sending the TP.CM_EndOfMsgACK. This allows the receiving control function to have a packet resent if necessary. If an End of Message Acknowledgement is received by the originating control function prior to the final Data Transfer, then the originating control function ignores it. One End of Message Acknowledgement is sent to inform the originating control function that the large message transfer has been received and assembled correctly. TP.CM_EndOfMsgACK is only transmitted by the receiving control function.

5.10.4.5 Transport Protocol Connection Abort (TP.Conn_Abort)

The TP.Conn_Abort message is used by either control function involved in a virtual connection to close the connection without completing the transfer of the message or to prevent a connection from being initialized.

Upon receipt of a Connection Mode Request To Send message, a control function shall determine if there are sufficient resources available to deal with the message for which this connection is sought, for example, if the control function has to acquire memory from the system heap and cannot claim enough to accept the entire message, or a control function is simply too occupied doing other tasks to expend processor cycles handling a large message. In these cases, a Connection Abort message can be sent, even though the connection has not been established. This can be done in order to allow the originating control function to attempt another virtual connection without first having to wait for a timeout to occur.

When, for any reason, either the originating or receiving control function decides to close out a connection prior to completing the data transfer, including a timeout, it shall send a Connection Abort message with the appropriate Connection Abort reason. See Table 8 for the list of Connection Abort reasons.

The originating control function (i.e. the RTS control function) shall immediately stop transmitting after the reception of the Connection Abort message by the CAN protocol control function. If this is not possible, the process to stop transmitting data packets shall take no more than 32 data packets and shall not exceed 50 ms. After sending or receiving a Connection Abort message, all related data packets received shall be ignored. TP.Conn_Abort is transmitted by the originating control function or the receiving control function.

Table 8 — Connection Abort reasons

Value	Description
0	Reserved for assignment in a future International Standard
1	Already in one or more connection-managed sessions and cannot support another
2	System resources were needed for another task so this connection managed session was terminated
3	A timeout occurred and this is the connection abort to close the session
4	CTS messages received when data transfer is in progress
5	Maximum retransmit request limit reached
6	Unexpected data transfer packet
7	Bad sequence number (and software is not able to recover)
8	Duplicate sequence number (and software is not able to recover)
9	"Total message size" is greater than 1 785 byte
10 to 249	Reserved for assignment in a future International Standard
250	If a Connection Abort reason is identified that is not listed in the table use code 250
251 to 255	According to ISO 11783-7 definitions

5.10.4.6 Broadcast Announce Message (TP.CM_BAM)

The TP.CM_BAM is used to inform all the control functions of the network that a large message is about to be broadcast. It defines the parameter group and the number of bytes to be sent. After the transmission of the TP.CM_BAM message, Data Transport messages are sent containing the "packetized" broadcast data.

TP.CM_BAM is only transmitted by the originating control function.

5.10.5 Transport Protocol — Data Transfer messages (TP.DT)

The TP.DT message is used to communicate the data associated with a parameter group. The TP.DT message is an individual packet of a multi-packet message transfer. For example, if a large message had to be divided into five packets in order to be communicated then there would be five TP.DT messages. (See Annex B for examples of TP.DT messages in use.) The format of this message is shown in the following parameter group definition.

TP.DT is only transmitted by the originating control function.

Parameter group name:	TRANSPORT PROTOCOL — DATA TRANSFER (TP.DT)
Definition:	Used for the transfer of data associated with Parameter Groups that have more than 8 byte of data.
Transmission repetition rate:	per the PGN to be transferred
Data length:	8 byte
Data page:	0
PF:	235
PS:	DA [global (DA = 255) for TP.CM.BAM data transfers; global not allowed for RTS/CTS data transfers]
Default priority:	7 (Priority used regardless of the PGN being transported)
Parameter group number:	60 160 ($00EB00_{16}$)

Data ranges for parameters used by this message type:

Sequence Number:		1 to 255 (1 byte)
Byte:	1	Sequence Number
Bytes:	2 to 8	"Packetized" data (7 byte)

The last packet of a multi-packet parameter group can require less than 8 byte. The extra bytes shall be filled with FF_{16}.

5.10.6 Transport Protocol Connection constraints

5.10.6.1 General

If a control function cannot handle another session, then it should reject the connection initiations that are pursued by other control functions. An RTS for a different PGN from the same SA to the same destination as an existing session shall be rejected as well. In either case, the newly requested session should be rejected by sending a Connection Abort with abort reason 1 from Table 8. This allows the device desiring a connection to move on to a new connection without having to wait for a timeout.

