BS ISO 21806-6:2020

BSI Standards Publication

# Road vehicles — Media Oriented Systems Transport (MOST)

Part 6: Data link layer

Documents
Document Sharing Hub

Full document details: BS ISO 21806-6:2020

# National foreword

This British Standard is the UK implementation of ISO 21806-6:2020.

The UK participation in its preparation was entrusted to Technical Committee AUE/16, Data Communication (Road Vehicles).

A list of organizations represented on this committee can be obtained on request to its committee manager.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2020
Published by BSI Standards Limited 2020

ISBN 978 0 539 03046 4

ICS 43.040.15

**Compliance with a British Standard cannot confer immunity from legal obligations.**

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 September 2020.

**Amendments/corrigenda issued since publication**

| Date | Text affected |
|---|---|

Documents
Document Sharing Hub

Full document details: BS ISO 21806-6:2020

INTERNATIONAL STANDARD

# ISO 21806-6

First edition
2020-09

# Road vehicles — Media Oriented Systems Transport (MOST) —

## Part 6: Data link layer

*Véhicules routiers — Système de transport axé sur les médias — Partie 6: Couche de liaison de données*

Reference number
ISO 21806-6:2020(E)

AVA Documents
Document Sharing Hub

© ISO 2020

Full document details: BS ISO 21806-6:2020

**COPYRIGHT PROTECTED DOCUMENT**

© ISO 2020

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

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

Published in Switzerland



© ISO 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO 21806-6:2020

# Contents

Page

**Foreword** ... **vi**
**Introduction** ... **vii**
**1 Scope** ... **1**
**2 Normative references** ... **1**
**3 Terms and definitions** ... **1**
**4 Symbols and abbreviated terms** ... **3**
4.1 Symbols ... 3
4.2 Abbreviated terms ... 3
**5 Conventions** ... **4**
**6 DLL — Service interface to upper layers** ... **4**
6.1 DLL — Overview ... 4
6.2 DLL — Data type definitions ... 4
6.3 DLL — Parameters ... 5
6.3.1 DLL — Parameters – DLL to TL/NL ... 5
6.3.2 DLL — Parameters – TL/NL to DLL ... 6
6.3.3 DLL — Parameters – DLL to TL/NL and TL/NL to DLL ... 8
6.4 DLL — Event indications and action requests ... 10
6.4.1 DLL — L_EVENT.INDICATE ... 10
6.4.2 DLL — L_NODE_POSITION.INDICATE ... 10
6.4.3 DLL — L_MAXIMUM_NODE_POSITION.INDICATE ... 10
6.4.4 DLL — L_ACTION.REQUEST ... 10
6.4.5 DLL — L_NETWORK_STARTUP.REQUEST ... 11
6.4.6 DLL — L_SET_GROUP_ADDRESS.REQUEST ... 11
6.4.7 DLL — L_SET_NODE_ADDRESS.REQUEST ... 11
6.4.8 DLL — L_SET_EUI_48.REQUEST ... 11
6.4.9 DLL — L_SET_TRANSMISSION_ATTRIBUTES.REQUEST ... 12
6.5 DLL — Control Data ... 12
6.5.1 DLL — L_CONTROL_DATA.RECEIVE ... 12
6.5.2 DLL — L_CONTROL_DATA.CONFIRM ... 12
6.5.3 DLL — L_CONTROL_DATA.SEND ... 13
6.6 DLL — Packet data ... 13
6.6.1 DLL — 16-bit addressing ... 13
6.6.2 DLL — 48-bit addressing ... 14
6.7 DLL — Streaming data ... 15
6.7.1 DLL — L_ALLOCATE.INDICATE ... 15
6.7.2 DLL — L_DEALLOCATE.INDICATE ... 15
6.7.3 DLL — L_CONNECT.INDICATE ... 16
6.7.4 DLL — L_DISCONNECT.INDICATE ... 16
6.7.5 DLL — L_SOURCE_DROP.INDICATE ... 16
6.7.6 DLL — L_STREAMING_DATA.RECEIVE ... 17
6.7.7 DLL — L_ALLOCATE.REQUEST ... 17
6.7.8 DLL — L_DEALLOCATE.REQUEST ... 17
6.7.9 DLL — L_CONNECT.REQUEST ... 18
6.7.10 DLL — L_DISCONNECT.REQUEST ... 18
6.7.11 DLL — L_STREAMING_DATA.SEND ... 18
**7 DLL — Network frame** ... **18**
7.1 DLL — General ... 18
7.2 DLL — Administrative area ... 20
7.3 DLL — Source data area ... 20
7.4 DLL — Indicators ... 21
**8 DLL — Channels** ... **21**

