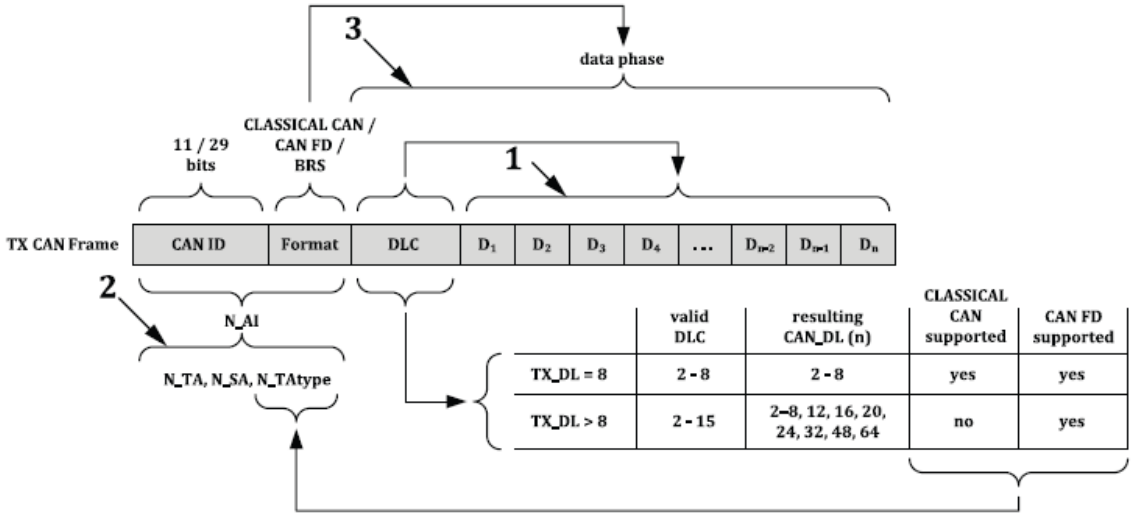
ISO 15765 -2 (2016)

Road vehicles -- Diagnostic communication over Controller Area Network (DoCAN) ---

Part 2: Transport protocol and network layer services

6.2 Illustration of CAN parameters for transport protocol and network layer services



Key

1 DLC value results in a CAN_DL value (n), which is the physical length of a CAN frame data/payload; in the receiver, CAN_DL is used to determine the sender TX_DL value

2 the shown N_AI mapping is an example for normal and normal fixed addressing only

3 the bit rate switch (BRS) in the 'Format' information defines the transmission speed of the data phase

Table 3 — CLASSICAL CAN/CAN FD data length comparison table

Data length code (DLC)	CLASSICAL CAN data length (CAN_DL)	CAN FD data length (CAN_DL)
0	0	0
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	8[a]	12
10	8[a]	16
11	8[a]	20
12	8[a]	24
13	8[a]	32
14	8[a]	48
15	8[a]	64

[a] For CLASSICAL CAN, the DLC values 9..15 are automatically reduced to the value of 8 which leads to the maximum possible CAN_DL for CLASSICAL CAN.

7 Network layer overview

7.1 General

unconformed network layer communication protocol for the exchange of data between network nodes, eg from ECU to ECU, or between external test equipment and an ECU.

7.2 Services provided by network layer to higher layers

two types of services are defined.

a) Communication services

1. N_USData.request : request the transfer of data. may segments the data
2. N_USData_FF.indication : signal the beginning of a segmented message reception to the upper layer
3. N_USData.indication : provide received data to the higher layers
4. N_USData.confirm : comfirms to higher layers that the requested services has been carried out (successfully or not)

b) Protocol parameter setting services

1. N_ChangeParameter.request : request the dynamic setting of specific internal parameters.
2. N_ChangeParameter.confirm : confirms to the upper layer that the request to change a specific protocol has completed (successfully or not)