5.10.6.2 Number and type of connections a control function shall support

Each control function on the network can originate one destination-specific transport protocol connection transfer with a given destination at a time. This is because TP.DT only contains the SA and DA and not the PGN of the data being transferred. An extended transport protocol session may be open in parallel with a transport protocol connection.

Only one multi-packet BAM (i.e. global destination) can be sent from an originating control function at a given time. This is because TP.DT does not contain the actual PGN or a connection identifier. However, receiving control functions shall recognize that multiple multi-packet messages can be received, interspersed with one another, from different originating control functions (i.e. source addresses).

A control function shall also be able to support one RTS/CTS session and one BAM session concurrently from the same SA. Therefore, the receiving control function shall use the destination address of the two transport protocol messages to keep them properly separated. One of the transport protocol messages has a global and the other a specific DA. The DA shall be used to distinguish the difference between the two because the TP.DT contains neither the actual PGN nor a connection identifier.

Regardless of whether a control function can support multiple simultaneous transport protocol sessions (RTS/CTS and/or BAM), it shall ensure that TP.DT/ETP.DT messages from the same SA but having different DA can be differentiated. Receiving control functions shall use the DA and SA to keep the data for the messages correct. The transmitter of parallel TP/ETP sessions shall be aware that the receiver can put one or both sessions on hold for any period of time (See 5.10.3.4.1), and therefore it is not possible to predict which of the sessions will be the first to complete.

5.10.6.3 Intended transport protocol use

Transport Protocol has been developed to provide a mechanism for transferring PGN with nine or more data bytes (see 5.2.8.2). A PGN defined as multi-packet capable having less than nine data bytes to transfer in a specific instance shall be sent in a single classical CAN data frame with the DLC set to 8 (see 5.2.8.1).

5.10.6.4 Concurrent PGN reception

It is possible that specific parameter groups can be sent in a non-transport-protocol form when they are less than or equal to 8 byte, and then also be sent in transport protocol form when they are greater than 8 byte. It is possible for these two forms of the same PG to be sent concurrently.

NOTE A non-transport-protocol form of a PGN is not considered to be a session, so its being sent does not close out the transport protocol form of the same PGN.

5.11 Extended transport protocol functions

5.11.1 Overview

Extended transport protocol functions are described as part of the data link layer in a manner similar to transport protocol functions. Due to the similarity, it is understood that 5.10 is the basis for the requirements, and this subclause focuses on the unique characteristics of extended transport protocol. While references to 5.10 define common behaviour and requirements, the methods in this subclause are implemented with the extended transport protocol messages (e.g. a reference to TP.CM_CTS defines the requirement, which is implemented with the ETP.CM_CTS message).

5.11.2 General

Messages greater than 1 785 byte in length are too large to fit into a transport protocol session. Therefore, they are broken into several smaller packets, and those packets are transmitted in separate classical CAN data frames with connection management of those frames that permits the receiving control function to reassemble the message in order of sequence number.

5.11.3 Message packets

The maximum number of packets that can be sent in single connection with extended transport protocol is restricted by the extended data packet offset (3 byte). This yields a maximum message size of $2^{24} - 1$ packet × 7 byte/packet = 117 440 505 byte for extended transport protocol.

5.11.4 Extended Transport Protocol — Connection Management

5.11.4.1 General

Extended functions of transport protocol are required when a control function requires the transfer of messages longer than the transport protocol length of 1 785 byte.

Extended transport protocol uses five control bytes for extended message connection requests and responses. These control bytes and messages are as follows.

— Control Byte 20 Extended message Request To Send

— Control Byte 21 Extended message Clear To Send

— Control Byte 22 Extended message Data Packet Offset

— Control Byte 23 Extended message End of Message Acknowledgement

— Control Byte 255 Extended message Connection Abort

The maximum number of bytes that can be sent in single connection with extended message transport protocol is 117 440 505 byte.

The sequence of messages is shown in Figure 8. The originating control function sends an Extended Message Request to Send (ETP.CM_RTS) to the receiving control function. The receiving control function sends an ETP.CM_CTS back to the originating control function with the number packets the originating control function can send, which can be between 0 and 255, and the packet number of the initial packet to send for the first session of transport protocol data transfer messages. Setting number of packets to send to zero, holds the connection open between the receiving and the originating control function. See 5.10.4.3.

