INTERNATIONAL STANDARD

ISO 15765-2

Second edition
2011-11-15

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

Part 2: Transport protocol and network layer services

Véhicules routiers — Communication de diagnostic sur gestionnaire de réseau de communication (DoCAN) —

Partie 2: Protocole de transport et services de la couche réseau

Reference number
ISO 15765-2:2011(E)

Contents

Page

Foreword

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

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

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

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

ISO 15765-2 was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 3, *Electrical and electronic equipment*.

This second edition cancels and replaces the first edition (ISO 15765-2:2004), which has been technically revised.

ISO 15765 consists of the following parts, under the general title *Road vehicles — Diagnostic communication over Controller Area Network (DoCAN)*:

— *Part 1: General information and use case definition*

— *Part 2: Transport protocol and network layer services*

— *Part 3: Implementation of unified diagnostic services (UDS on CAN)*

— *Part 4: Requirements for emissions-related systems*

Introduction

This part of ISO 15765 has been established in order to define common requirements for vehicle diagnostic systems implemented on a controller area network (CAN) communication link, as specified in ISO 11898. Although primarily intended for diagnostic systems, it also meets requirements from other CAN-based systems needing a network layer protocol.

To achieve this, it is based on the Open Systems Interconnection (OSI) Basic Reference Model in accordance with ISO/IEC 7498-1 and ISO/IEC 10731, which structures communication systems into seven layers as shown in Table 1.

Table 1 — Enhanced and legislated on-board diagnostics specifications applicable to the OSI layers

<table>
<tr><th>Applicability</th><th>OSI 7 layers</th><th>Vehicle-manufacturer-enhanced diagnostics</th><th colspan="2">Legislated OBD (on-board diagnostics)</th><th colspan="2">Legislated WWH-OBD (on-board diagnostics)</th></tr>
<tr><td rowspan="7">Seven layers according to ISO/IEC 7498-1 and ISO/IEC 10731</td><td>Application (layer 7)</td><td>ISO 14229-1,
ISO 14229-3</td><td colspan="2">ISO 15031-5</td><td colspan="2">ISO 27145-3, ISO 14229-1</td></tr>
<tr><td>Presentation (layer 6)</td><td>Vehicle manufacturer specific</td><td colspan="2">ISO 15031-2,
ISO 15031-5,
ISO 15031-6,
SAE J1930-DA,
SAE J1979-DA,
SAE J2012-DA</td><td colspan="2">ISO 27145-2, SAE 1930-DA,
SAE J1979-DA, SAE J2012-DA,
SAE J1939:2011,
Appendix C (SPN),
SAE J1939-73:2010,
Appendix A (FMI)</td></tr>
<tr><td>Session (layer 5)</td><td colspan="5">ISO 14229-2</td></tr>
<tr><td>Transport protocol (layer 4)</td><td rowspan="2">ISO 15765-2</td><td rowspan="2">ISO 15765-2</td><td rowspan="4">ISO 15765-4</td><td rowspan="2">ISO 15765-4,
ISO 15765-2</td><td rowspan="4">ISO 27145-4</td></tr>
<tr><td>Network (layer 3)</td></tr>
<tr><td>Data link (layer 2)</td><td rowspan="2">ISO 11898-1
ISO 11898-2
ISO 11898-3
ISO 11898-5
or
user defined</td><td rowspan="2">ISO 11898-1,
ISO 11898-2</td><td rowspan="2">ISO 15765-4,
ISO 11898-1,
ISO 11898-2</td></tr>
<tr><td>Physical (layer 1)</td></tr>
</table>

The application layer services covered by ISO 14229-3 have been defined in compliance with diagnostic services established in ISO 14229-1 and ISO 15031-5, but are not limited to use only with them. ISO 14229-3 is also compatible with most diagnostic services defined in national standards or vehicle manufacturer's specifications.

The transport protocol and network layer services covered by this part of ISO 15765 have been defined to be independent of the physical layer implemented, and a physical layer is only specified for legislated OBD.

For other application areas, ISO 15765 can be used with any CAN physical layer.

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

Part 2: Transport protocol and network layer services

1 Scope

This part of ISO 15765 specifies a transport protocol and network layer services tailored to meet the requirements of CAN-based vehicle network systems on controller area networks as specified in ISO 11898. It has been defined in accordance with the diagnostic services established in ISO 14229-1 and ISO 15031-5, but is not limited to use with them and is also compatible with most other communication needs for in-vehicle networks. The protocol specifies an unconfirmed communication.

The diagnostic communication over controller area network (DoCAN) protocol supports the standardized service primitive interface as specified in ISO 14229-2.

This part of ISO 15765 provides the transport protocol and network layer services to support different application-layer implementations such as

— enhanced vehicle diagnostics (emissions-related system diagnostics beyond legislated functionality, non-emissions-related system diagnostics),

— emissions-related on-board diagnostics (OBD) as specified in ISO 15031, and

— world-wide harmonized on-board diagnostics (WWH-OBD) as specified in ISO 27145.

2 Normative references

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

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

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

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 7498-1 apply.

3.2 Abbreviated terms

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

BS	BlockSize
CAN	controller area network
CF	ConsecutiveFrame

confirm	confirmation service primitive
CTS	continue to send
DL	DataLength
DoCAN	diagnostic communication over CAN
ECU	electronic control unit
FC	FlowControl
FF	FirstFrame
FF_DL	FirstFrame data length
FMI	failure mode indicator
FS	FlowStatus
indication	indication service primitive
Mtype	message type
N_AE	network address extension
N_AI	network address information
N_Ar	network layer timing parameter Ar
N_As	network layer timing parameter As
N_Br	network layer timing parameter Br
N_Bs	network layer timing parameter Bs
N_ChangeParameter	network layer service name
N_Cr	network layer timing parameter Cr
N_Cs	network layer timing parameter Cs
N_Data	network data
N_PCI	network protocol control information
N_PCItype	network protocol control information type
N_PDU	network protocol data unit
N_SA	network source address
N_SDU	network service data unit
N_TA	network target address
N_TAtype	network target address type
N_USData	network layer unacknowledged segmented data transfer service name
NW	network
NWL	network layer
OBD	on-board diagnostics

OSI	Open Systems Interconnection
PCI	protocol control information
SF	SingleFrame
SN	SequenceNumber
SPN	suspect parameter number
STmin	SeparationTime minimum
UDS	unified diagnostic services
WWH-OBD	world-wide harmonized OBD

4 Conventions

ISO 15765 is based on the conventions discussed in the OSI service conventions (ISO/IEC 10731) as they apply for diagnostic services.

5 Document overview

Figure 1 illustrates the most applicable application implementations utilizing the DoCAN protocol.

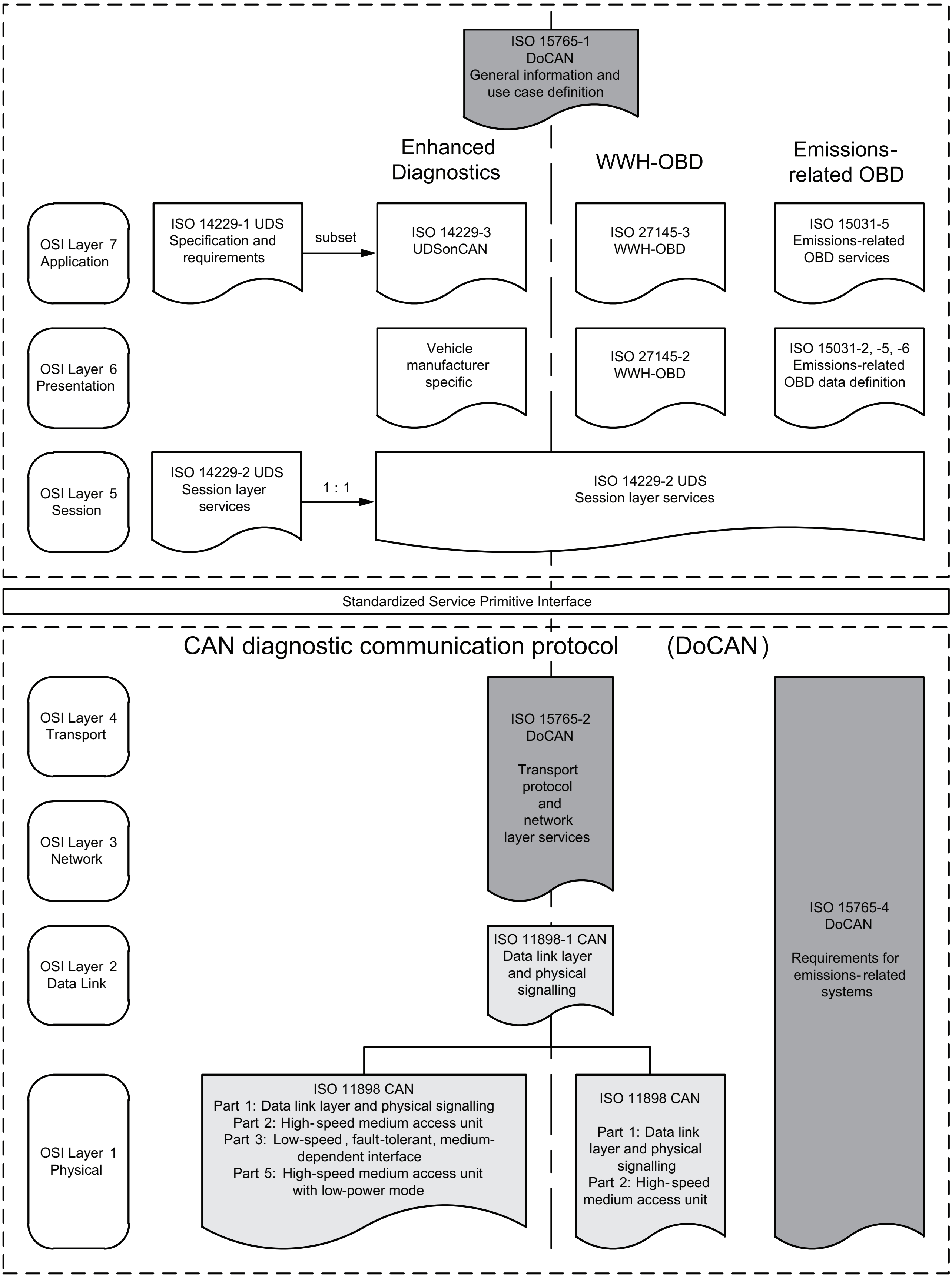


Figure 1 — DoCAN document reference according to the OSI model

6 Network layer overview

6.1 General

This part of ISO 15765 specifies an unconfirmed network layer communication protocol for the exchange of data between network nodes, e.g. from ECU to ECU, or between external test equipment and an ECU. If the data to be transferred do not fit into a single CAN frame, a segmentation method is provided.

In order to describe the functioning of the network layer, it is necessary to consider services provided to higher layers and the internal operation of the network layer.