7.3 Internal operation of network layer

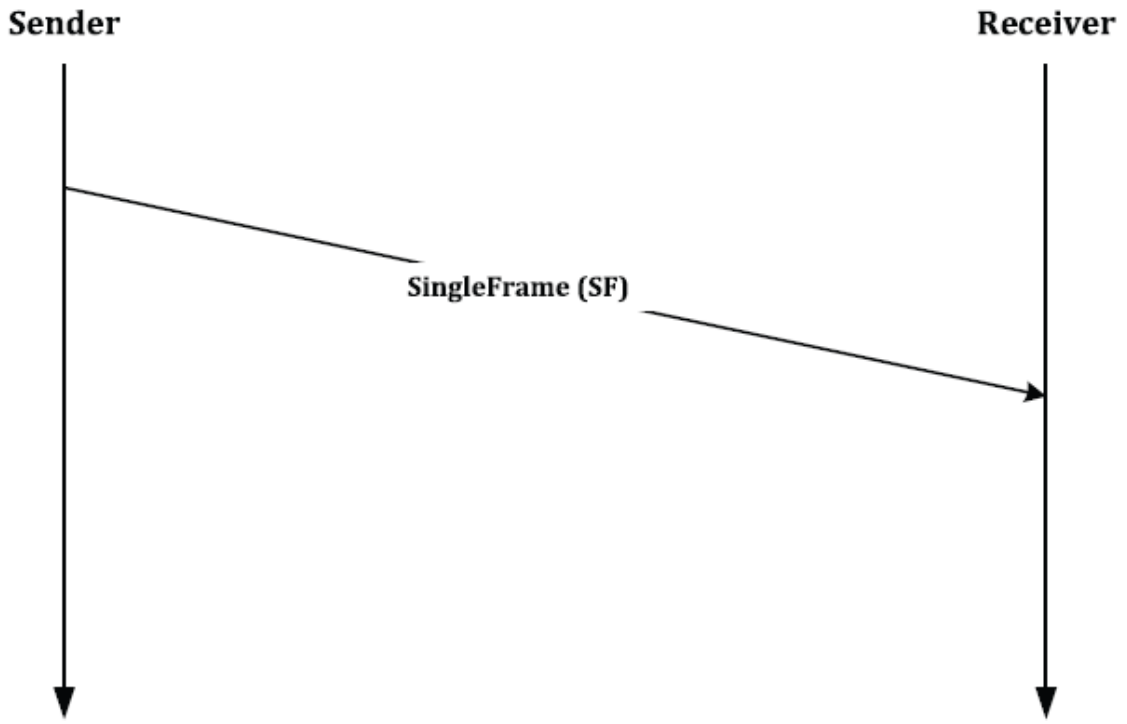


Figure 3 — Example of an unsegmented message

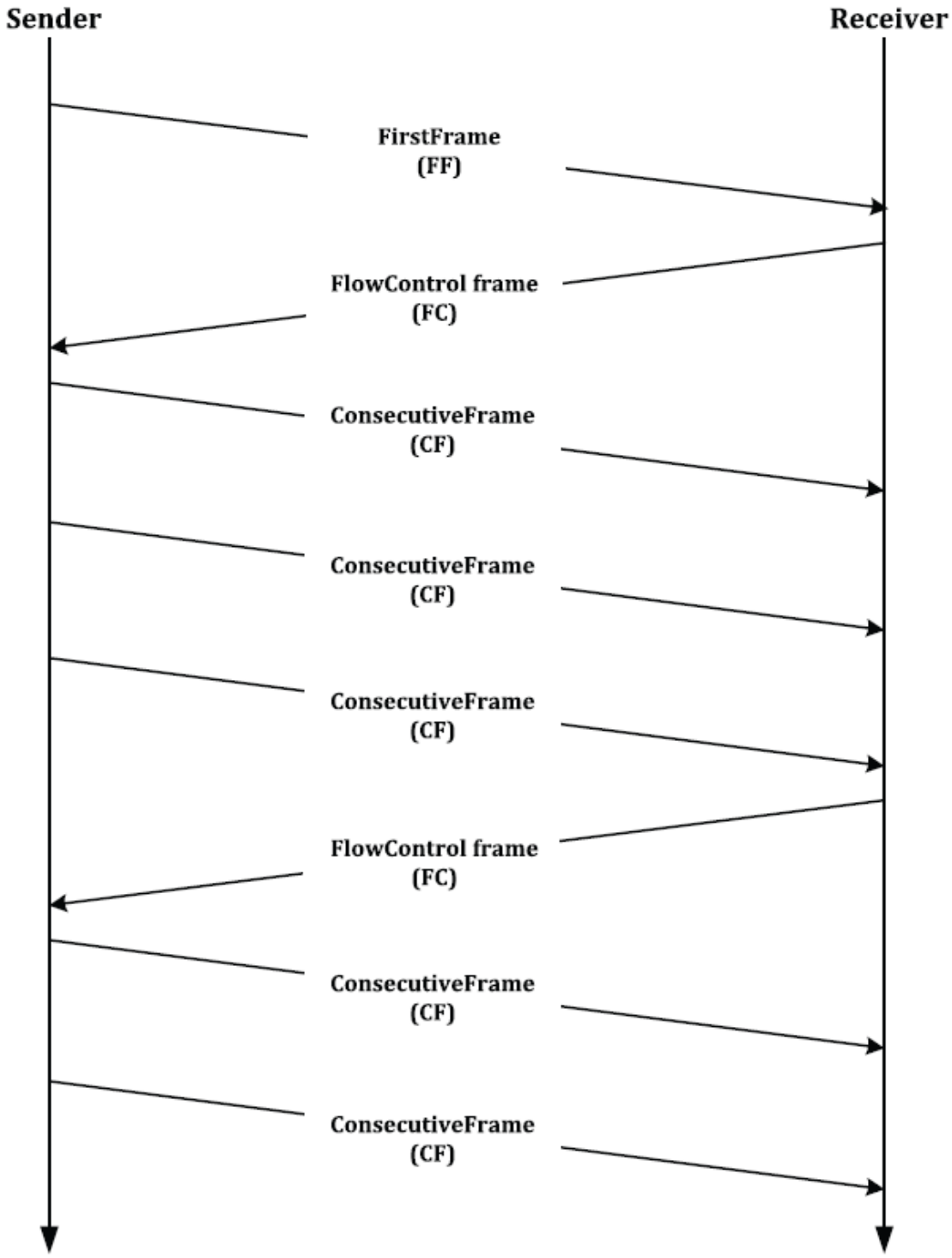


Figure 4 — Example of a segmented message

8 Network layer services

8.1 General

three types of service primitive list as below

- *request*, used by higher communication layers or the application to pass **control information** and **data** required to be transmitted to the network layer
- *indication*, used by the network layer to pass **status information** and **received data** to upper communication layer or the application
- *confirmation*, used by the network layer to pass **status information** to higher communication or the application

8.2 Specification of network layer service primitives

8.2.1 N_USData.request

N_USData.request (

Mtype

N_SA

N_TA

N_TAtype

[N_AE]

<MessageData>

<Length>

)

8.2.2 N_USData.confirm

N_USData.confirm(

Mtype

N_SA

N_TA

N_TAtype

[N_AE]

<N_Result>

)

8.2.3 N_USData_FF.indication

N_USData_FF.indication(

Mtype

N_SA

N_TA

N_TAtype

[N_AE]

<Length>

)

The N_USData_FF.indication service shall always be followed by an U_USData.indication service all from the network layer, indicating the completion (or failure) of message reception.

only issued by the network layer if a correct FF message segment has been received.

8.2.4 N_USData.indication

N_USData.indication(

Mtype

N_SA

N_TA

N_TAtype

[N_AE]

<MessageData>

<Length>

<N_Result>

)

only issued after reception of a SF message or as an indication of the completion (or failure) of a segmented message reception.

8.2.5 N_ChangeParameters.request

N_ChangeParameter.request (

Mtype

N_SA

N_TA

N_TAtype

[N_AE]

<Parameter>

<Parameter_Value>

)

A parameter change is always possible, except after reception of the FF (N_USData_FF.indication) and until the end of reception of the corresponding mesasge (N_USData.indication)

8.2.6 N_ChangeParameter.confirm

N_ChangeParameter.request (

Mtype

N_SA

N_TA

N_TAtype

[N_AE]

<Result_ChangeParameter>

)

8.3 Service data unit specification

8.3.1 Mtype, message type

enumeration, range:

- diagnotics, N_AI consist of N_SA, N_TA, and N_TAtype
- remote diagnotics, N_AI consist of N_SA, N_TA, N_TAtype and N_AE

8.3.2 N_AI, address information

8.3.2.1 N_AI description

N_SA, N_TA, N_TAtype, N_AE

8.3.2.2 N_SA, source address

8 bits

8.3.2.3 N_TA, target address

8 bits

8.3.2.4 N_TAtype, network target address type

Table 4 — Allowed combinations of N_TAtype communication models

N_TAtype	Physical/Functional addressing	<Format>
N_TAtype #1	Physical[a]	CAN base format (CLASSICAL CAN, 11-bit)
N_TAtype #2	Functional[b]	
N_TAtype #3	Physical[a]	CAN FD base format (CAN FD, 11-bit)
N_TAtype #4	Functional[b]	
N_TAtype #5	Physical[a]	CAN extended format (CLASSICAL CAN, 29-bit)
N_TAtype #6	Functional[b]	
N_TAtype #7	Physical[a]	CAN FD extended format (CAN FD, 29-bit)
N_TAtype #8	Functional[b]	

[a] Physical addressing (1 to 1 communication) shall be supported for all types of network layer messages.

[b] Functional addressing (1 to *n* communication) shall only be supported for SingleFrame transmission.

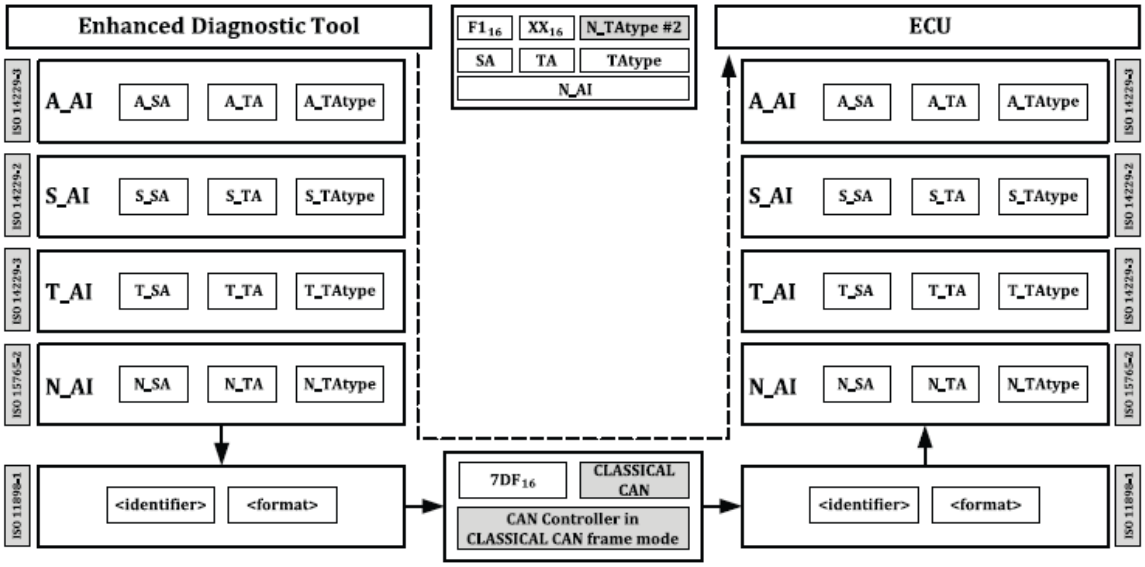


Figure 5 — Example of enhanced diagnostic tool CLASSICAL CAN request for normal addressing (N_TAtype #2)