The originating control function then sends the Extended Data Packet Offset (ETP.CM_DPO) followed by the Extended Transport Protocol Data Transfer messages with the required number of data packets. It then waits for the next ETP.CM_CTS message from the receiving control function before repeating with the next Extended Data Packet Offset message and the required number of Extended Transport

Protocol Data Transfer messages. The actual sequence of each packet in the extended message is equal to the extended data packet offset last received added to the sequence number in the extended transport protocol data transfer message.

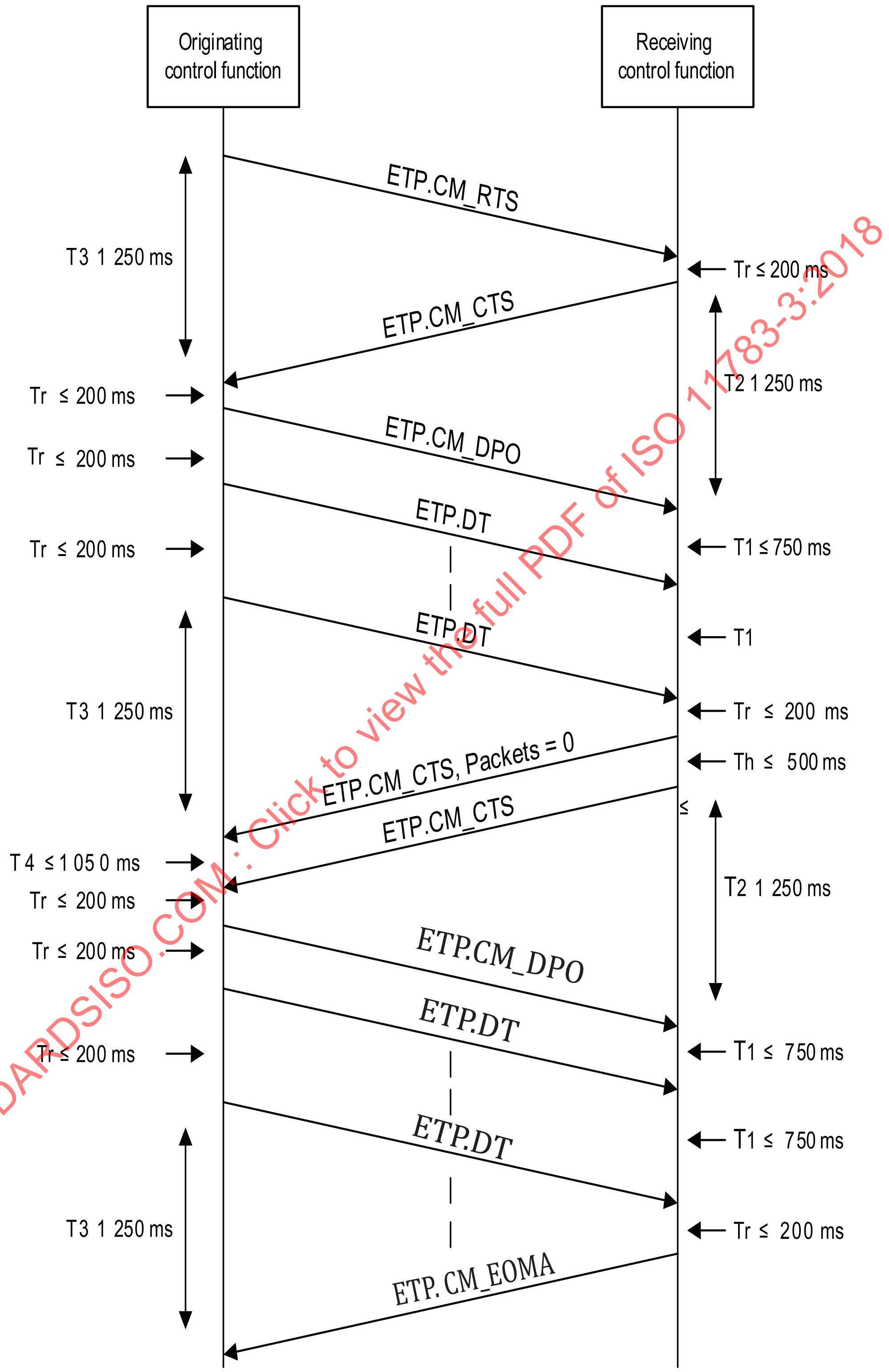


Figure 8 — Extended transport message sequence

The timing for extended transport protocol is the same as transport protocol.