6.2 Services provided by network layer to higher layers

The service interface defines a set of services that are needed to access the functions offered by the network layer, i.e. transmission/reception of data and setting of protocol parameters.

Two types of service are defined.

a) **Communication services**

   These services, of which the following are defined, enable the transfer of up to 4 095 bytes of data.

   1) N_USData.request

      This service is used to request the transfer of data. If necessary, the network layer segments the data.

   2) N_USData_FF.indication

      This service is used to signal the beginning of a segmented message reception to the upper layer.

   3) N_USData.indication

      This service is used to provide received data to the higher layers.

   4) N_USData.confirm

      This service confirms to the higher layers that the requested service has been carried out (successfully or not).

b) **Protocol parameter setting services**

   These services, of which the following are defined, enable the dynamic setting of protocol parameters.

   1) N_ChangeParameter.request

      This service is used to request the dynamic setting of specific internal parameters.

   2) N_ChangeParameter.confirm

      This service confirms to the upper layer that the request to change a specific protocol has been carried out (successfully or not).

6.3 Internal operation of network layer

The internal operation of the network layer provides methods for segmentation, transmission with FlowControl, and reassembly. The main purpose of the network layer is to transfer messages that might or might not fit in a single CAN frame. Messages that do not fit into a single CAN frame are segmented into multiple parts, where each can be transmitted in a CAN frame.

Figures 2 and 3 show, respectively, an example of an unsegmented message transmission and of a segmented message transmission.

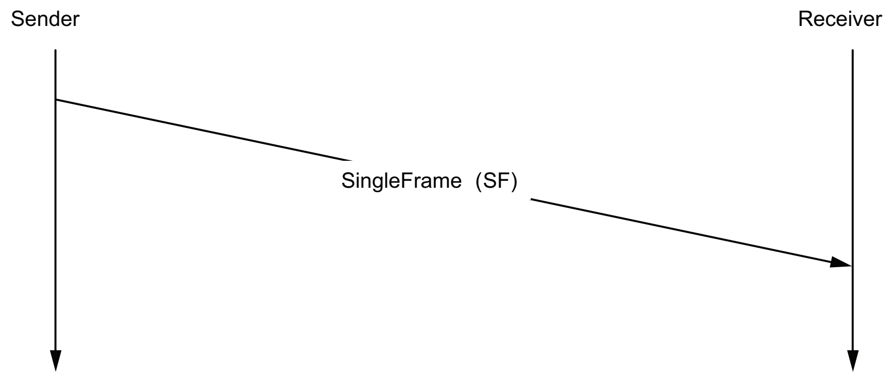


Figure 2 — Example of an unsegmented message

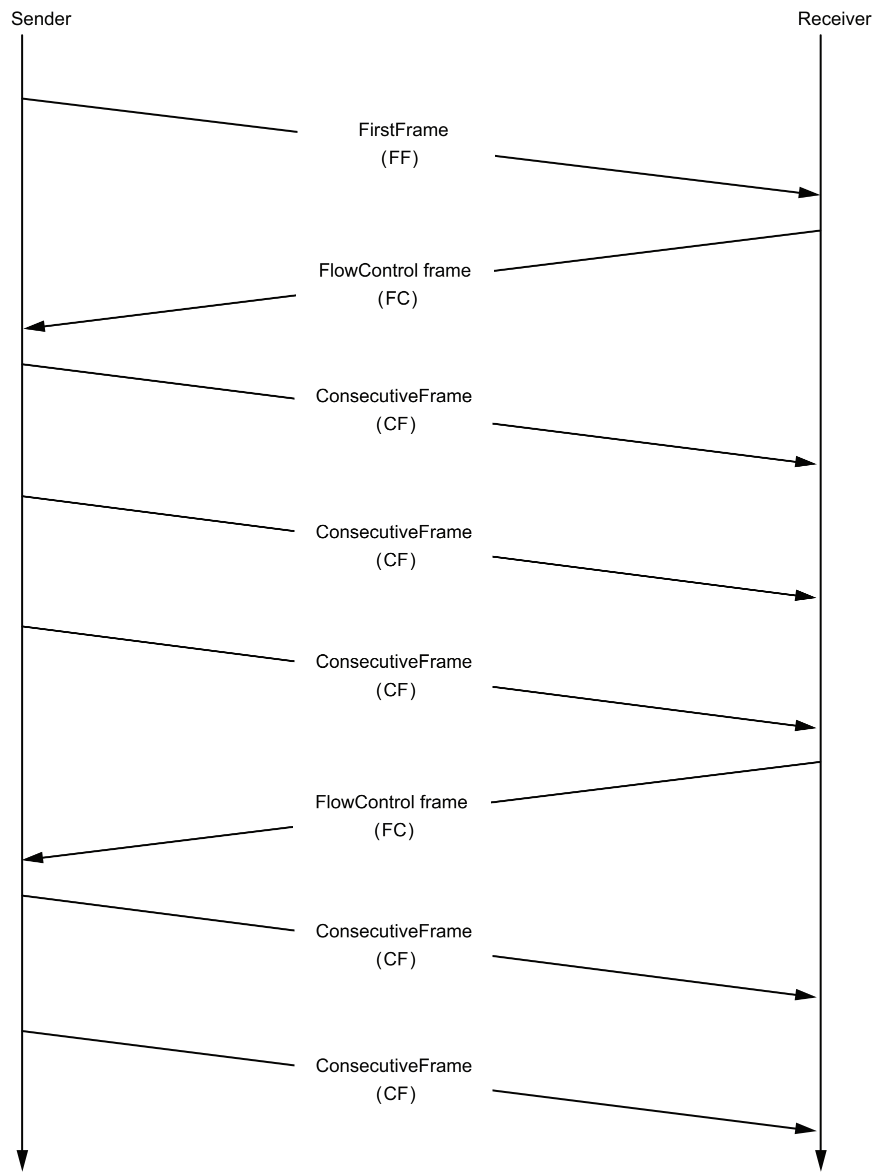


Figure 3 — Example of a segmented message

FlowControl is used to adjust the sender to the network layer capabilities of the receiver. This FlowControl scheme allows the use of diagnostic gateways and sub-networks.

7 Network layer services

7.1 General

All network layer services have the same general structure. To define the services, three types of service primitive are specified:

— a *service request primitive*, used by higher communication layers or the application to pass control information and data required to be transmitted to the network layer;

— a *service indication primitive*, used by the network layer to pass status information and received data to upper communication layers or the application;

— a *service confirmation primitive*, used by the network layer to pass status information to higher communication layers or the application.

This service specification does not specify an application programming interface, but only a set of service primitives that are independent of any implementation.

All network layer services have the same general format. Service primitives are written in the form:

```
service_name.type    (
                     parameter A,
                     parameter B
                     [,parameter C, ...]
                     )
```

where “service_name” is the name of the service, e.g. N_USData, “type” indicates the type of service primitive, and “parameter A, parameter B [,parameter C, ...]” are the N_SDU as a list of values passed by the service primitive. The square brackets indicate that this part of the parameter list may be empty.

The service primitives define how a service user (e.g. diagnostic application) cooperates with a service provider (e.g. network layer). The following service primitives are specified in this part of ISO 15765: request, indication and confirm.

— Using the service primitive *request* (service_name.request), a service user requests a service from the service provider.

— Using the service primitive *indication* (service_name.indication), the service provider informs a service user about an internal event of the network layer or the service request of a peer protocol layer entity service user.

— With the service primitive *confirm* (service_name.confirm), the service provider informs the service user about the result of a preceding service request of the service user.

7.2 Specification of network layer service primitives

7.2.1 N_USData.request

The service primitive requests transmission of <MessageData> with <Length> bytes from the sender to the receiver peer entities identified by the address information in N_SA, N_TA, N_TAtype [and N_AE] (see 7.3 for parameter definition).

```
N_USData.request    (
                    Mtype
                    N_SA
                    N_TA
                    N_TAtype
                    [N_AE]
                    <MessageData
                    <Length>
                    )
```

Each time the N_USData.request service is called, the network layer shall signal the completion (or failure) of the message transmission to the service user by issuing an N_USData.confirm service call.

7.2.2 N_USData.confirm

The N_USData.confirm service is issued by the network layer. The service primitive confirms the completion of an N_USData.request service identified by the address information in N_SA, N_TA, N_TAtype [and N_AE]. The parameter <N_Result> provides the status of the service request (see 7.3 for parameter definition).

```
N_USData.confirm    (
                    Mtype
                    N_SA
                    N_TA
                    N_TAtype
                    [N_AE]
                    <N_Result>
                    )
```

7.2.3 N_USData_FF.indication

The N_USData_FF.indication service is issued by the network layer. The service primitive indicates to the adjacent upper layer the arrival of a FirstFrame (FF) of a segmented message received from a peer protocol entity, identified by the address information in N_SA, N_TA, N_TAtype [and N_AE] (see 7.3 for parameter definition). This indication shall take place upon receipt of the FF of a segmented message.

```
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 N_USData.indication service call from the network layer, indicating the completion (or failure) of message reception.

An N_USData_FF.indication service call shall only be issued by the network layer if a correct FF message segment has been received.

If the network layer detects any type of error in an FF, then the message shall be ignored by the network layer and no N_USData_FF.indication shall be issued to the adjacent upper layer.

If the network layer receives an FF with a data length value (FF_DL) that is greater than the available receiver buffer size, then this shall be considered as an error condition and no N_USData_FF.indication shall be issued to the adjacent upper layer.

7.2.4 N_USData.indication

The N_USData.indication service is issued by the network layer. The service primitive indicates <N_Result> events and delivers <MessageData> with <Length> bytes received from a peer protocol entity identified by the address information in N_SA, N_TA, N_TAtype [and N_AE] to the adjacent upper layer (see 7.3 for parameter definition).

The parameters <MessageData> and <Length> are valid only if <N_Result> equals N_OK.

```
N_USData.indication        (
                           Mtype
                           N_SA
                           N_TA
                           N_TAtype
                           [N_AE]
                           <MessageData>
                           <Length>
                           <N_Result>
                           )
```

The N_USData.indication service call is issued after reception of a SingleFrame (SF) message or as an indication of the completion (or failure) of a segmented message reception.

If the network layer detects any type of error in an SF, then the message shall be ignored by the network layer and no N_USData.indication shall be issued to the adjacent upper layer.

7.2.5 N_ChangeParameters.request

The service primitive is used to request the change of an internal parameter's value on the local protocol entity. The <Parameter_Value> is assigned to the <Parameter> (see 7.3 for parameter definition).

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 message (N_USData.indication).

```
N_ChangeParameter.request    (
                             Mtype
                             N_SA
                             N_TA
                             N_TAtype
                             [N_AE]
                             <Parameter>
                             <Parameter_Value>
                             )
```

This is an optional service that can be replaced by implementation of fixed parameter values.

7.2.6 N_ChangeParameter.confirm

The service primitive confirms completion of an N_ChangeParameter.confirm service applying to a message identified by the address information in N_SA, N_TA, N_TAtype [and N_AE] (see 7.3 for parameter definition).