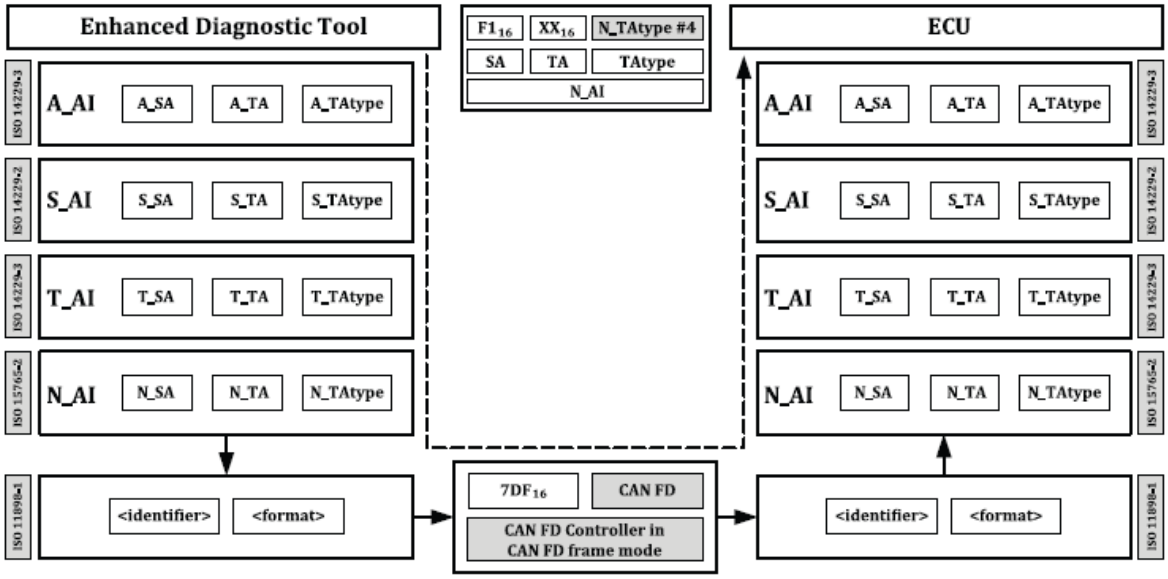


Figure 6 — Example of enhanced diagnostic tool CAN FD request for normal addressing (N_TAtype #4)

8.3.2.5 N_AE, address externsion

8 bits, only part of the addressing information if Mtype is set to remote diagnostics.

8.3.3 <Length>

32bits

8.3.4 <MessageData>

string of bytes

8.3.5 <Parameter>

enumeration, Range: STmin, BS

8.3.6 <Parameter_Value>

8bits

8.3.7 <N_Result>

Type	Cause	Issue to	Issue by
N_OK		service user	sender + receiver
N_TIMEOUT_A	timer N_Ar/N_As exceed the N_Asmax/N_Armax	protocol user	sender + receiver
N_TIMEOUT_Bs	timer N_Bs exceed N_Bsmax	service user	sender
N_TIMEOUT_Cr	timer N_Cr exceed N_Crmax	service user	receiver
N_WRONG_SN	receipt an unexpected SequenceNumber(PCI.SN)	service user	receiver
N_INVALID_FS	received an invalid or unkown Flowstatus value	service user	sender
N_UNEXP_PDU	receipt an unexpected protocol data unit	service user	receiver
N_WFT_OVRN	receiver has transmitted N_WFTmax FlowControl N_PDUs with FlowStatus= WAIT in a row and following this	service user	receiver
N_BUFFER_OVFLW	receipt a FlowControl (FC) with FlowStatus = OVFLW.	service user	sender
N_ERROR		service user	sender + receiver

8.3.8 <Result_ChangeParameter>

Type	Cause	Issue to	Issue by
N_OK		service user	sender + receiver
N_RX_ON	reception of the message identified by <N_AI> was taking place	service user	receiver
N_WRONG_PARAMETER	an undefined	service user	sender + receiver
N_WRONG_VALUE	an out of range <Parameter_Value>	service user	sender + receiver

9 Transport layer protocol

9.1 Protocol functions

- transmission/reception of message up to 2^{32} - 1 data bytes
- reporting of transmission/reception completion (or failure)

9.2 SingleFrame transmission

9.2.1 SF with TX_DL=8

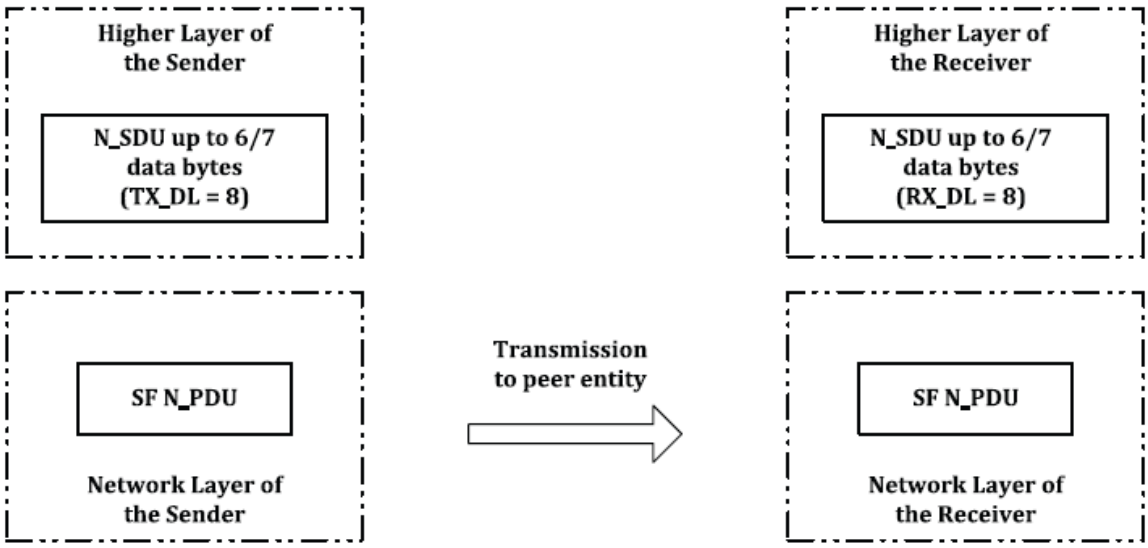


Figure 7 — Example of a SingleFrame (SF) transmission (TX_DL = 8)

- six (TX_DL - 2) : extended or mixed addressing
- seven (TX_DL - 1) : normal addressing

9.2.2 SF with TX_DL > 8

- TX_DL - 3 : extended or mixed addressing
- TX_DL - 2 : normal addressing

9.3 Multiple-frame transmission

FlowControl mechanism as follows

- BlockSize (BS) : The maximum number of N_PDUs the receiver allows the sender to send before waiting for an authorization to continue transmission of the following N_PDUs. When BS is ZERO, not waiting to continuce transmission
- SeparationTime minmum (STmin) : The minimum time the sender is to wait between transmission of two CF N_PDUs.

two different modes for the adoption of above values for the receiver of a segmented message:

- dynamic : BS and STmin are updated for the subsequent PDU
- static : constant BS and STmin

ALL blocks , except the last one , will consist of BS N_PDUs. the last one will contain the remaining N_PDUs (from 1 up to BS)