5.11.5 Extended Transport Protocol — Connection Management messages (ETP.CM)

5.11.5.1 General

Extended transport protocol connection management messages use a PGN different from that used for the transport protocol (see 5.10.4).

5.11.5.2 Connection management message definition

Parameter group name:	EXTENDED TRANSPORT PROTOCOL — CONNECTION MANAGEMENT (ETP.CM)
Definition:	Used for the transfer of parameter groups that have 1 786 byte or more of data. A definition of each specific message defined as part of the extended transport protocol is in the following.
Transmission repetition rate:	per the PGN to be transferred
Data length:	8 byte
Data page:	0
PF:	200
PS:	DA
Default priority:	7
Parameter group number:	51 200 ($00C800_{16}$)

Data ranges for parameters used by this group function:

Control byte:	20, 21, 22, 23, 255 for extended connection management messages
	0–19, 24–254 reserved for assignment in a future International Standard

Total message size, byte:	1 786 to 117 440 505 (4 byte) ((2^{24} – 1) * 7)
Total number of packets:	256 to 16 777 215 (3 byte), zero to 255 not allowed
Number of packets that can be sent:	256 to 16 777 215 (3 byte), zero to 255 not allowed
Next packet number to be sent:	1 to 16 777 215 (3 byte)
Data packet offset number:	0 to 16 777 214 (3 byte)
Sequence number:	1 to 255 (1 byte), zero not allowed

Extended Connection Mode RTS (ETP.CM_RTS): destination-specific

Byte:	1	Control byte = 20, Extended Request to Send
Bytes:	2 to 5	Number of bytes to transfer (1 786 byte to 117 440 505 byte max.)
Bytes:	6 to 8	PGN of extended packeted message

Extended Connection Mode CTS (ETP.CM_CTS): destination-specific

Byte:	1	Control byte = 21, Extended Clear to Send
Byte:	2	Number of packets to send (0 or 1 to 255)
Bytes:	3 to 5	Next packet number to send (1 to 16 777 215)
Bytes:	6 to 8	PGN of extended packeted message

NOTE 1 Number of packets to send + Next packet to send – 1 cannot exceed 16 777 215.

NOTE 2 A zero in byte 2 means put connection on hold (see 5.11.3.1).

Extended Connection Mode Data Packet Offset (ETP.CM_DPO): destination-specific

Byte:	1	Control byte = 22, Extended Data Packet Offset
Byte:	2	Number of packets to which to apply the offset (1 to 255)
Bytes:	3 to 5	Data packet offset (0 to n) (Always 1 less than bytes 3 to 5 of the ETP.CM_CTS)
Bytes:	6 to 8	PGN of extended packeted message

NOTE 3 n + sequence number of any packet cannot exceed 16 777 215.

ETP.CM_DPO byte 2 shall be less than or equal to the ETP.CM_CTS byte 2. If byte 2 is less than byte 2 of the ETP.CM_CTS message, then the receiver shall make necessary adjustments to its session to accept the data block defined by the ETP.CM_DPO message and the subsequent ETP.DT packets.

Extended End-of-Message Acknowledgement (ETP.CM_EOMA): destination-specific

Byte:	1	Control byte = 23, Extended End-of-Message Acknowledgement
Bytes:	2 to 5	Number of bytes transferred (1 786 byte to 117 440 505 byte)
Bytes:	6 to 8	PGN of extended packeted message

Extended Connection Abort (ETP.Conn_Abort): destination-specific

Byte:	1	Control byte = 255, Connection Abort
Byte:	2	Connection Abort reason
Bytes:	3 to 5	Reserved for assignment by ISO, these bytes should be set to FF_{16}
Bytes:	6 to 8	PGN of packeted message

5.11.5.3 Extended Connection Mode Request To Send (ETP.CM_RTS)

The ETP.CM_RTS message informs a control function that another control function on the network wishes to open a virtual connection with it for extended transport protocol. It consists of a message with the SA field equal to that of the originating control function, DA set to that of the intended receiving control function of a large extended message, and the remaining fields set appropriately for the PGN being sent.