```
N_ChangeParameter.confirm    (
                             Mtype
                             N_SA
                             N_TA
                             N_TAtype
                             [N_AE]
                             <Parameter>
                             <Result_ChangeParameter>
                             )
```


7.3 Service data unit specification

7.3.1 Mtype, Message type

Type: enumeration

Range: diagnostics, remote diagnostics

Description: The parameter Mtype shall be used to identify the type and range of address information parameters included in a service call. This part of ISO 15765 specifies a range of two values for this parameter. The intention is that users of this part of ISO 15765 can extend the range of values by specifying other types and combinations of address information parameters to be used with the network layer protocol specified in this part of ISO 15765. For each such new range of address information, a new value for the Mtype parameter shall be specified to identify the new address information.

— If Mtype = diagnostics, then the address information N_AI shall consist of the parameters N_SA, N_TA, and N_TAtype.

— If Mtype = remote diagnostics, then the address information N_AI shall consist of the parameters N_SA, N_TA, N_TAtype, and N_AE.

7.3.2 N_AI, Address Information

7.3.2.1 N_AI description

These parameters refer to addressing information. As a whole, the N_AI parameters are used to identify the source address (N_SA), the target address (N_TA) of message senders and recipients, as well as the communication model for the message (N_TAtype) and the optional address extension (N_AE).

7.3.2.2 N_SA, Network Source Address

Type: 1 byte unsigned integer value

Range: 0x00 – 0xFF

Description: The N_SA parameter shall be used to encode the sending network layer protocol entity.

7.3.2.3 N_TA, Network Target Address

Type: 1 byte unsigned integer value

Range: 0x00 – 0xFF

Description: The N_TA parameter shall be used to encode one or multiple (depending on the N_TAtype: physical or functional) receiving network layer protocol entities.

7.3.2.4 N_TAtype, Network Target Address type

Type: enumeration

Range: physical, functional

Description: The parameter N_TAtype is an extension to the N_TA parameter. It shall be used to encode the communication model used by the communicating peer entities of the network layer. Two communication models are specified: 1 to 1 communication, called physical addressing, and 1 to n communication, called functional addressing.

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

— Functional addressing (1 to n communication) shall only be supported for SingleFrame communication.

7.3.2.5 N_AE, Network Address Extension

Type: 1 byte unsigned integer value

Range: 0x00 – 0xFF

Description: The N_AE parameter is used to extend the available address range for large networks, and to encode both sending and receiving network layer entities of sub-networks other than the local network where the communication takes place. N_AE is only part of the addressing information if Mtype is set to remote diagnostics.

7.3.3 <Length>

Type: 12 bits

Range: 0x001 - 0xFFF

Description: This parameter includes the length of data to be transmitted/received.

7.3.4 <MessageData>

Type: string of bytes

Range: not applicable

Description: This parameter includes all data that the higher-layer entities exchange.

7.3.5 <Parameter>

Type: enumeration

Range: STmin, BS

Description: This parameter identifies a parameter of the network layer.

7.3.6 <Parameter_Value>

Type: 1 byte unsigned integer value

Range: 0x00 – 0xFF

Description: This parameter is assigned to a protocol parameter <Parameter> as indicated in 7.2.5.

7.3.7 <N_Result>

Type: enumeration

Range: N_OK, N_TIMEOUT_A, N_TIMEOUT_Bs, N_TIMEOUT_Cr, N_WRONG_SN, N_INVALID_FS, N_UNEXP_PDU, N_WFT_OVRN, N_BUFFER_OVFLW, N_ERROR

Description: This parameter contains the status relating to the outcome of a service execution. If two or more errors are discovered at the same time, then the network layer entity shall use the parameter value found first in this list when indicating the error to the higher layers.

— N_OK

This value means that the service execution has been completed successfully; it can be issued to a service user on both the sender and receiver sides.

— N_TIMEOUT_A

This value is issued to the protocol user when the timer N_Ar/N_As has passed its time-out value N_As_{max}/N_Ar_{max}; it can be issued to service users on both the sender and receiver sides.

— N_TIMEOUT_Bs

This value is issued to the service user when the timer N_Bs has passed its time-out value N_Bs_{max}; it can be issued to the service user on the sender side only.

— N_TIMEOUT_Cr

This value is issued to the service user when the timer N_Cr has passed its time-out value N_Cr_{max}; it can be issued to the service user on the receiver side only.

— N_WRONG_SN

This value is issued to the service user upon receipt of an unexpected SequenceNumber (PCI.SN) value; it can be issued to the service user on the receiver side only.

— N_INVALID_FS

This value is issued to the service user when an invalid or unknown FlowStatus value has been received in a FlowControl (FC) N_PDU; it can be issued to the service user on the sender side only.

— N_UNEXP_PDU

This value is issued to the service user upon receipt of an unexpected protocol data unit; it can be issued to the service user on the receiver side only.

— N_WFT_OVRN

This value is issued to the service user when the receiver has transmitted N_WFT_{max} FlowControl N_PDUs with FlowStatus = WAIT in a row and following this it cannot meet the performance requirement for the transmission of a FlowControl N_PDU with FlowStatus = ClearToSend. It can be issued to the service user on the receiver side only.

— N_BUFFER_OVFLW

This value is issued to the service user upon receipt of a FlowControl (FC) N_PDU with FlowStatus = OVFLW. It indicates that the buffer on the receiver side of a segmented message transmission cannot store the number of bytes specified by the FirstFrame DataLength (FF_DL) parameter in the FirstFrame and therefore the transmission of the segmented message was aborted. It can be issued to the service user on the sender side only.

— N_ERROR

This is the general error value. It shall be issued to the service user when an error has been detected by the network layer and no other parameter value can be used to better describe the error. It can be issued to the service user on both the sender and receiver sides.

7.3.8 <Result_ChangeParameter>

Type: enumeration.

Range: N_OK, N_RX_ON, N_WRONG_PARAMETER, N_WRONG_VALUE

Description: This parameter contains the status relating to the outcome of a service execution.

— N_OK

This value means that the service execution has been completed successfully; it can be issued to a service user on both the sender and receiver sides.

— N_RX_ON

This value is issued to the service user to indicate that the service did not execute since reception of the message identified by <N_AI> was taking place; it can be issued to the service user on the receiver side only.

— N_WRONG_PARAMETER

This value is issued to the service user to indicate that the service did not execute due to an undefined <Parameter>; it can be issued to a service user on both the receiver and sender sides.

— N_WRONG_VALUE

This value is issued to the service user to indicate that the service did not execute due to an out-of-range <Parameter_Value>; it can be issued to a service user on both the receiver and sender sides.

8 Transport layer protocol

8.1 Protocol functions

The transport layer protocol performs the following functions:

a) transmission/reception of messages up to 4 095 data bytes;

b) reporting of transmission/reception completion (or failure).

8.2 SingleFrame transmission

Transmission of messages of up to six (in the case of extended or mixed addressing) or seven (in the case of normal addressing) data bytes is performed via transmission of a unique N_PDU (see 8.4), called SF (see Figure 4).

Reception of messages of up to six or seven data bytes is performed via reception of a unique N_PDU.

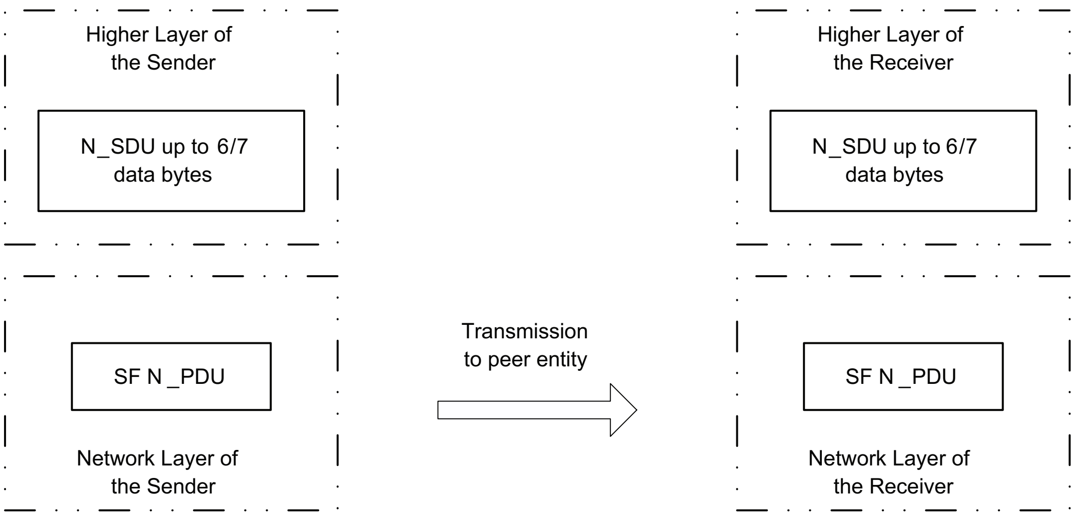


Figure 4 — Example of a SingleFrame (SF) transmission

8.3 Multiple-frame transmission

Transmission of longer messages is performed by segmenting the message and transmitting multiple N_PDUs. Reception of longer messages is performed by receiving multiple N_PDUs and reassembling of received data bytes (concatenation). The multiple N_PDUs are called FirstFrame (for the first N_PDU of the message) and ConsecutiveFrame (for all the following N_PDUs).

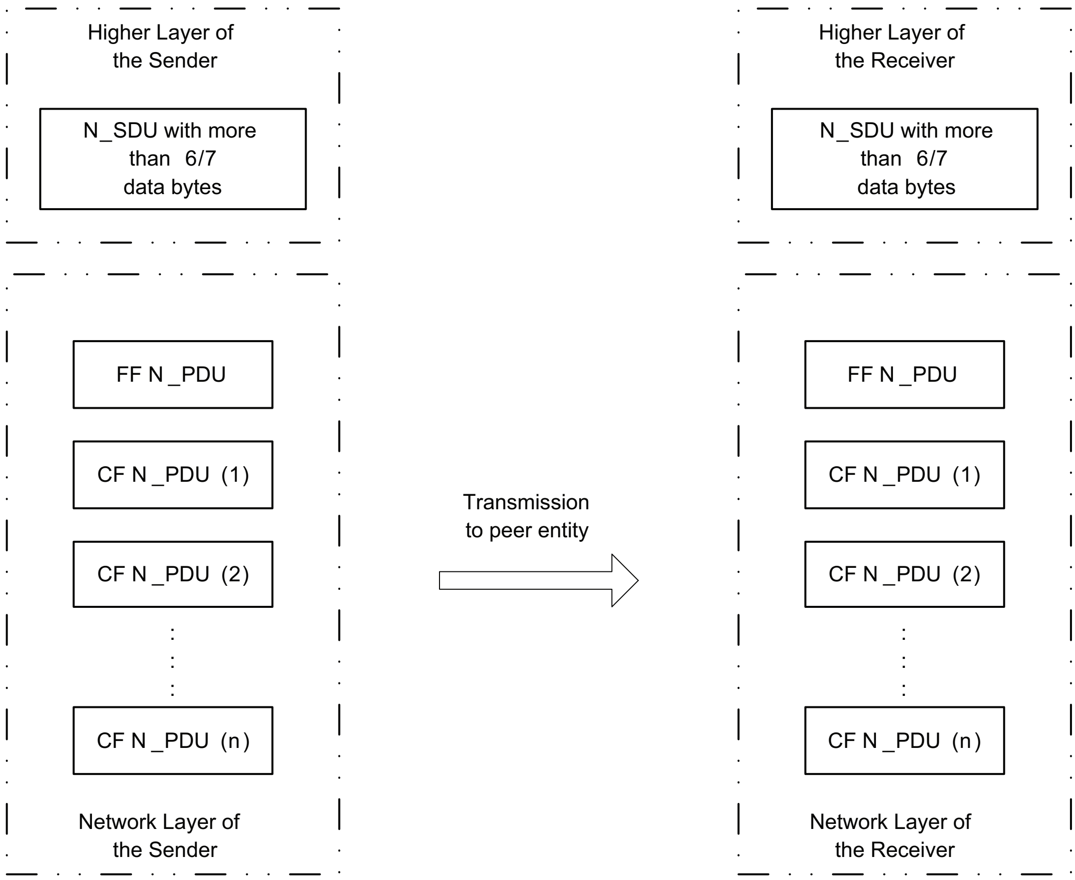
The receiver of a multiple N_PDU message has the possibility of adapting the transmission throughput to its capability by means of the FlowControl mechanism, using the FlowControl protocol data units (FC N_PDU).

Messages longer than six or seven data bytes are segmented into

— a FirstFrame protocol data unit (FF N_PDU), containing the first five (in the case of extended or mixed addressing) or six (in the case of normal addressing) data bytes, and

— one or more ConsecutiveFrame protocol data units (CF N_PDU), each containing six or seven data bytes. The last CF N_PDU contains only the remaining valid data bytes, and may therefore be less than six or seven data bytes long.