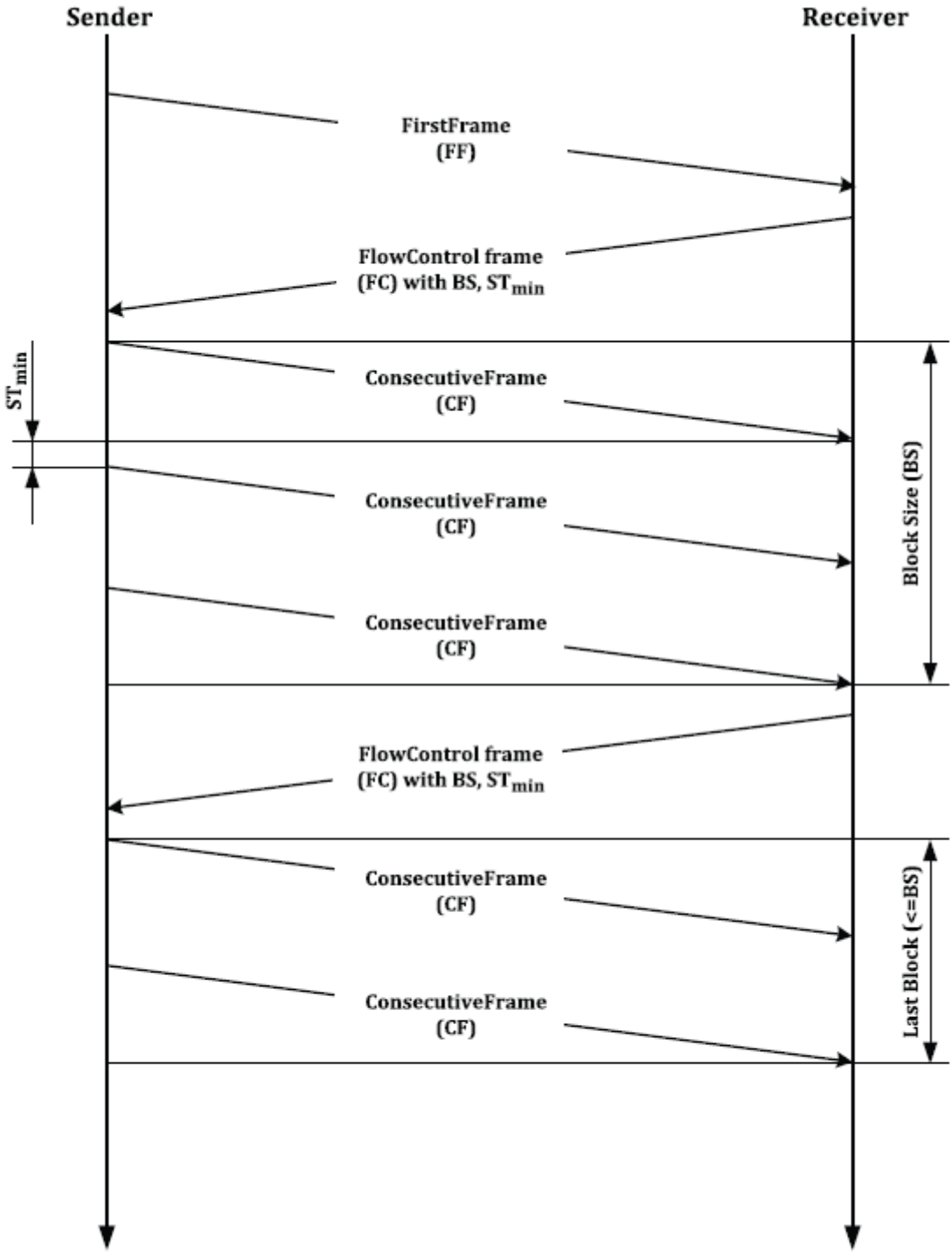


Figure 8 — FlowControl (FC) mechanism

9.4 Transport layer protocol data units

9.4.1 Protocol data unit types

four different types of transport layer protocol data units,

- SingleFrame (SF N_PDU)
- FirstFrame (FF N_PDU)
- ConsecutiveFrame (CF N_PDU)
- FlowControl (FC N_PDU)

9.4.2 SF N_PDU

sent out by the sending network entity and can be received by one or multiple receiving network entities.

9.4.3 FF N_PDU

sent out by the sending network entity and received by a unique network entity for the duration of the segmented message transmission.

9.4.4 CF N_PDU

sent out by the sending network entity and received by a unique network entity for the duration of the segmented message transmission.

pass the assembled message to the service user of the network receiving entity **after** *the last* CF N_PDU has been received.

9.4.5 FC N_PDU

sent by the receiving network layer entity to the sending network layer entity, when ready to receive more data, after correct reception of

- an FF N_PDU or
- the last CF N_PDU of a block of CF, if further CF need to be sent

except to start, stop and resume transmission of CF N_PDUs, the FC can also inform to pause transmission of CF during a segmented message transmission or to abort the transmission of a segmented message if the length information (FF_DL) exceeds the buffer size of the receiving entity.

9.4.6 Protocol data unit field description

9.4.6.1 N_PDU format

Address information	Protocol control information	Data field
N_AI	N_PCI	N_Data

9.4.6.2 N_AI

the N_AI received in the N_SDU (N_SA, N_TA, N_TAtype [and N_AE]) should be copied and include in the N_PDU. if the message data (<MessageData> and <Length>) received in the N_SDU requires segmentation for the network layer to transmit the complete message, the N_AI shall be copied and included (repeated) in every N_PDU that is transmitted.

- N_SA, network source address, 8 bits
- N_TA, network target address, 8 bits
- N_TAtype, network target address type, enumeration
- N_AE, network address extension, 8 bits, only for Mtype is set to remote diagnostics
- <Length>, 32bits
- <MessageData>,string of bytes
- <parameter>, enumeration, range: STmin ~ BS
- <parameter_Value>, 8 bits

- <N_Result>, enumeration, error priority:

9.4.6.3 N_PCI

refer to 9.6.1

9.4.6.4 N_Data

the size of N_Data depends on the N_PDU type, the address format chosen, and the value of TX_DL.

9.5 TX_DL configuration

9.5.1 definition of TX_DL

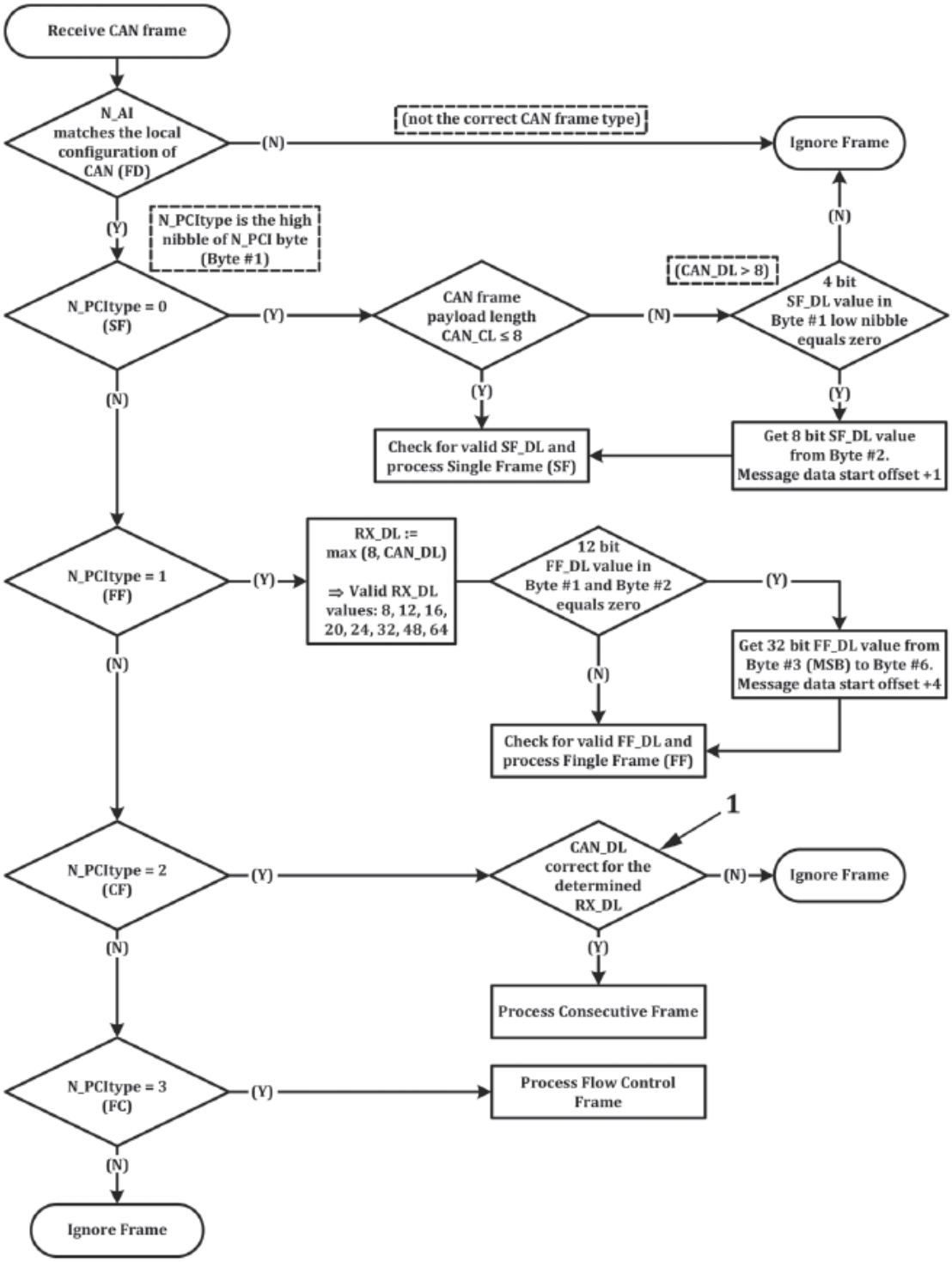
Table 6 — Definition of TX_DL configuration values