5.11.5.4 Extended Connection Mode Clear To Send (ETP.CM_CTS)

The ETP.CM_CTS message is used to respond to the Extended Request to Send message. It informs the peer control function that it is ready for a certain amount of large extended message data. The number of packets to send can be set to 0 to hold the connection open between the control functions. In this case the ETP.CM_CTS “0” is not acknowledged by the ETP.CM_DPO message.

5.11.5.5 Extended Connection Mode Packet Offset (ETP.CM_DPO)

The ETP.CM_DPO establishes the offset from which transmitted packets are numbered within one CTS group of data. The actual extended message sequence number = (ETP.DT Sequence number + ETP.CM_DPO Data Packet Offset).

5.11.5.6 Extended End of Message Acknowledgement (ETP.CM_EOMA)

The ETP.CM_EOMA message is passed from the receiving control function of a large message to its originating control function, indicating that the entire extended message was received and reassembled correctly.

5.11.5.7 Extended Connection Abort (ETP.Conn_Abort)

The ETP.Conn_Abort message is used by either control function involved in a virtual connection to close the connection without completing the transfer of the message.

Upon receipt of an Extended Connection Mode Request to Send message, a control function shall determine if there are sufficient resources available to deal with the message for which this connection is sought — for example, if the control function has to acquire memory from the system heap it could not be able to claim enough to accept the entire message, or a control function is simply too occupied doing other tasks to expend processor cycles handling a large message. The control function can either respond with the number of packets to send set to 0, which holds the connection open between the originating control function and the receiving control function, or send an Extended Connection Abort message to close the connection.

When, for any reason, either the originating or receiving control function decides to close out a connection prior to completing the data transfer, including a timeout, it shall send an Extended Connection Abort message with the appropriate Connection Abort reason. See Table 9. If a Connection Abort reason is identified that is not listed in Table 9, use code 254.

Table 9 — Extended transport protocol Connection Abort reasons

Value	Description
0	Reserved for assignment in a future International Standard
1	Already in one or more connection-managed sessions and cannot support another
2	System resources were needed for another task so this connection managed session was terminated
3	A timeout occurred and this is the connection abort to close the session
4	CTS messages received when data transfer is in progress
5	Maximum retransmit request limit reached
6	Unexpected data transfer packet
7	Bad sequence number (and software is not able to recover)
8	Duplicate sequence number (and software is not able to recover)
9	Unexpected EDPO packet
10	Unexpected EDPO PGN (PGN in EDPO is bad)
11	EDPO number of packets is greater than CTS
12	Bad EDPO offset
13	Deprecated. Use 250 instead (Any other reason)
14	Unexpected ECTS PGN (PGN in ECTS is bad)
15	ECTS requested packets exceeds message size

Table 9 *(continued)*

Value	Description
16 to 249	Reserved for assignment in a future International Standard
250	Any other reason (if a Connection Abort reason is identified that is not listed in the table use code 250)
251 to 255	According to ISO 11783-7 definitions

5.11.6 Extended Transport Protocol — Data Transfer messages (ETP.DT)

The ETP.DT message is used to communicate the data associated with a parameter group. The ETP.DT message is an individual packet of a multi-packet message transfer. For example, if a large message had to be divided into 300 packets in order to be communicated then there would be 300 ETP.DT messages. Extended transport protocol data transfer messages use a PGN different from that used for the transport protocol (see 5.10.5).

Parameter group name: EXTENDED TRANSPORT PROTOCOL — DATA TRANSFER (ETP.DT)

Definition: Used for the transfer of data associated with parameter groups that have more than 1 785 byte of data.

Transmission repetition rate: per the PGN to be transferred

Data length: 8 byte

Data page: 0

PF: 199

PS: DA

Default priority: 7

Parameter group number: 50 944 ($00C700_{16}$)

Data ranges for parameters used by this message type:

Sequence Number: 1 to 255 (1 byte)

Byte: 1 — Sequence number. The first ETP.DT packet following an ETP.CM_DPO message shall always be sequence number 1.

Bytes: 2 to 8 — “Packetized” data (7 byte).

The last packet of a multipacket parameter group can require less than 8 byte. The extra bytes are filled with FF_{16}.

5.11.7 Extended Transport Protocol — Connection constraints

As identified in 5.10.6.2, the transmitter of parallel TP/ETP sessions shall be aware that the receiver can put one or both sessions on hold for any period of time (See 5.11.5.3), and therefore it is not possible to predict which of the sessions will be the first to complete.