Figure 5 shows segmentation on the sender side and reassembly on the receiver side.



NOTE The FC N_PDU issued by the receiver in response to reception of the FF N_PDU is not shown.

Figure 5 — Example of a multiple-frame transmission (segmentation and reassembly)

The message length is transmitted in the FF N_PDU. All CF N_PDUs are numbered by the sender to help the receiver reassemble them in the same order.

The FlowControl mechanism (see Figure 6) allows the receiver to inform the sender about the receiver's capabilities. Since different nodes may have different capabilities, the FlowControl sent by the receiver informs the sender about its capabilities. The sender shall conform to the receiver's capabilities.

These capabilities are defined 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.

— SeparationTime minimum (STmin): the minimum time the sender is to wait between transmission of two CF N_PDUs.

If the server is the receiver of a segmented message transfer (i.e. the sender of the FlowControl frame) it may choose either to use the same values for BS and STmin in subsequent FC (CTS) frames of the same segmented message or to vary these values from FC to FC frame.

If the client, connected to an ISO 15765-compliant in-vehicle diagnostic network, is the receiver of a segmented message transfer (i.e. the sender of the FlowControl frame), it shall use the same values for BS and STmin in subsequent FC (CTS) frames of the same segmented message.

NOTE 1 If the client deviates from the requirement not to change BS and STmin during the same transmission, there is no guarantee that the server will successfully communicate with the client.

If the client is the sender of a segmented data transmission (i.e. the receiver of the FlowControl frame), it shall adjust to the values of BS and STmin from each FC (CTS) received during the same segmented data transmission.

NOTE 2 For in-vehicle gateway implementations (i.e. routing takes place on OSI layer 4; see ISO 14229-2[4]) the vehicle manufacturer chooses either that the FC parameters BS and STmin vary during the transmission of a single segmented message or that these parameters are static values. Depending on this design decision the vehicle manufacturer needs to ensure that server implementations are compatible with the respective in-vehicle gateway implementation.

Figure 6 depicts the FlowControl (FC) mechanism.

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).

Each time the sender/receiver waits for an N_PDU from the receiver/sender, a timeout mechanism allows detection of a transmission failure (see 8.7.2).

By means of FC N_PDUs, the receiver has the possibility of authorizing transmission of the following CF N_PDUs, to delay transmission of that authorization or to deny reception of a segmented message in the case that the number of bytes to be transferred exceeds the number of bytes that can be stored in the receiver buffer:

— FC.CTS: continue to send, the authorization to continue;

— FC.WAIT: the request to continue to wait;

— FC.OVFLW: buffer overflow, the indication that the number of bytes specified in the FirstFrame of the segmented message exceeds the number of bytes that can be stored in the buffer of the receiver entity.

There is an upper limit to the number of FC.WAIT a receiver is allowed to send in a row, called N_WFTmax. This parameter is a system design constant and is not transmitted in the first FC N_PDU.

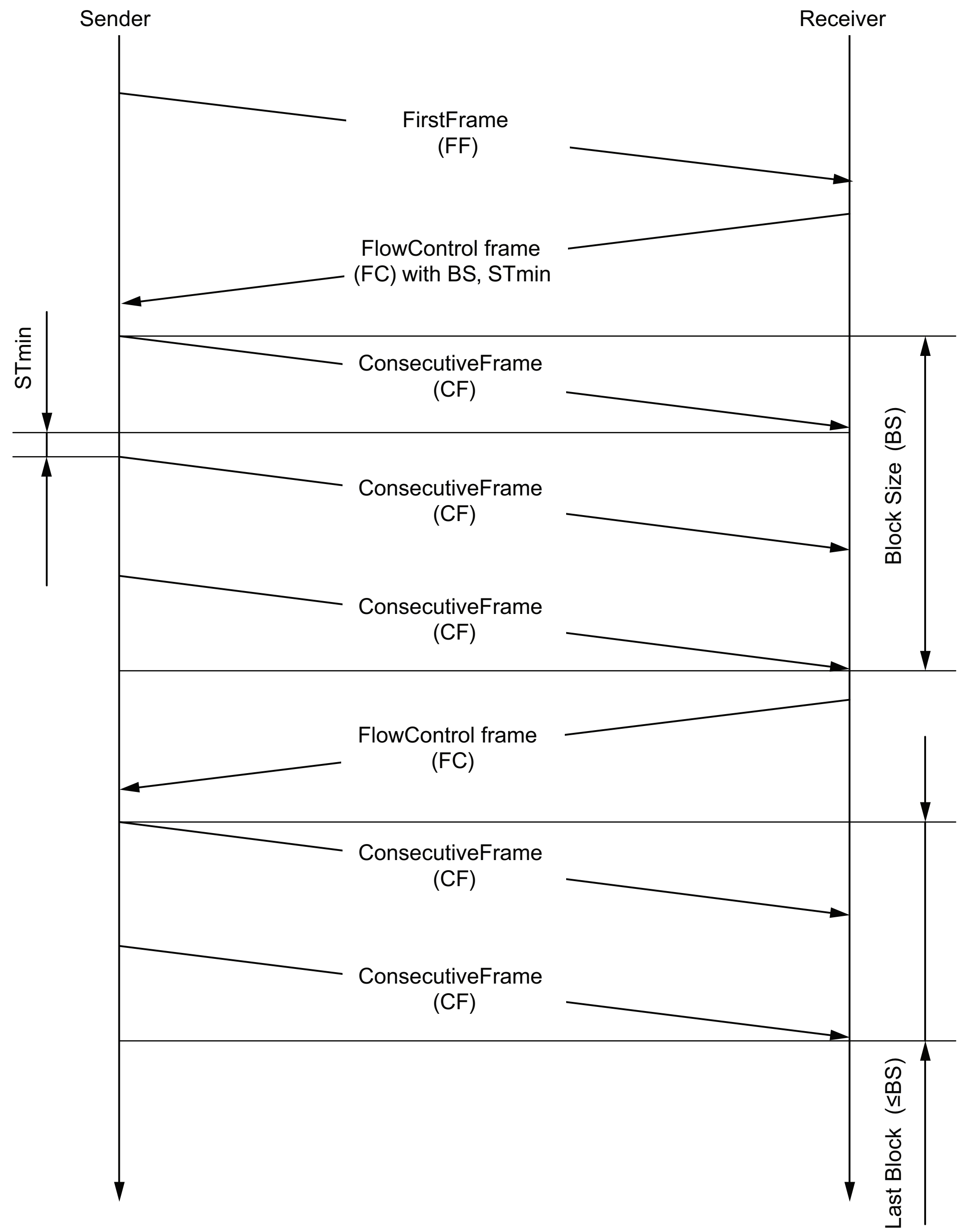


Figure 6 — FlowControl (FC) mechanism

8.4 Transport layer protocol data units

8.4.1 Protocol data unit types

The communication between the peer protocol entities of the network layer in different nodes is done by means of exchanging N_PDUs.

This part of ISO 15765 specifies four different types of transport layer protocol data units — single-frame (SF N_PDU), first-frame (FF N_PDU), consecutive-frame (CF N_PDU) and flow-control (FC N_PDU) — which are used to establish a communication path between the peer network layer entities, to exchange communication parameters, to transmit user data and to release communication resources.

8.4.2 SF N_PDU

The SF N_PDU is identified by the single-frame protocol control information (SF N_PCI). The SF N_PDU shall be sent out by the sending network entity and can be received by one or multiple receiving network entities. It shall be sent out to transfer a service data unit that can be transferred via a single service request to the data link layer, and to transfer unsegmented messages.

8.4.3 FF N_PDU

The FF N_PDU is identified by the first-frame protocol control information (FF N_PCI). The FF N_PDU shall be sent out by the sending network entity and received by a unique receiving network entity for the duration of the segmented message transmission. It identifies the first N_PDU of a segmented message transmitted by a network sending entity and received by a receiving network entity. The receiving network layer entity shall start assembling the segmented message on receipt of a FF N_PDU.

8.4.4 CF N_PDU

The CF N_PDU is identified by the consecutive-frame protocol control information (CF N_PCI). The CF N_PDU transfers segments (N_Data) of the service data unit message data (<MessageData>). All N_PDUs transmitted by the sending entity after the FF N_PDU shall be encoded as CF N_PDUs. The receiving entity shall pass the assembled message to the service user of the network receiving entity after the last CF N_PDU has been received. The CF N_PDU shall be sent out by the sending network entity and received by a unique receiving network entity for the duration of the segmented message transmission.

8.4.5 FC N_PDU

The FC N_PDU is identified by the flow-control protocol control information (FC N_PCI). The FC N_PDU instructs a sending network entity to start, stop or resume transmission of CF N_PDUs. It shall be sent by the receiving network layer entity to the sending network layer entity, when ready to receive more data, after correct reception of

a) an FF N_PDU, or

b) the last CF N_PDU of a block of ConsecutiveFrames, if further ConsecutiveFrames need to be sent.

The FC N_PDU can also inform a sending network entity to pause transmission of CF N_PDUs during a segmented message transmission or to abort the transmission of a segmented message if the length information (FF_DL) in the FF N_PDU transmitted by the sending entity exceeds the buffer size of the receiving entity.

8.4.6 Protocol data unit field description

8.4.6.1 N_PDU format

The protocol data unit (N_PDU) enables the transfer of data between the network layer in one node and the network layer in one or more other nodes (peer protocol entities). All N_PDUs consist of three (3) fields, as given in Table 2.

Table 2 — N_PDU format

Address information	Protocol control information	Data field
N_AI	N_PCI	N_Data

8.4.6.2 Address information (N_AI)

The N_AI is used to identify the communicating peer entities of the network layer. The N_AI information received in the N_SDU — N_SA, N_TA, N_TAtype [and N_AE] — shall be copied and included in the N_PDU. If the message data (<MessageData> and <Length>) received in the N_SDU is so long that segmentation is needed 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.

This field contains address information that identifies the type of message exchanged, and the recipient(s) and sender between whom data exchange takes place.

NOTE For a detailed description of address information, see 7.3.2.

8.4.6.3 Protocol control information (N_PCI)

This field identifies the type of N_PDUs exchanged. It is also used to exchange other control parameters between the communicating network layer entities.

NOTE For a detailed specification of all N_PCI parameters, see 8.5.

8.4.6.4 Data Field (N_Data)

The N_Data in the N_PDU is used to transmit the service user data received in the <MessageData> parameter in the N_USData.request service call. The <MessageData>, if needed, is segmented into smaller parts that each fit into the N_PDU data field before they are transmitted over the network.

The size of N_Data depends on the N_PDU type and the address format chosen.

8.5 Protocol control information specification

8.5.1 N_PCI

Each N_PDU is identified by means of an N_PCI. See Table 3 and Table 4.

Table 3 — Summary of N_PCI bytes

N_PDU name	N_PCI bytes			
	Byte #1		Byte #2	Byte #3
	Bits 7 – 4	Bits 3 – 0		
SingleFrame (SF)	N_PCItype = 0	SF_DL	N/A	N/A
FirstFrame (FF)	N_PCItype = 1	FF_DL		N/A
ConsecutiveFrame (CF)	N_PCItype = 2	SN	N/A	N/A
FlowControl (FC)	N_PCItype = 3	FS	BS	STmin

Table 4 — Definition of N_PCItype values

Value	Description
0x0	**SingleFrame** For unsegmented messages, the network layer protocol provides an optimized implementation of the network protocol with the message length embedded in the PCI byte only. SingleFrame (SF) shall be used to support the transmission of messages that can fit in a single CAN frame.
0x1	**FirstFrame** A FirstFrame (FF) shall only be used to support the transmission of messages that cannot fit in a single CAN frame, i.e. segmented messages. The FirstFrame of a segmented message is encoded as an FF. On receipt of an FF, the receiving network layer entity shall start assembling the segmented message.
0x2	**ConsecutiveFrame** When sending segmented data, all ConsecutiveFrames following the FF are encoded as ConsecutiveFrame (CF). On receipt of a CF, the receiving network layer entity shall assemble the received data bytes until the whole message is received. The receiving entity shall pass the assembled message to the adjacent upper protocol layer after the last frame of the message has been received without error.