TX_DL	Description
<8	**Invalid** This range of values is invalid.
=8	**Configured CAN frame maximum payload length of 8 bytes** For the use with ISO 11898-1 CLASSICAL CAN type frames and CAN FD type frames: — Valid DLC value range: 2..8; — Valid CAN_DL value range: 2..8.
>8	**Configured CAN frame maximum payload length greater than 8 bytes** For the use with ISO 11898-1 CAN FD type frames only: — Valid DLC value range: 2..15; — Valid CAN_DL value range: 2..8, 12, 16, 20 , 24, 32, 48, 64; — Valid TX_DL value range: 12, 16, 20 , 24, 32, 48, 64; — CAN_DL ≤ TX_DL.

9.5.2 Creating CAN frames based on N_TAtype and TX_DL

CAN frames are generated based upon **N_AI**, the configured **addressing format** for the given N_AI, the configured **TX_DL** value, and the size of the message to be transmitted.

9.5.3 Verifying the correctness of received CAN Frames



Key

1 CAN_DL shall be correct if the value matches RX_DL for all CF's except for the last (or only) CF; the last (or only) CF shall pass this check if CAN_DL is less or equal than RX_DL and the requirements in 9.6.4.2 are met; RX_DL comes from the FF and is fixed for this PDU reception process

Figure 9 — State flow — Verifying received CAN frames

9.5.4 Receiver determination RX_DL

Table 7 — Received CAN_DL to RX_DL mapping table

Received CAN_DL	RX_DL
0 to 7	invalid
8	8
12	12
16	16
20	20
24	24
32	32
48	48
64	64

9.6 Protocol control information specification

9.6.1 N_PCI

Table 9 — Summary of N_PCI bytes

N_PDU name	N_PCI bytes						
	Byte #1		Byte #2	Byte #3	Byte #4	Byte #5	Byte #6
	Bits 7 – 4	Bits 3 – 0					
SingleFrame (SF) (CAN_DL ≤ 8)	0000_2	SF_DL	—	—	—	—	—
SingleFrame (SF) (CAN_DL > 8)[a]	0000_2	0000_2	SF_DL	—	—	—	—
FirstFrame (FF) (FF_DL ≤ 4 095)	0001_2	FF_DL		—	—	—	—
FirstFrame (FF) (FF_DL > 4 095)[b]	0001_2	0000_2	$0000\ 0000_2$	FF_DL			
ConsecutiveFrame (CF)	0010_2	SN	—	—	—	—	—
FlowControl (FC)	0011_2	FS	BS	ST_{min}	N/A	N/A	N/A

[a] Messages with CAN_DL > 8 shall use an escape sequence where the lower nibble of Byte #1 is set to 0 (invalid length). This signifies to the network layer that the value of SF_DL is determined based on the next byte in the frame (Byte #2). As CAN_DL is defined to be greater than 8, this definition is only valid for CAN FD type frames.

[b] Messages larger than 4 095 bytes shall use an escape sequence where the lower nibble of Byte #1 and all bits in Byte #2 are set to 0 (invalid length). This signifies to the network layer that the value of FF_DL is determined based on the next 32 bits in the frame (Byte #3 is the MSB and Byte #6 the LSB).

9.6.2 SF N_PCI parameter definition

9.6.2.1 SF N_PCI byte

SF_DL is used to specify the number of service message data bytes. The ranges of valid SF_DL values depend on the configured transmit data link layer data length (**TX_DL**) and the **actual payload** to be transmitted.

Table 10 — Definition of SF_DL values with CAN_DL ≤ 8

Value Bits 7 – 4	Description
0000_2	**Reserved** This value is reserved by this part of ISO 15765.
0001_2 .. 0110_2	**SingleFrame DataLength (SF_DL)** SF_DL shall be assigned the value of the service parameter <Length>.
0111_2	**SingleFrame DataLength (SF_DL) with normal addressing only** SF_DL shall be assigned the value of the service parameter <Length>. SF_DL = 7 is only allowed with normal addressing.
other values	**Invalid** This range of values is invalid.

NOTE 1 SF_DL is encoded in the low nibble of first N_PCI byte (Byte #1) value.

Table 11 — Definition of SF_DL values (CAN_DL > 8)

Value	Description
$0000\ 0000_2$ .. $0000\ 0110_2$	**Reserved** This value is reserved by this part of ISO 15765.
$0000\ 0111_2$	**SingleFrame DataLength (SF_DL) with extended addressing or mixed addressing** SF_DL shall be assigned the value of the service parameter <Length>. SF_DL = 7 is only allowed with extended addressing or mixed addressing.
$0000\ 1000_2$ .. (CAN_DL – 3)	**SingleFrame DataLength (SF_DL)** SF_DL shall be assigned the value of the service parameter <Length>.
(CAN_DL – 2)	**SingleFrame DataLength (SF_DL) with normal addressing only** SF_DL shall be assigned the value of the service parameter <Length>. SF_DL = (CAN_DL – 2) is only allowed with normal addressing.
other values	**Invalid** This range of values is invalid.

NOTE 2 SF_DL is encoded in the second N_PCI byte (Byte #2) value. This is only allowed for CAN FD type frames.

9.6.2.2 SF_DL error handling

- Received CAN_DL is less or equal to 8

Table 12 — Allowed SF_DL values for a given addressing scheme with optimized CAN_DL

Addressing type	CAN_DL value							
	0 .. 1	2	3	4	5	6	7	8
Normal	Invalid	SF_DL = 1	SF_DL = 2	SF_DL = 3	SF_DL = 4	SF_DL = 5	SF_DL = 6	SF_DL = 7
Mixed or extended	Invalid	Invalid	SF_DL = 1	SF_DL = 2	SF_DL = 3	SF_DL = 4	SF_DL = 5	SF_DL = 6

- Received CAN_DL is greater than 8

Table 13 — Allowed SF_DL values for a given CAN_DL greater than 8 and addressing scheme

Addressing type	CAN_DL value						
	12	16	20	24	32	48	64
Normal	8 ≤ SF_DL ≤ 10	11 ≤ SF_DL ≤ 14	15 ≤ SF_DL ≤ 18	19 ≤ SF_DL ≤ 22	23 ≤ SF_DL ≤ 30	31 ≤ SF_DL ≤ 46	47 ≤ SF_DL ≤ 62
Mixed or extended	7 ≤ SF_DL ≤ 9	10 ≤ SF_DL ≤ 13	14 ≤ SF_DL ≤ 17	18 ≤ SF_DL ≤ 21	22 ≤ SF_DL ≤ 29	30 ≤ SF_DL ≤ 45	46 ≤ SF_DL ≤ 61