5.12 PDU processing requirements

Processing of the PDU requires the following of specific procedures. The suggested sequence for interpreting PDU is described in Annex A. Annex C shows example ISO 11783 message types and PDU formats being used.

Controllers shall be able to process data link messages fast enough to prevent losing messages when the data link is at 100 % utilization. This also means that in low-utilization situations, when there are back-to-back messages, each controller shall be able to process the messages fast enough not to lose messages due to their back-to-back nature. Processing the messages fast enough does not mean that a message has to be immediately generated, but that a new message shall not overrun the previous messages.

5.13 Application notes

5.13.1 High data rates

Data that is to be updated at a high rate and that has tight latency requirements should, if possible, allow hardware-based message filtering to be used.

5.13.2 Request scheduling

The scheduling of a request should be cancelled if the information that is getting ready to be requested is received prior to the request being sent. That is, if the information is received 50 ms prior to request scheduling, then the request should not be issued. Parameter groups should not be requested if they are recommended to be broadcast. Exceptions can arise when the recommended broadcast time exceeds a special case need.

5.13.3 Controller response time and timeout defaults

All controllers, when required to provide a response, shall do so within 0,20 s (T_r). All controllers expecting a response shall wait at least 1,25 s (T3) before giving up or retrying. These times ensure that any latencies due to bus access or message forwarding across bridges do not cause unwanted timeouts. Different time values can be used for specific applications when required. For instance, a 20 ms response can be expected for high-speed control messages. Reordering of any buffered messages can be necessary to reach a faster response. There is no restriction on minimum response time.

Time between packets of a multi-packet message directed to a specific destination is 0 ms to 200 ms. This means that back-to-back messages can occur and they can contain the same identifier. The CTS mechanism can be used to ensure a given time spacing between packets (flow control). The required time interval between packets of a multi-packet broadcast message is 10 ms to 200 ms. A minimum time of 10 ms ensures the receiving control function has time to pull the message from the CAN hardware. The receiving control function shall use a timeout (i.e. T1) of 750 ms. This provides a timeout that is greater than the maximum transmit spacing of 200 ms.

EXAMPLE Use of the 1,25 s (T3) timing.

a) Maximum forward delay time within a bridge is 50 ms. The total number of bridges is equal to 10 (i.e. one tractor + five trailers + four dollies = 10 bridges). Therefore, the total network delay is 500 ms in one direction.

b) Number of request retries is equal to two (three requests total), this applies to the situation where the CTS is used to request the retransmission of data packet(s). If the retransmit request limit is reached, then the connection abort shall be sent with abort reason 5 from Table 8.

c) The margin for timeouts is 50 ms.

Figures B.1 and B.3 show the timing requirements identified. In Figure B.1 the time numbers are computed assuming the worst-case number of bridges, that is, 10 bridges. The timeout numbers for receiving control functions are identified as a time value while originating control function requirements are specified as a less than or equal to time value.

NOTE An originating control function has transmitter and receiving requirements and a receiving control function has transmitter and receiving requirements.

5.13.4 Required responses

A response is required for a global request from all control functions that use the requested parameter group, even the requestor. Acknowledgements are not allowed for global requests.

A control function which uses a global DA (255) for a request (e.g. "address request") shall itself send a response if it has requested the data. This is a requirement because all control functions are expected to respond. If the control function issuing the request does not respond, then the other network control functions can determine the wrong conclusion about the requested information.

5.13.5 Transmission of PGN to specific or global destinations

Most of the time it is desired to send periodically broadcasted PGN to a global destination.

5.13.6 CTS number of packet recommendation

During normal implement operation, it is recommended that the maximum number of packets that can be sent per CTS be set to 16.

Annex A
(informative)

ISO 11783 PDU processing — Typical receive routine

RECEIVE INTERRUPT:

NOTE 1 When a message is received by the microprocessor via the CAN IC, several tests are performed in order to parse it and determine if and where it is to be stored. The three priority bits are used only for bus arbitration and are therefore not needed (used) by the receiving control function.