0x3	**FlowControl** The purpose of FlowControl (FC) is to regulate the rate at which CF N_PDUs are sent to the receiver. Three distinct types of FC protocol control information are specified to support this function. The type is indicated by a field of the protocol control information called FlowStatus (FS), as defined in 8.5.5.2.
0x4 – 0xF	**Reserved** This range of values is reserved by this part of ISO 15765.

8.5.2 SingleFrame N_PCI parameter definition

8.5.2.1 SF N_PCI byte

Table 5 provides an overview of the SF N_PCI byte.

Table 5 — Overview of SF N_PCI byte

N_PDU name	SF N_PCI byte							
	Byte #1							
	7	6	5	4	3	2	1	0
SingleFrame	0	0	0	0	SF_DL			

The parameter SingleFrame DataLength (SF_DL) is used in the SF N_PDU to specify the number of service user data bytes. See Table 6.

Table 6 — Definition of SF_DL values

Value	Description
0x0	**Reserved** This value is reserved by this part of ISO 15765.
0x1 – 0x6	**SingleFrame DataLength (SF_DL)** The SF_DL is encoded in the low nibble of N_PCI byte #1 value. It shall be assigned the value of the service parameter <Length>.
0x7	**SingleFrame DataLength (SF_DL) with normal addressing** SF_DL = 7 is only allowed with normal addressing.
0x8 – 0xF	**Invalid** This range of values is invalid.

8.5.2.2 SF_DL error handling

If the network layer receives an SF with an SF_DL equal to zero then the network layer shall ignore the received SF N_PDU.

If the network layer receives an SF with an SF_DL greater than seven, when using normal addressing, or greater than six for extended or mixed addressing, then the network layer shall ignore the received SF N_PDU.

8.5.3 FirstFrame N_PCI parameter definition

8.5.3.1 FF N_PCI bytes

Table 7 provides an overview of the FF N_PCI bytes.

Table 7 — Overview of FF N_PCI bytes

N_PDU name	FF N_PCI bytes								
	Byte #1								Byte #2
	7	6	5	4	3	2	1	0	7 – 0
FirstFrame	0	0	0	1	FF_DL				

8.5.3.2 FirstFrame DataLength (FF_DL) parameter definition

The parameter FF_DL is used in the FF N_PDU to specify the number of service user data bytes. See Table 8.

Table 8 — Definition of FF_DL values

Value	Description
0x0 – 0x6	**Invalid** This range of values is invalid.
0x7	**FirstFrame DataLength (FF_DL) with extended addressing or mixed addressing** FF_DL = 7 is only allowed with extended or mixed addressing.
0x8 – 0xFFF	**FirstFrame DataLength (FF_DL)** 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 N_PCI byte #2 and the most significant bit (MSB) is bit 3 of the N_PCI 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>.

8.5.3.3 FF_DL error handling

If the network layer receives an FF with an FF_DL that is greater than the available receiver buffer size, then this shall be considered as an error condition. The network layer shall abort the message reception and send an FC N_PDU with the parameter FlowStatus = Overflow.

If the network layer receives an FF with an FF_DL that is less than eight when using normal addressing or less than seven when using extended or mixed addressing, then the network layer shall ignore the received FF N_PDU and not transmit an FC N_PDU.

8.5.4 ConsecutiveFrame N_PCI parameter definition

8.5.4.1 CF N_PCI byte

Table 9 provides an overview of the CF N_PCI byte.

Table 9 — Overview of CF N_PCI byte

<table>
<tr><th rowspan="3">N_PDU name</th><th colspan="8">CF N_PCI byte</th></tr>
<tr><th colspan="8">Byte #1</th></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>ConsecutiveFrame</td><td>0</td><td>0</td><td>1</td><td>0</td><td colspan="4">SN</td></tr>
</table>

8.5.4.2 SequenceNumber (SN) parameter definition

The parameter SN is used in the ConsecutiveFrame (CF) N_PDU to specify the following:

— the numeric ascending order of the ConsecutiveFrames;

— that the SN shall start with zero for all segmented messages; the FirstFrame (FF) shall be assigned the value zero;. it does not include an explicit SequenceNumber in the N_PCI field but shall be treated as the segment number zero;

— that the SN of the first CF immediately following the FF shall be set to one;

— that the SN shall be incremented by one for each new CF that is transmitted during a segmented message transmission;

— that the SN value shall not be affected by any FlowControl (FC) frame;

— that when the SN reaches the value of 15, it shall wraparound and be set to zero for the next CF.

This shall lead to the sequence given in Table 10.

Table 10 — Summary of SN definition

N_PDU	FF	CF	CF	CF	CF	CF	CF	CF
SN	0x0	0x1	…	0xE	0xF	0x0	0x1	…

See Table 11 for a definition of SN values.

Table 11 — Definition of SN values

Value	Description
0x0 – 0xF	**SequenceNumber (SN)** The SequenceNumber (SN) shall be encoded in the lower nibble bits of N_PCI byte #1. The SN shall be set to a value within the range of zero to 15.

8.5.4.3 SequenceNumber (SN) error handling

If a CF N_PDU message is received with an unexpected SequenceNumber not in accordance with the definition in 8.5.4.2, the message reception shall be aborted, 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.

8.5.5 FlowControl N_PCI parameter definition

8.5.5.1 FlowControl N_PCI bytes

Table 12 provides an overview of the FC N_PCI bytes.

Table 12 — Overview of FC N_PCI bytes

N_PDU name	FC N_PCI bytes									
	Byte #1								Byte #2	Byte #3
	7	6	5	4	3	2	1	0		
FlowControl	0	0	1	1	FS				BS	STmin

8.5.5.2 FlowStatus (FS) parameter definition

The parameter FlowStatus (FS) indicates whether the sending network entity can proceed with the message transmission.

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

Table 13 defines the FS values.

Table 13 — Definition of FS values

Value	Description
0x0	**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.
0x1	**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 STmin (SeparationTime minimum) in the FlowControl message are not relevant and shall be ignored.
0x2	**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 STmin (SeparationTime minimum) in the FlowControl message are not relevant and shall be ignored.
0x3 – 0xF	**Reserved** This range of values is reserved by this part of ISO 15765.

8.5.5.3 FlowStatus (FS) error handling

If an FC N_PDU message is received with an invalid (reserved) FS parameter value, the message transmission shall be 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.

8.5.5.4 BlockSize (BS) parameter definition

The BS parameter shall be encoded in byte #2 of the FC N_PCI.

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

EXAMPLE If BS is equal to 20 (decimal), then the block will consist of 20 (decimal) CF N_PDUs.

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

Table 14 provides an overview of the FC N_PCI byte.

Table 14 — Definition of BS values

Value	Description
0x00	**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.
0x01 – 0xFF	**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.

8.5.5.5 SeparationTime minimum (STmin) parameter definition

The STmin parameter shall be encoded in byte #3 of the FC N_PCI.

This time is specified by the receiving entity. The STmin parameter value specifies the minimum time gap allowed between the transmissions of two ConsecutiveFrame network protocol data units (CFs). See Table 15.

Table 15 — Definition of STmin values

Value	Description
0x00 – 0x7F	**SeparationTime minimum (STmin) range: 0 ms – 127 ms** The units of STmin in the range 0x00 – 0x7F (0_d – 127_d) are absolute milliseconds (ms).
0x80 – 0xF0	**Reserved** This range of values is reserved by this part of ISO 15765.
0xF1 – 0xF9	**SeparationTime minimum (STmin) range: 100 μs – 900 μs** The units of STmin in the range 0xF1 – 0xF9 (241_d – 249_d) are even 100 μs, where parameter value 0xF1 represents 100 μs and parameter value 0xF9 represents 900 μs.
0xFA – 0xFF	**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 ConsecutiveFrame (CF) and ends at the request for the transmission of the next CF.

EXAMPLE If STmin is equal to 10 (decimal), then the minimum ST authorized between ConsecutiveFrame network protocol data units is equal to 10 ms.

8.5.5.6 SeparationTime minimum (STmin) error handling

If an FC N_PDU message is received with a reserved STmin parameter value, then the sending network entity shall use the longest STmin value specified by this part of ISO 15765 (0x7F = 127 ms) instead of the value received from the receiving network entity for the duration of the on-going segmented message transmission.

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.

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

The purpose of this variable is to avoid communication sender nodes being potentially hooked up in case of a fault condition whereby the latter could be waiting continuously. 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 parameter shall indicate how many FC N_PDU WAITs can be transmitted by the receiver in a row.

— The N_WFTmax parameter upper limit shall be user defined at system generation time.

— The N_WFTmax parameter shall only be used on the receiving network entity during message reception.

— If the N_WFTmax parameter value is set to zero, then FlowControl shall rely upon FlowControl continue to send FC N_PDU CTS only. FlowControl wait (FC N_PDU WT) shall not be used by that network entity.

8.7 Network layer timing