9.6.3 FF N_PCI parameter definition

9.6.3.1 FF_DL definition

for the sender, the range of valid **FF_DL** values depends on the addressing scheme and the configured transmit data link layer data length (**TX_DL**). The minimum values of FF_DL (**FF_DLmin**) based on addressing scheme and TX_DL are specified in below Table

Table 14 — Minimum value of FF_DL based on the addressing scheme

Condition	FF_DL_{min} value
If the configured TX_DL is 8 and normal addressing is used.	8
If the configured TX_DL is 8 and mixed or extended addressing is used.	7
If the configured TX_DL > 8 and normal addressing is used.	TX_DL – 1
If the configured TX_DL > 8 and mixed or extended addressing is used.	TX_DL – 2

Table 15 — Valid FF_DL values

Value	Description
0 ... (FF_DL_{min} – 1)	**Invalid** This range of values is invalid.
FF_DL_{min} ... 4 095	**FirstFrame DataLength (FF_DL) without escape sequence** The encoding of the segmented message length results in a twelve bit length value (FF_DL) where the least significant bit (LSB) is specified to be bit 0 of the second N_PCI byte (Byte #2) and the most significant bit (MSB) is bit 3 of the first N_PCI byte (Byte #1). The maximum segmented message length supported is equal to 4 095 bytes of user data. It shall be assigned the value of the service parameter <Length>.
4 096 .. 4 294 967 295 (2^{32}-1)	**FirstFrame DataLength (FF_DL) with escape sequence** The encoding of the segmented message length results in a 32 bit length value (FF_DL) where the least significant bit (LSB) is specified to be bit 0 of the sixth N_PCI byte (Byte #6) and the most significant bit (MSB) is bit 7 of the third N_PCI byte (Byte #3). The maximum segmented message length supported is equal to 4 294 967 295 bytes of user data. It shall be assigned the value of the service parameter <Length>.

9.6.3.2 FF_DL error handling

- FF and CAN_DL < 8, ignore the FF N_PDU
- FF_DL greater than the available receiver buffer size, abort and send FC N_PDU with the parameter FlowStatus = Overflow
- FF_DL less than FF_DLmin, ignore
- FF with the escape sequence and the FF_DL =< 4095, ignore

9.6.4 CF N_PCI parameter definition

9.6.4.1 CF N_PCI byte

the payload data length CAN_DL of the received CAN frame has to match the RX_DL value which was determined in the reception process of the FirstFrame. Only the last CF in the multi-frame transmission may contain less than RX_DL bytes.

9.6.4.2 Transmitter requirements for last consecutive frame

9.6.4.3 SequenceNumber (SN) parameter definition

- the number ascending order of the ConsecutiveFrames
- the SN shall start with zero for all segmented message, include FF
- the SN of the first CF immediately following the FF shall be set to one

- the SN shall be incremented by one for each new CF
- the SN value shall not be affected by any FlowControl (FC) frame
- when the SN reaches the value of 15, it shall wrapparound and be set to zero for the next CF.

Table 16 — Summary of SN definition

N_PDU	FF	CF	CF	CF	CF	CF	CF	CF
SN	0_{16}	1_{16}	...	E_{16}	F_{16}	0_{16}	1_{16}	...

9.6.4.4 SN error handling

when violated the above SN rule, abort the receipt message and the network layer shall make an N_USData.indication service call with the parameter <N_Result> = N_WRONG_SN to the adjacent upper layer.

9.6.5 FlowControl N_PCI parameter definition

9.6.5.1 FlowStatus (FS) parameter definition

A **sending** network entity shall support all specified (not reserved)values of the FS parameter.

Table 18 — Definition of FS values

Value	Description
0_{16}	**ContinueToSend (CTS)** The FlowControl ContinueToSend parameter shall be encoded by setting the lower nibble of the N_PCI byte #1 to "0". It shall cause the sender to resume the sending of ConsecutiveFrames. The meaning of this value is that the receiver is ready to receive a maximum of BS number of ConsecutiveFrames.
1_{16}	**Wait (WAIT)** The FlowControl Wait parameter shall be encoded by setting the lower nibble of the N_PCI byte #1 to "1". It shall cause the sender to continue to wait for a new FlowControl N_PDU and to restart its N_BS timer. If FlowStatus is set to Wait, the values of BS (BlockSize) and ST_{min} (SeparationTime minimum) in the FlowControl message are not relevant and shall be ignored.
2_{16}	**Overflow (OVFLW)** The FlowControl Overflow parameter shall be encoded by setting the lower nibble of the N_PCI byte #1 to "2". It shall cause the sender to abort the transmission of a segmented message and make an N_USData.confirm service call with the parameter <N_Result> = N_BUFFER_OVFLW. This N_PCI FlowStatus parameter value is only allowed to be transmitted in the FlowControl N_PDU that follows the FirstFrame N_PDU and shall only be used if the message length FF_DL of the received FirstFrame N_PDU exceeds the buffer size of the receiving entity. If FlowStatus is set to Overflow, the values of BS (BlockSize) and ST_{min} (SeparationTime minimum) in the FlowControl message are not relevant and shall be ignored.
3_{16} – F_{16}	**Reserved** This range of values is reserved by this part of ISO 15765.

9.6.5.2 FlowStatus (FS) error handling

aborted and the network layer shall make an N_USData.confirm service call with the parameter <N_Result>= N_INVALID_FS to the adjacent upper layer.

9.6.5.3 BlockSize (BS) parameter definition

the units of BS are the absolute number of CF N_PDUs per block.

eg, If BS is 20, then the block will consist of 20 CF N_PDUs.

Only the last block of CF in a segmented data transmission may have less than the BS number of frames.

Table 19 — Definition of BS values

Value	Description
00_{16}	**BlockSize (BS)** The BS parameter value 0 shall be used to indicate to the sender that no more FC frames shall be sent during the transmission of the segmented message. The sending network layer entity shall send all remaining ConsecutiveFrames without any stop for further FC frames from the receiving network layer entity.
01_{16} – FF_{16}	**BlockSize (BS)** This range of BS parameter values shall be used to indicate to the sender the maximum number of ConsecutiveFrames that can be received without an intermediate FC frame from the receiving network entity.

9.6.5.4 SeparationTime minimum (STmin) parameter definition

specified by the receiving entity, which is the minimum time gap allowed between the transmissions of two CFs.

Table 20 — Definition of ST_{min} values

Value	Description
00_{16} – $7F_{16}$	**SeparationTime minimum (ST_{min}) range:** 0 ms – 127 ms The units of ST_{min} in the range 00_{16} – $7F_{16}$ (0 – 127) are absolute milliseconds (ms).
80_{16} – $F0_{16}$	**Reserved** This range of values is reserved by this part of ISO 15765.
$F1_{16}$ – $F9_{16}$	**SeparationTime minimum (ST_{min}) range:** 100 µs – 900 µs The units of ST_{min} in the range $F1_{16}$ – $F9_{16}$ are even multiples of 100 µs, where parameter value $F1_{16}$ represents 100 µs and parameter value $F9_{16}$ represents 900 µs.
FA_{16} – FF_{16}	**Reserved** This range of values is reserved by this part of ISO 15765.

The measurement of the STmin starts after completion of transmission of a CF and ends at the request for the transmission of the next CF.

9.6.5.5 STmin error handling

if got a reserved STmin value, the sending network entity shall use the longest STmin value (0x7F = 127ms).

if the time between two subsequent CFs of a segmented data transmission (N_As + N_Cs) is smaller than the value commanded by the receiver via STmin, there is no guarantee that the receiver of the segmented data transmission will correctly receive and process all frames.

In any case, the receiver of the segmented data transmission is not required to monitor adherence to the STmin value.