NOTE 2 A given controller can have more than one address if it performs multiple functions.

```
IF PGN = REQUEST PGN AND THE DESTINATION IS SPECIFIC          ; specific request
THEN
    IF DA = ASSIGNED ADDRESS (destination)
    THEN
        SAVE 4 BYTE ID AND 3 DATA BYTES IN REQUEST QUEUE
IF PGN = REQUEST PGN AND THE DESTINATION IS GLOBAL            ; global request
THEN
    SAVE 4 BYTE ID AND 3 DATA BYTES IN REQUEST QUEUE
IF PF < 240
THEN
    IF DA = GLOBAL                                            ; PDU1 Format (DA = global)
    THEN
        USE JUMP TABLE FOR PGN VALUES OF INTEREST AND
        IF SA = ID OF SPECIAL INTEREST
        THEN
            SAVE 8 BYTES OF DATA IN DEDICATED BUFFER
        ELSE
            SAVE 12 BYTE MESSAGE (ID AND DATA) IN CIRCULAR QUEUE
    ELSE DA = SPECIFIC                                        ; PDU1 Format (DA = specific)
        USE JUMP TABLE FOR PGN VALUES OF INTEREST AND
        IF SA = ID OF SPECIAL INTEREST VALUES
        THEN
            SAVE 8 BYTES OF DATA IN DEDICATED BUFFER
```



```
        ELSE
            SAVE 12 BYTE MESSAGE (ID AND DATA) IN CIRCULAR QUEUE
IF PF > OR = 240                                              ; PDU2 Format
THEN
    USE JUMP TABLE FOR PGN VALUES OF INTEREST AND
        IF SA = ID OF SPECIAL INTEREST
        THEN
            SAVE 8 BYTES OF DATA IN DEDICATED BUFFER
        ELSE
            SAVE 12 BYTE MESSAGE (ID AND DATA) IN CIRCULAR QUEUE
```

Annex B
(informative)

Transport protocol transfer sequences — Examples of connection mode data transfer

Under normal circumstances, the flow model for data transfer follows that shown in Figure B.1. The originating control function sends the TP.CM_RTS indicating that there are 23 byte in the packet message, which will be transferred in four packets. The PGN for the data in the transfer is 65 259, component identification.

The receiving control function replies with a TP.CM_CTS indication that it is ready to process two packets, beginning with packet 1.

The originating control function passes the first two packets across the network using TP.DT. The receiving control function then issues a TP.CM_CTS indicating that it wants to hold the connection open but cannot receive any packets at that moment. A maximum of 500 ms later, it is required to send another TP.CM_CTS message to hold the connection. In this example, it sends another TP.CM_CTS indicating that it can take two more packets, beginning with packet 3. Once packets 3 and 4 have been transferred, the receiving control function transmits a TP.EndOfMsgACK message indicating that all the packets expected were transmitted and that the connection is now considered closed. Note that packet 4 contains 2 byte of valid data, bytes 22 and 23, the remaining data characters in this packet are transmitted as 255 (FF_{16}), data not available, such that the message is 8 byte in length.

Message transfer in the event of an error on the link is shown in Figure B.2. The TP.CM_RTS is transferred and responded to properly, then data are lost during the data transfer phase.

In this situation, the request to send is sent in the same manner as the earlier example. The first two packets are transferred, but packet two fails checksum, or was considered in error by the receiving control function. The receiving control function then transfers a TP.CM_CTS indicating that it wants a single packet and that the packet is packet 2. The originating control function complies, transferring packet 2. The receiving control function then passes a CTS indicating it wants two packets, starting with packet 3. This TP.CM_CTS is the acknowledgement that packets 1 and 2 were received correctly. Once the last packet is received correctly, the receiving control function passes a TP.EndOfMsgACK signalling that the entire message has been correctly received.

In the situation shown in Figure B.3, a control function indicates to the network that it is about to transfer a multi-packet message utilizing the services of the transport protocol. In this example, PGN 65 260, tractor identification is being broadcast to the network. The originating control function first transmits a TP.CM_BAM, followed by the data packets. No acknowledgement is performed by any of the receiving control functions.

In Figure B.4, the originating control function uses the Maximum Number of Packets parameter to limit the number of packets the receiving control function requests to be transferred. In this example, both control functions support the Maximum Number of Packets parameter.

Figure B.5 illustrates the situation where the originating control function supports the RTS parameter, Maximum Number of Packets, but the receiving control function does not. In this situation, the originating control function shall comply with the receiving control function's CTS limits. In this example, the originating control function would have to send seven data transfer packets, even though it preferred to send only five at a time.

CAUTION — In this example, if the receiving control function were to send a CTS for packet 1 after the data transfer of packet 7, it is possible that the originating control function would have to recompute the information and, therefore, the second transmission of packet 1 could contain