8.7.1 Timing parameters

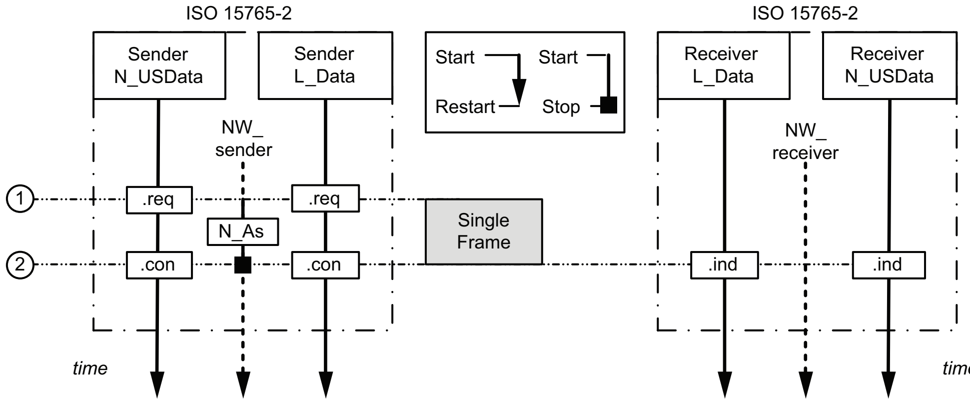
Performance requirement values are the binding communication requirements to be met by each communication peer in order to be compliant with the specification. A certain application may define specific performance requirements within the ranges defined in Table 16.

Timeout values are defined to be higher than the values for the performance requirements in order to ensure a working system and to overcome communication conditions where the performance requirement can absolutely not be met (e.g. high bus load). Specified timeout values shall be treated as the lower limit for any given implementation. The real timeout shall occur no later than at the specified timeout value + 50 %.

The network layer shall issue an appropriate service primitive to the network layer service user upon detection of an error condition.

If a communication path is established between peer protocol entities, identified by N_AI (see 7.3.2.1 and 8.4.6.2 for further details), a single set of network layer timing parameters is assigned statically to this communication path. For the selection of the network layer timing parameters, no other information besides N_AI is used. If different network layer timing parameters are required for different use cases, then separate communication paths shall be established using different N_AI parameters, e.g. different N_TA and/or N_SA shall be defined for each individual use case that requires different network layer timing parameters.

Figure 7 shows the network layer timing parameters of an unsegmented message, while Table 16 defines the network layer timing parameter values and their corresponding start and end positions based on the data link layer services.



1 **Sender N_USData.req**: the session layer issues an unsegmented message to the transport/network layer.
Sender L_Data.req: the transport/network layer transmits the SingleFrame to the data link layer and starts the N_As timer.

2 **Receiver L_Data.ind**: the data link layer issues to the transport/network layer the reception of the CAN frame.
Receiver N_USData.ind: the transport/network layer issues to the session layer the completion of the unsegmented message.
Sender L_Data.con: the data link layer confirms to the transport/network layer that the CAN frame has been acknowledged. The sender stops the N_As timer.
Sender N_USData.con: the transport/network layer issues to the session layer the completion of the unsegmented message.

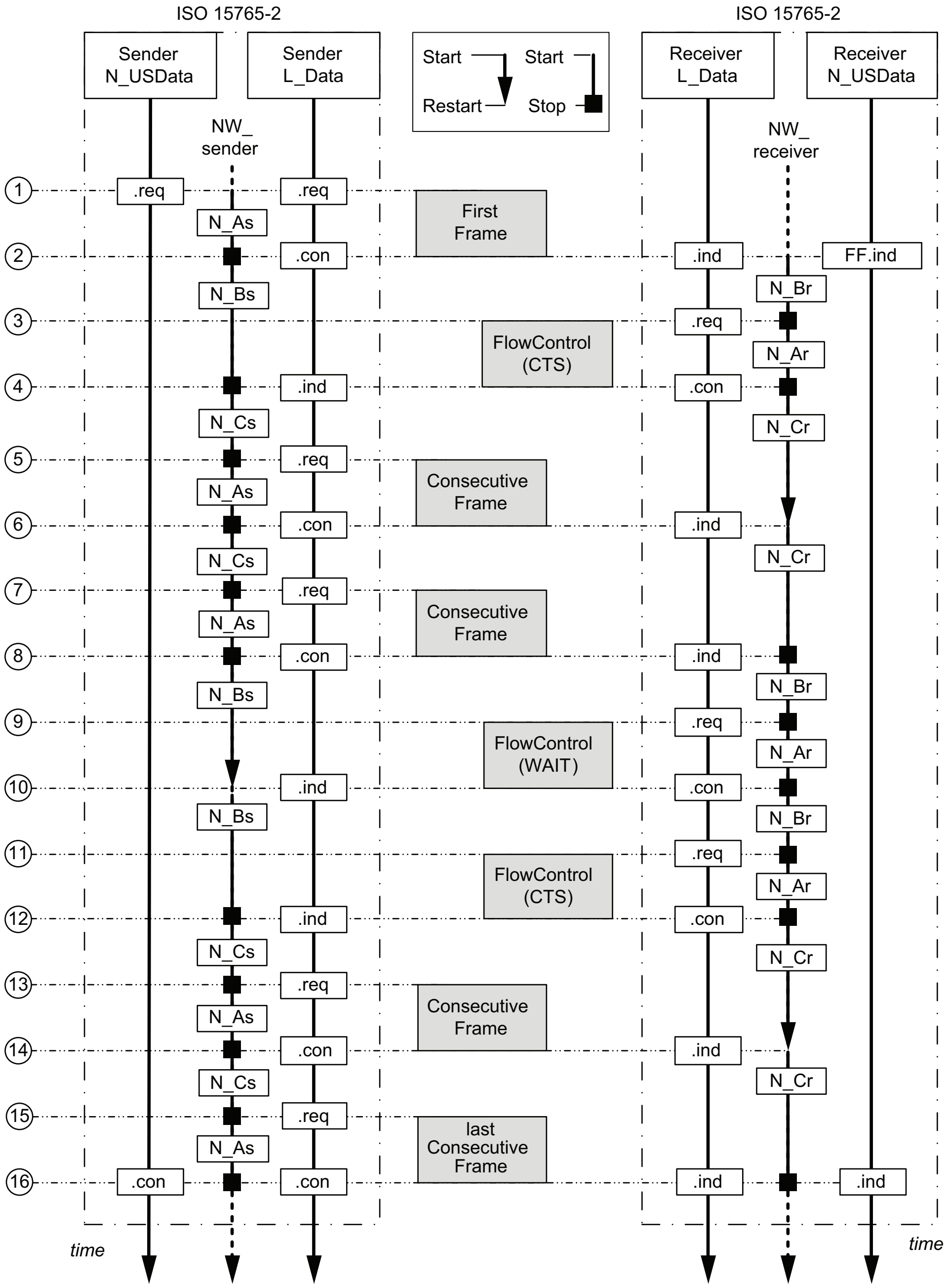
Figure 7 — Placement of network layer timing parameters — Unsegmented message

Table 16 — Network layer timing parameter values

Timing parameter[a]	Description	Data link layer service		Timeout	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	—

[a] s sender of the message
r receiver of the message

Figure 8 shows the network layer timing parameters of a segmented message, while Table 16 defines the network layer timing parameter values and their corresponding start and end positions based on the data link layer services.



1 **Sender N_USData.req**: the session layer issues a segmented message to the transport/network layer.
Sender L_Data.req: the transport/network layer transmits the FirstFrame to the data link layer and starts the N_As timer.

2 **Receiver L_Data.ind**: the data link layer issues to the transport/network layer the reception of the CAN frame. The receiver starts the N_Br timer.
Receiver N_USDataFF.ind: the transport/network layer issues to the session layer the reception of a FirstFrame of a segmented message.
Sender L_Data.con: the data link layer confirms to the transport/network layer that the CAN frame has been acknowledged. The sender stops the N_As timer and starts the N_Bs timer.

3 **Receiver L_Data.req**: the transport/network layer transmits the FlowControl (ContinueToSend and BlockSize value = 2d) to the data link layer and starts the N_Ar timer.

4 **Sender L_Data.ind**: the data link layer issues to the transport/network layer the reception of the CAN frame. The sender stops the N_Bs timer and starts the N_Cs timer.
Receiver L_Data.con: the data link layer confirms to the transport/network layer that the CAN frame has been acknowledged. The receiver stops the N_Ar timer and starts the N_Cr timer.

5 **Sender L_Data.req**: the transport/network layer transmits the first ConsecutiveFrame to the data link layer and starts the N_As timer.

6 **Receiver L_Data.ind**: the data link layer issues to the transport/network layer the reception of the CAN frame. The receiver restarts the N_Cr timer.
Sender L_Data.con: the data link layer confirms to the transport/network layer that the CAN frame has been acknowledged. The sender stops the N_As timer and starts the N_Cs timer according to the separation time value (STmin) of the previous FlowControl.

7 **Sender L_Data.req**: when the N_Cs timer is elapsed (STmin), the transport/network layer transmits the next ConsecutiveFrame to the data link layer and starts the N_As timer.

8 **Receiver L_Data.ind**: the data link layer issues to the transport/network layer the reception of the CAN frame. The receiver stops the N_Cr timer and starts the N_Br timer.
Sender L_Data.con: the data link layer confirms to the transport/network layer that the CAN frame has been acknowledged. The sender stops the N_As timer and starts the N_Bs timer. The sender is waiting for the next FlowControl.

9 **Receiver L_Data.req**: the transport/network layer transmits the FlowControl (Wait) to the data link layer and starts the N_Ar timer.

10 **Sender L_Data.ind**: the data link layer issues to the transport/network layer the reception of the CAN frame. The sender restarts the N_Bs timer.
Receiver L_Data.con: the data link layer confirms to the transport/network layer that the CAN frame has been acknowledged. The receiver stops the N_Ar timer and starts the N_Br timer.

11 **Receiver L_Data.req**: the transport/network layer transmits the FlowControl (ContinueToSend) to the data link layer and starts the N_Ar timer.

12 **Sender L_Data.ind**: the data link layer issues to the transport/network layer the reception of the CAN frame. The sender stops the N_Bs timer and starts the N_Cs timer.
Receiver L_Data.con: the data link layer confirms to the transport/network layer that the CAN frame has been acknowledged. The receiver stops the N_Ar timer and starts the N_Cr timer.

13 **Sender L_Data.req**: the transport/network layer transmits the ConsecutiveFrame to the data link layer and starts the N_As timer.

14 **Receiver L_Data.ind**: the data link layer issues to the transport/network layer the reception of the CAN frame. The receiver restarts the N_Cr timer.
Sender L_Data.con: the data link layer confirms to the transport/network layer that the CAN frame has been acknowledged. The sender stops the N_As timer and starts the N_Cs timer according to the separation time value (STmin) of the previous FlowControl.

15 **Sender L_Data.req**: when the N_Cs timer is elapsed (STmin), the transport/network layer transmits the last ConsecutiveFrame to the data link layer and starts the N_As timer.

16 **Receiver L_Data.ind**: the data link layer issues to the transport/network layer the reception of the CAN frame. The receiver stops the N_Cr timer.
Receiver N_USData.ind: the transport/network layer issues to the session layer the completion of the segmented message.
Sender L_Data.con: the data link layer confirms to the transport/network layer that the CAN frame has been acknowledged. The sender stops the N_As timer.
Sender N_USData.con: the transport/network layer issues to session layer the completion of the segmented message.

Figure 8 — Placement of network layer timing parameters — Segmented message

8.7.2 Network layer timeouts

Table 17 defines the cause and action in a network layer timeout.

Table 17 — Network layer timeout error handling

Timeout	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

8.7.3 Unexpected arrival of N_PDU

An unexpected N_PDU is defined as one that has been received by a node outside the expected order of N_PDUs. It could be an N_PDU defined by this part of ISO 15765 (SF N_PDU, FF N_PDU, CF N_PDU or FC N_PDU) that is received out of the normal expected order, or else it could be an unknown N_PDU that cannot be interpreted by the definitions given in this part of ISO 15765.

As a general rule, arrival of an unexpected N_PDU from any node shall be ignored, with the exception of SF N_PDUs and physically addressed FF N_PDUs; functionally addressed FirstFrames shall be ignored. When the specified action is to ignore an unexpected N_PDU, this means that the network layer shall not notify the upper layers of its arrival.

Depending on the network layer design decision to support full- or half-duplex communication, the interpretation of “unexpected” differs:

a) with half-duplex, point-to-point communication between two nodes is only possible in one direction at a time;

b) with full-duplex, point-to-point communication between two nodes is possible in both directions at once.

In addition to this network layer design decision, it is necessary to consider the possibility that a reception or transmission from/to a node with the same address information (N_AI) as contained in the received unexpected N_PDU is in progress.