9.6.5.6 Dynamic BS/STmin values in subsequent FlowControl frames

9.7 Maximum number of FC.WAIT frame transmissions (N_WFTmax)

avoid sender nodes within while_1_loop. This parameter is local to communication peers and is not transmitted and is hence not part of the FC protocol data unit.

- The N_WFTmax shall indicate how many FC N_PDU WAITs can be transmitted by the receiver in a row
- the upper limit shall be user defined at system generation time
- only be used on the receiving network entity during message reception

- if set to zero, then FlowControl shall reply upon FlowControl continue to send FC N_PDU CTS only.

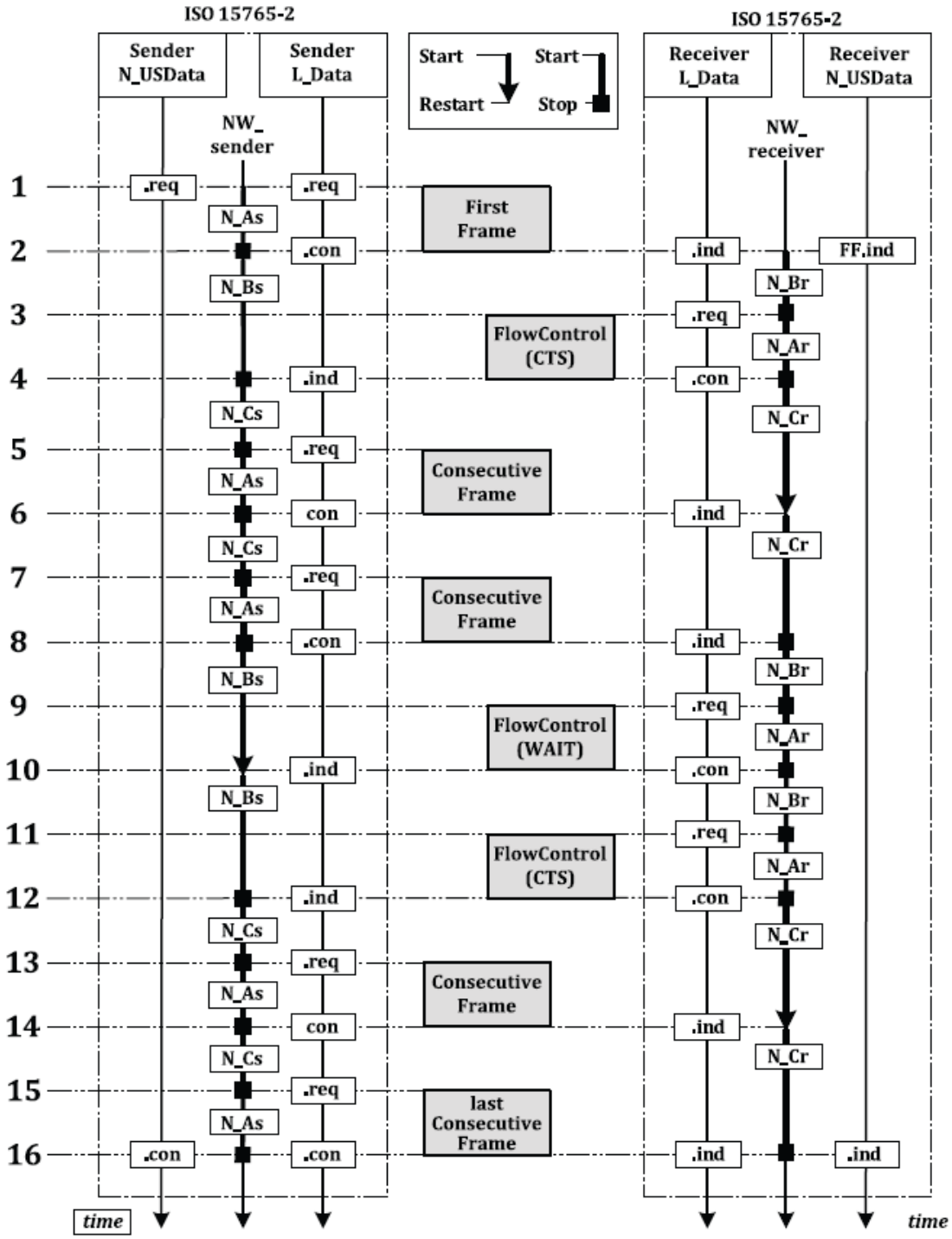
9.8 Network layer timing

9.8.1 Timing parameters

Table 21 — Network layer timing parameter values

Timing parameter	Description	Data link layer service		Time-out	Performance requirement
		Start	End	ms	ms
N_As	Time for transmission of the CAN frame (any N_PDU) on the sender side	L_Data.request	L_Data.confirm	1 000	—
N_Ar	Time for transmission of the CAN frame (any N_PDU) on the receiver side	L_Data.request	L_Data.confirm	1 000	—
N_Bs	Time until reception of the next FlowControl N_PDU	L_Data.confirm (FF) L_Data.confirm (CF) L_Data.indication (FC)	L_Data.indication (FC)	1 000	—
N_Br	Time until transmission of the next FlowControl N_PDU	L_Data.indication (FF) L_Data.indication (CF) L_Data.confirm (FC)	L_Data.request (FC)	N/A	(N_Br + N_Ar) < (0,9 × N_Bs timeout)
N_Cs	Time until transmission of the next ConsecutiveFrame N_PDU	L_Data.indication (FC) L_Data.confirm (CF)	L_Data.request (CF)	N/A	(N_Cs + N_As) < (0,9 × N_Cr timeout)
N_Cr	Time until reception of the next Consecutive Frame N_PDU	L_Data.confirm (FC) L_Data.indication (CF)	L_Data.indication (CF)	1 000	—

- .req -> request
- .con -> confirm
- .ind -> indication



9.8.2 Network layer timeouts

Table 22 — Network layer timeout error handling

Error	Cause	Action
N_As	Any N_PDU not transmitted in time on the sender side	Abort message transmission and issue N_USData.confirm with <N_Result> = N_TIMEOUT_A
N_Ar	Any N_PDU not transmitted in time on the receiver side	Abort message reception and issue N_USData.indication with <N_Result> = N_TIMEOUT_A
N_Bs	FlowControl N_PDU not received (lost, overwritten) on the sender side or preceding FirstFrame N_PDU or ConsecutiveFrame N_PDU not received (lost, overwritten) on the receiver side	Abort message transmission and issue N_USData.confirm with <N_Result> = N_TIMEOUT_Bs
N_Cr	ConsecutiveFrame N_PDU not received (lost, overwritten) on the receiver side or preceding FC N_PDU not received (lost, overwritten) on the sender side	Abort message reception and issue N_USData.indication with <N_Result> = N_TIMEOUT_Cr

Error	Cause	Action	Comments
N_As			report to upper layer with confirm
			TODO
			??

9.8.3 Unexpected arrival of N_DPU

- out of the normal expected order
- it could be an unkown N_PDU that cannot be interpreted by the definitions given in the ISO 15765

9.8.4 Wait frame error handling

- *Receiver*: has tranmitted N_WFTmax in a row. the receiver side shall abort the message reception and issue an N_USData.indication with <N_Result> set to N_WFT_OVRN to the higher layer.
- *Sender*: informed about the aborted message reception via an N_USData.confirm with <N_Result> set to N_TIMEOUT_Bs. (Because of the missing FC from the receiver, an N_Bs timeout occurs in the sender.) TODO

9.9 Interleaving of message

use case of gateway need this feature.