Table 18 defines the network layer behaviour in the case of the reception of an unexpected N_PDU, in consideration of the actual network layer internal status (NWL status) and the design decision to support half- or full-duplex communication. Table 18 only applies if the received N_PDU contains the same N_AI as the reception or transmission that is in progress at the time the N_PDU is received.

If the N_AI of the received N_PDU is different from the segmented message, an on-going reception/transmission shall be continued.

Table 18 — Handling of unexpected arrival of N_PDU with same N_AI as currently being processed

NWL status	Reception of				
	SF N_PDU	FF N_PDU	CF N_PDU	FC N_PDU	Unknown N_PDU
Segmented transmit in progress	Full-duplex: If a reception is in progress, see the corresponding cell below in this table; otherwise, process the SF N_PDU as the start of a new reception.	Full-duplex: If a reception is in progress, see the corresponding cell below in this table; otherwise, process the FF N_PDU as the start of a new reception.	Full-duplex: If a reception is in progress, see the corresponding cell below in this table.	Ignore[a]	Ignore
	Half-duplex: Ignore	Half-duplex: Ignore	Half-duplex: Ignore		
Segmented receive in progress	Terminate the current reception, report an N_USData.indication, with <N_Result> set to N_UNEXP_PDU, to the upper layer, and process the SF N_PDU as the start of a new reception.	Terminate the current reception, report an N_USData.indication, with <N_Result> set to N_UNEXP_PDU, to the upper layer, and process the FF N_PDU as the start of a new reception.	Ignore[b]: If awaited, process the CF N_PDU in the on-going reception and perform the required checks (e.g. SN in right order); otherwise, ignore it.	Full-duplex: If a transmission is in progress, see corresponding cell above in this table.	Ignore
				Half-duplex: Ignore	
Idle[c]	Process the SF N_PDU as the start of a new reception.	Process the FF N_PDU as the start of a new reception.	Ignore	Ignore	Ignore

a FC parameter error handling is described separately in 8.5.5.3 and 8.5.5.6.

b Handling of an unexpected SN is described separately in 8.5.4.3.

c Neither a segmented transmission nor a segmented reception is in progress. This status “Idle” only describes the network layer itself and is not an indication of the availability of the layers above, which might be busy and thus might not be able to accept a new (SF) request or provide a diagnostic buffer for the data of a multi-frame request (FF).

8.7.4 Wait frame error handling

If the receiver has transmitted N_WFTmax FlowControl wait network protocol data units (FC N_PDU WAIT) in a row and, following this, it cannot meet the performance requirement for the transmission of a FlowControl ContinueToSend network protocol data unit (FC N_PDU CTS), then 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.

The sender of the message is informed about the aborted message reception via an N_USData.confirm with <N_Result> set to N_TIMEOUT_Bs. (Because of the missing FlowControl N_PDU from the receiver, an N_Bs timeout occurs in the sender.)

8.8 Interleaving of messages

The network layer protocol shall be capable of carrying out parallel transmission of different messages that are not mapped onto the same N_AI. This is necessary to ensure that the receiving peer is able to reassemble in a consistent manner the received network protocol data units. This scheme enables, for example, gateway operation that needs to handle different message transmissions concurrently across distinct sub-networks.

9 Data link layer usage

9.1 Data link layer service parameters

The following data link layer service parameters are defined in ISO 11898-1:

— <Data>: CAN frame data

— <DLC>: data length code

— <Identifier>: CAN identifier

— <Transfer_Status>: status of a transmission

9.2 Data link layer interface services

9.2.1 L_Data.request

The service primitive requests transmission of <Data> that shall be mapped within specific attributes of the data link protocol data unit selected by means of <Identifier>.

The <Identifier> shall provide reference to the specific addressing format used to transmit <Data>:

L_Data.request (
<Identifier>
<DLC>
<Data>
)

9.2.2 L_Data.confirm

The service primitive confirms the completion of an L_Data.request service for a specific <Identifier>.

The parameter <Transfer_Status> provides the status of the service request:

L_Data.confirm (
<Identifier>
<Transfer_Status>
)

9.2.3 L_Data.indication

The service primitive indicates a data link layer event to the adjacent upper layer and delivers <Data> identified by <Identifier>:

L_Data.indication (
<Identifier>
<DLC>
<Data>
)

9.3 Mapping of the N_PDU fields

9.3.1 Addressing formats

The exchange of network layer data is supported by three addressing formats: normal, extended and mixed addressing. Each addressing format requires a different number of CAN frame data bytes to encapsulate the addressing information associated with the data to be exchanged. Consequently, the number of data bytes transported within a single CAN frame depends on the type of addressing format chosen.

Subclauses 9.3.2 to 9.3.5 specify the mapping mechanisms for each addressing format, based on the data link layer services and service parameters defined in ISO 11898-1.

9.3.2 Normal addressing

For each combination of N_SA, N_TA, N_TAtype and Mtype, a unique CAN identifier is assigned. N_PCI and N_Data is placed in the CAN frame data field. See Table 19.

Table 19 — Mapping of N_PDU parameters into CAN frame — Normal addressing, N_TAtype = physical

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						
FirstFrame (FF)	N_AI	N_PCI		N_Data					
ConsecutiveFrame (CF)	N_AI	N_PCI	N_Data						
FlowControl (FC)	N_AI	N_PCI			N/A				

Table 20 defines the mapping of N_PDU parameters into CAN frame where the addressing format is normal and N_TAtype is functional.

Table 20 — Mapping of N_PDU parameters into CAN frame — Normal addressing, N_TAtype = functional

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						

9.3.3 Normal fixed addressing

Normal fixed addressing is a subformat of normal addressing in which the mapping of the address information into the CAN identifier is further defined. In the general case of normal addressing, described above, the correspondence between N_AI and the CAN identifier is left open.

For normal fixed addressing, only 29 bit CAN identifiers are allowed. Tables 21 and 22 define the mapping of the address information (N_AI) into the CAN identifier, depending on the target address type (N_TAtype). N_PCI and N_Data are placed in the CAN frame data field.

Table 21 defines normal fixed addressing where N_TAtype is physical.

Table 21 — Normal fixed addressing, N_TAtype = physical

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	2	3	4	5	6	7	8
SingleFrame (SF)	110 (bin.)	0	0	218 (dec.)	N_TA	N_SA	N_PCI	N_Data						
FirstFrame (FF)	110 (bin.)	0	0	218 (dec.)	N_TA	N_SA	N_PCI		N_Data					
ConsecutiveFrame (CF)	110 (bin.)	0	0	218 (dec.)	N_TA	N_SA	N_PCI	N_Data						
FlowControl (FC)	110 (bin.)	0	0	218 (dec.)	N_TA	N_SA	N_PCI			N/A				

Table 22 defines normal fixed addressing where N_TAtype is functional.

Table 22 — Normal fixed addressing, N_TAtype = functional

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	2	3	4	5	6	7	8
SingleFrame (SF)	110 (bin.)	0	0	219 (dec.)	N_TA	N_SA	N_PCI	N_Data						

9.3.4 Extended addressing

For each combination of N_SA, N_TAtype and Mtype, a unique CAN identifier is assigned. N_TA is placed in the first data byte of the CAN frame data field. N_PCI and N_Data are placed in the remaining bytes of the CAN frame data field.

Table 23 defines the mapping of N_PDU parameters into CAN frame where the addressing format is extended and N_TAtype is physical.

Table 23 — Mapping of N_PDU parameters into CAN frame — Extended addressing, N_TAtype = physical

N_PDU type	CAN identifier	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
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			N/A			

Table 24 defines the mapping of N_PDU parameters into CAN frame where the addressing format is extended and N_TAtype is functional.

Table 24 — Mapping of N_PDU parameters into CAN frame — Extended addressing, N_TAtype = functional

N_PDU type	CAN identifier	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
SingleFrame (SF)	N_AI, except N_TA	N_TA	N_PCI	N_Data					

9.3.5 Mixed addressing

9.3.5.1 29 bit CAN identifier

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

Table 25 and Table 26 define the mapping of the address information (N_AI) into the 29 bit CAN identifier scheme and the first CAN frame data byte, depending on the target address type (N_TAtype). N_PCI and N_Data are placed in the remaining bytes of the CAN frame data field.

Table 25 — Mixed addressing with 29 bit CAN identifier, N_TAtype = physical

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	2	3	4	5	6	7	8
SingleFrame (SF)	110 (bin.)	0	0	206 (dec.)	N_TA	N_SA	N_AE	N_PCI	N_Data					
FirstFrame (FF)	110 (bin.)	0	0	206 (dec.)	N_TA	N_SA	N_AE	N_PCI		N_Data				
ConsecutiveFrame (CF)	110 (bin.)	0	0	206 (dec.)	N_TA	N_SA	N_AE	N_PCI	N_Data					
FlowControl (FC)	110 (bin.)	0	0	206 (dec.)	N_TA	N_SA	N_AE	N_PCI			N/A			

Table 26 — Mixed addressing with 29 bit CAN identifier, N_TAtype = functional

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	2	3	4	5	6	7	8
SingleFrame (SF)	110 (bin.)	0	0	205 (dec.)	N_TA	N_SA	N_AE	N_PCI	N_Data					

9.3.5.2 11 bit CAN identifier

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

Table 27 and Table 28 define the mapping of the address information (N_AI) into the 11 bit CAN identifier scheme. For each combination of N_SA, N_TA and N_TAtype, a unique CAN identifier is assigned. N_AE is placed in the first data byte of the CAN frame data field. N_PCI and N_Data are placed in the remaining bytes of the CAN frame data field.

Table 27 — Mixed addressing with 11 bit CAN identifier, N_TAtype = physical

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_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			N/A			

Table 28 — Mixed addressing with 11 bit CAN identifier, N_TAtype = functional

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_AE	N_PCI	N_Data					

9.4 CAN frame data length code (DLC)

9.4.1 DLC parameter

The DLC parameter specifies the number of data bytes transmitted in a CAN frame. This part of ISO 15765 does not specify any requirements concerning the length of the data field in a CAN frame other than those implied by the size of the network layer protocol data units.

An application that implements the network layer as defined in this part of ISO 15765 might either pad all CAN frames to their full length (see 9.4.2.1) or optimize the DLC to the applicable length of the network layer protocol data unit (see 9.4.2.2).

9.4.2 CAN frame data

9.4.2.1 CAN frame data padding

If this solution is used, the DLC is always set to 8, even if the N_PDU to be transmitted is shorter than 8 bytes. The sender has to pad any unused bytes in the frame. In particular, this can be the case for an SF, FC frame or the last CF of a segmented message.

The DLC parameter of the CAN frame is set by the sender and read by the receiver to determine the number of data bytes per CAN frame to be processed by the network layer. The 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.

9.4.2.2 CAN frame data optimization

If this solution is used, the DLC does not always need to be 8. If the N_PDU to be transmitted is shorter than 8 bytes, then the sender may optimize the CAN bus load by shortening the CAN frame data to contain only the number of bytes occupied by the N_PDU (no padding of unused data bytes). CAN frame data optimization can only be used for an SF, FC frame or the last CF of a segmented message.

The DLC parameter of the CAN frame is set by the sender and read by the receiver to determine the number of data bytes per CAN frame to be processed by the network layer. The DLC parameter cannot be used to determine the message length; this information shall be extracted from the N_PCI information in the beginning of a message.

9.4.3 Data length code (DLC) error handling

Depending on the N_PCI value, the network layer can calculate the smallest expected value for the CAN DLC parameter in a received CAN frame.

Reception of a CAN frame with a DLC value smaller than expected (less than 8 for applications which pad the CAN frames or smaller than implied by the size of the network protocol data unit for implementations using data optimization) shall be ignored by the network layer without any further action.

Annex A
(normative)

Use of normal fixed and mixed addressing with data link layer according to SAE J1939

A.1 Overview

This annex describes how to map address information parameters, N_AI, into the CAN frame when a data link layer according to SAE J1939 is used.

A.2 Rules

A.2.1 Normal fixed addressing

Table A.1 shows the mapping of address information parameters, N_AI, into the CAN frame when Network Target Address type, N_TAtype, physical addressing is used.

Table A.1 — Normal addressing, physical addressed messages

SAE J1939 name	P	R	DP	PF	PS	SA	Data field
Bits	3	1	1	8	8	8	64
Content	default 110 (bin.)	0	0	218 (dec.)	N_TA	N_SA	N_PCI, N_Data
CAN Id bits	28 – 26	25	24	23 – 16	15 – 8	7 – 0	—
CAN data byte	—	—	—	—	—	—	1 – 8
CAN field	Identifier						Data
NOTE See A.2.3 to A.2.8 for definitions of the abbreviated terms used in this table.							

Table A.2 shows the mapping of address information parameters, N_AI, into the CAN frame when Network Target Address type, N_TAtype, functional addressing is used.

Table A.2 — Normal addressing, functional addressed messages

SAE J1939 name	P	R	DP	PF	PS	SA	Data field
Bits	3	1	1	8	8	8	64
Content	default 110 (bin.)	0	0	219 (dec.)	N_TA	N_SA	N_PCI, N_Data
CAN Id bits	28 – 26	25	24	23 – 16	15 – 8	7 – 0	—
CAN data byte	—	—	—	—	—	—	1 – 8
CAN field	Identifier						Data
NOTE See A.2.3 to A.2.8 for definitions of the abbreviated terms used in this table.							

A.2.2 Mixed addressing

Table A.3 shows the mapping of address information parameters, N_AI, into the CAN frame when the Network Target Address type, N_TAtype, physical addressing is used.

Table A.3 — Mixed addressing, physical addressed messages

SAE J1939 name	P	R	DP	PF	PS	SA	Data field	
Bits	3	1	1	8	8	8	8	56
Content	default 110 (bin.)	0	0	206 (dec.)	N_TA	N_SA	N_AE	N_PCI, N_Data
CAN Id bits	28 – 26	25	24	23 – 16	15 – 8	7 – 0		—
CAN data byte	—	—	—	—	—	—	1	2 – 8
CAN field	Identifier						Data	
NOTE See A.2.3 to A.2.8 for definitions of the abbreviated terms used in this table.								

Table A.4 shows the mapping of address information parameters, N_AI, into the CAN frame when Network Target Address type, N_TAtype, functional addressing is used.

Table A.4 — Mixed addressing, functional addressed messages

SAE J1939 name	P	R	DP	PF	PS	SA	Data field	
Bits	3	1	1	8	8	8	8	56
Content	default 110 (bin.)	0	0	205 (dec.)	N_TA	N_SA	N_AE	N_PCI, N_Data
CAN Id bits	28 – 26	25	24	23 – 16	15 – 8	7 – 0		—
CAN data byte	—	—	—	—	—	—	1	2 – 8
CAN field	Identifier						Data	
NOTE See A.2.3 to A.2.8 for definitions of the abbreviated terms used in this table.								

A.2.3 Priority (P)

The priority is user defined with a default value of six.

The 3 bit priority field is used to optimize message latency for transmission onto the CAN bus only. The priority field should be masked off by the receiver (ignored). The priority of any CAN message can be set from highest, 0 (000 bin.), to lowest, 7 (111 bin.).

A.2.4 Reserved bit (R)

The reserved bit shall be set to “0”.

A.2.5 Data page (DP)

The data page bit shall be set to “0”.

A.2.6 Protocol data unit format (PF)

The format is of the type PDU1, “destination specific”.

Diagnostic messages shall use the following parameter group numbers (PGN).

— Mixed addressing: 52480 (dec.) for N_TAtype = functional, which gives PF = 205 (dec.).

— Mixed addressing: 52736 (dec.) for N_Tatype = physical, which gives PF = 206 (dec.).

— Normal fixed addressing: 55808 (dec.) for N_TAtype = physical, which gives PF = 218 (dec.).

— Normal fixed addressing: 56064 (dec.) for N_TAtype = functional, which gives PF = 219 (dec.).

A.2.7 PDU-specific (PS)

The PDU-specific field shall contain the target address (destination address), N_TA.

A.2.8 Source address (SA)

The SA field shall contain the source address, N_SA.

A.2.9 Update rate

Update rate is defined according to user requirements.

A.2.10 Data length

Data length shall be 8 bytes.

Annex B
(normative)

Reserved CAN Ids

The purpose of this annex is to reserve CAN Ids out of the 11 bit CAN Id range for future use in International Standards.

Table B.1 defines the reserved CAN Id ranges.

Table B.1 — ISO-reserved CAN Ids

CAN identifier	Description
0x7F4 – 0x7F6	Reserved by ISO 15765-2
0x7FA – 0x7FB	Reserved by ISO 15765-2

Bibliography

[1] ISO/IEC 10731, *Information technology — Open Systems Interconnection — Basic Reference Model — Conventions for the definition of OSI services*

[2] ISO 11898 (all parts), *Road vehicles — Controller area network (CAN)*

[3] ISO 14229-1, *Road vehicles — Unified diagnostic services (UDS) — Part 1: Specification and requirements*

[4] ISO 14229-2, *Road vehicles — Unified diagnostic services (UDS) — Session layer services*

[5] ISO 14229-3, *Road vehicles — Unified diagnostic services (UDS) — Part 3: Unified diagnostic services on CAN implementation (UDSonCAN)*

[6] ISO 15031-2, *Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics — Part 2: Guidance on terms, definitions, abbreviations and acronyms*

[7] ISO 15031-5, *Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics — Part 5: Emissions-related diagnostic services*

[8] ISO 15031-6, *Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics — Part 6: Diagnostic trouble code definitions*

[9] ISO 15765-1, *Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) — Part 1: General information and use case definition*

[10] ISO 15765-4, *Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) — Part 4: Requirements for emissions-related systems*

[11] ISO 27145-2, *Road vehicles — Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements — Part 2: Common data dictionary*

[12] ISO 27145-3, *Road vehicles — Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements — Part 3: Common message dictionary*

[13] ISO 27145-4, *Road vehicles — Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements — Part 4: Connection between vehicle and test equipment*

[14] SAE J1930, *Electrical/Electronic Systems Diagnostic Terms, Definitions, Abbreviations and Acronyms*

[15] SAE J1939:2011, *Serial Control and Communications Heavy Duty Vehicle Network — Top Level Document*

[16] SAE J1939-73:2010, *Application layer — Diagnostics*

[17] SAE J1979, *E/E Diagnostic Test Modes*

[18] SAE J2012, *Diagnostic Trouble Code Definitions*

This page is intentionally blank.

This page is intentionally blank.

ICS 43.040.15

Price based on 39 pages