10 Data link layer usage

10.1 Data link layer service parameters

defined in ISO11898-1 :

- <Data> : CAN frame data
- <DLC> : data length code
- <Identifier> : CAN identifier
- <Transfer_Status> : status of a transmission
- <Format> : frame format (CAN, CAN_FD, base: 11-bit, extend: 29-bit)

10.2 Data Link layer interface services

10.2.1 L_Data.request

L_Data.request (

<Identifier>

<Format>

<DLC>

<Data>

)

10.2.2 L_Data.confirm

L_Data.confirm(

<Identifier>

<Transfer_Status>

)

10.2.3 L_Data.indication

L_Data.indication(

<Identifier>

<Format>

<DLC>

<Data>

)

10.3 Mapping of the N_PDU fields

10.3.1 Addressing formats

three addressing formats

- normal
- extended
- mixed addressing

10.3.2 Normal addressing

Base on N_TAtype, Table 9

Table 24 — Mapping of N_PDU parameters into CAN frame — Normal addressing, N_TAtype = #1, #3, #5 and #7

N_PDU type	CAN identifier	CAN frame data field Byte 1 - n[a]
SingleFrame (SF)	N_AI	N_PCI, N_Data
FirstFrame (FF)	N_AI	N_PCI, N_Data
ConsecutiveFrame (CF)	N_AI	N_PCI, N_Data
FlowControl (FC)	N_AI	N_PCI

[a] See Table 3 and Table 9.

Table 25 — Mapping of N_PDU parameters into CAN frame — Normal addressing, N_TAtype = #2, #4, #6 and #8

N_PDU type	CAN identifier	CAN frame data field Byte 1 - n[a]
SingleFrame (SF)	N_AI	N_PCI, N_Data

[a] See Table 3 and Table 9.

10.3.3 Normal fixed addressing

only 29 bit CAN identifiers are allowed. Base on N_TAtype, Table 9

Table 26 — Normal fixed addressing, N_TAtype = #5 and 7

N_PDU type	29 bit CAN identifier bit position						CAN frame data field byte position
	28 – 26	25	24	23 – 16	15 – 8	7 – 0	Byte 1 - n[a]
SingleFrame (SF)	110_2	0	0	218	N_TA	N_SA	N_PCI, N_Data
FirstFrame (FF)	110_2	0	0	218	N_TA	N_SA	N_PCI, N_Data
ConsecutiveFrame (CF)	110_2	0	0	218	N_TA	N_SA	N_PCI, N_Data
FlowControl (FC)	110_2	0	0	218	N_TA	N_SA	N_PCI

[a] See Table 3 and Table 9.

Table 27 — Normal fixed addressing, N_TAtype = #6 and 8

N_PDU type	29 bit CAN identifier bit position						CAN frame data field byte position
	28 – 26	25	24	23 – 16	15 – 8	7 – 0	Byte 1 - n[a]
SingleFrame (SF)	110_2	0	0	219	N_TA	N_SA	N_PCI, N_Data

[a] See Table 3 and Table 9.

10.3.4 Extended addressing

Base on N_TAtype, Table 9

Table 28 — Mapping of N_PDU parameters into CAN frame — Extended addressing, N_TAtype = #1, #3, #5 and 7 可用于网关转发TP报文。

N_PDU type	CAN identifier	Byte 1	Byte 2 - n[a]
SingleFrame (SF)	N_AI, except N_TA	N_TA	N_PCI, N_Data
FirstFrame (FF)	N_AI, except N_TA	N_TA	N_PCI, N_Data
ConsecutiveFrame (CF)	N_AI, except N_TA	N_TA	N_PCI, N_Data
FlowControl (FC)	N_AI, except N_TA	N_TA	N_PCI

[a] See Table 3 and Table 9.

Table 29 — Mapping of N_PDU parameters into CAN frame — Extended addressing, N_TAtype = #2, #4, #6 and 8

N_PDU type	CAN identifier	Byte 1	Byte 2 - n[a]
SingleFrame (SF)	N_AI, except N_TA	N_TA	N_PCI, N_Data

[a] See Table 3 and Table 9.

10.3.5 Mixed addressing

10.3.5.1 29 bit CAN identifier

Mixed addressing is the addressing format to be used if *Mtype* is set to **remote diagnostics**

Base on N_TAtype,

Table 30 — Mixed addressing with 29 bit CAN identifier, N_TAtype = #5 and 7

N_PDU type	29 bit CAN identifier bit position						CAN frame data field byte position	
	28 – 26	25	24	23 – 16	15 – 8	7 – 0	1	Byte 2 - n[a]
SingleFrame (SF)	110_2	0	0	206	N_TA	N_SA	N_AE	N_PCI, N_Data
FirstFrame (FF)	110_2	0	0	206	N_TA	N_SA	N_AE	N_PCI, N_Data
ConsecutiveFrame (CF)	110_2	0	0	206	N_TA	N_SA	N_AE	N_PCI, N_Data
FlowControl (FC)	110_2	0	0	206	N_TA	N_SA	N_AE	N_PCI
[a] See Table 3 and Table 9.								

Table 31 — Mixed addressing with 29 bit CAN identifier, N_TAtype = #6 and 8

N_PDU type	29 bit CAN identifier bit position						CAN frame data field byte position	
	28 – 26	25	24	23 – 16	15 – 8	7 – 0	1	Byte 2 - n[a]
SingleFrame (SF)	110_2	0	0	205	N_TA	N_SA	N_AE	N_PCI, N_Data
[a] See Table 3 and Table 9.								

10.3.5.2 11 bit CAN identifier

Table 32 — Mixed addressing with 11 bit CAN identifier, N_TAtype = #1 and 3

N_PDU type	CAN identifier	CAN frame data field	
		Byte 1	Byte 2 - n[a]
SingleFrame (SF)	N_AI	N_AE	N_PCI, N_Data
FirstFrame (FF)	N_AI	N_AE	N_PCI, N_Data
ConsecutiveFrame (CF)	N_AI	N_AE	N_PCI, N_Data
FlowControl (FC)	N_AI	N_AE	N_PCI
[a] See Table 3 and Table 9.			

Table 33 — Mixed addressing with 11 bit CAN identifier, N_TAtype = #2 and 4

N_PDU type	CAN identifier	CAN frame data field	
		Byte 1	Byte 2 - n[a]
SingleFrame (SF)	N_AI	N_AE	N_PCI, N_Data
[a] See Table 3 and Table 9.			

10.4 CAN frame data length code (DLC)

10.4.1 DLC parameter

10.4.2 CAN frame data

10.4.2.1 CAN frame data padding (TX_DL = 8)

DLC is always set to 8. this can be the case for an SF, FC frame or the last CF of a segmented message. The default value 0xCC should be used for frame padding.

DLC parameter cannot be used to determine the message length; this information shall be extracted from the N_PCI information at the beginning of a message.

Table 34 — Data padding example (TX_DL = 8), normal addressing, N_PDU size 6 byte, DLC = 8

N_PDU type	CAN Identifier	CAN frame data field							
		Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
SingleFrame (SF)	N_AI	N_PCI	N_Data					padding	
	345_{16}	05_{16}	44_{16}	55_{16}	66_{16}	77_{16}	88_{16}	CC_{16}	CC_{16}

10.4.2.2 CAN frame data optimization (TX_DL = 8)

DLC is **not** always need to be 8. this can **only** be the case for an SF, FC frame or the last CF of a segmented message.

Table 35 — Data optimized example (TX_DL = 8), normal addressing, N_PDU size 6 byte, DLC = 6

N_PDU type	CAN Identifier	CAN frame data field					
		Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
SingleFrame (SF)	N_AI	N_PCI	N_Data				
	345_{16}	05_{16}	44_{16}	55_{16}	66_{16}	77_{16}	88_{16}

10.4.2.3 Mandatory padding of CAN FD frames (TX_DL > 8)

Table 36 — Data padding example (TX_DL > 8), normal addressing, N_PDU size 11 bytes, DLC = 9

N_PDU type	CAN Identifier	CAN frame data field										
		Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12
Single-Frame (SF)	N_AI	N_PCI		N_Data									Manda-tory padding
	345_{16}	00_{16}	09_{16}	11_{16}	22_{16}	33_{16}	44_{16}	55_{16}	66_{16}	77_{16}	88_{16}	99_{16}	CC_{16}

10.4.3 DLC error handling

ignore

For details, see specific error handling in 9.6.