AVA Documents
Document Sharing Hub

© ISO 2020 – All rights reserved


Full document details: BS ISO 21806-6:2020

8.1 DLL — Allocation channel 21
8.1.1 DLL — General 21
8.1.2 DLL — Allocation frame structure 21
8.1.3 DLL — Common allocation channel related subjects 23
8.1.4 DLL — Allocation channel related subjects for the TimingMaster 24
8.1.5 DLL — Allocation channel related subjects for a TimingSlave 25
8.1.6 DLL — De-allocating 27
8.1.7 DLL — Source-drop recognition 27
8.1.8 DLL — Error handling 28
8.2 DLL — Control channel 30
8.2.1 DLL — General 30
8.2.2 DLL — Control frame structure 31
8.3 DLL — Protected system channel 31
8.3.1 DLL — General 31
8.3.2 DLL — Protected system frame structure 32
8.4 DLL — Timestamp channel 34
8.4.1 DLL — General 34
8.4.2 DLL — Timestamp frame structure 34
8.4.3 DLL — Behaviour 35
8.5 DLL — Packet channel 35
8.5.1 DLL — General 35
8.5.2 DLL — Packet frame structure 36
8.5.3 DLL — Ethernet data frame structure 37
8.5.4 DLL — Short packet frame or short Ethernet data frame 38
8.6 DLL — Synchronous channel 38
8.6.1 DLL — General 38
8.6.2 DLL — Synchronous frame structure 38
8.7 DLL — Isochronous channel 38
8.7.1 DLL — General 38
8.7.2 DLL — Isochronous frame structure 38
8.8 DLL — Channel frame delay 39

**9 DLL — Flow control 41**
9.1 DLL — Pre-emptive acknowledge byte 41
9.2 DLL — Early ending 41
9.3 DLL — Low-level retries 42

**10 DLL — Arbitration 42**
10.1 DLL — General 42
10.2 DLL — Load-adaptive arbitration 43
10.2.1 DLL — General 43
10.2.2 DLL — Downstream arbitration 46
10.2.3 DLL — Downstream or upstream arbitration 47
10.2.4 DLL — Conditional upstream arbitration 48
10.3 DLL — Round-robin arbitration 48
10.3.1 DLL — Basics 48
10.3.2 DLL — Ensuring round-robin transmit order 50
10.3.3 DLL — Examples 50

**11 DLL — Addressing 51**
11.1 DLL — General 51
11.2 DLL — 16-bit address types 52
11.2.1 DLL — General 52
11.2.2 DLL — Free-up address 52
11.2.3 DLL — Logical node address 52
11.2.4 DLL — Group address 52
11.2.5 DLL — Blocking broadcast address 53
11.2.6 DLL — Non-blocking broadcast address 54
11.2.7 DLL — Node position address 54
11.2.8 DLL — Debug address 54


© ISO 2020 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21806-6:2020

11.3 DLL — 48-bit address types ... 55

**12 DLL — Cyclic redundancy check (CRC) ... 55**

12.1 DLL — General ... 55
12.2 DLL — 4-bit CRC ... 55
12.3 DLL — 16-bit CRC ... 56
12.4 DLL — 32-bit CRC ... 56
12.5 DLL — CRC usage ... 56
12.6 DLL — CRC acknowledge ... 56

**Annex A** (normative) **MOST network configurations ... 58**

**Annex B** (normative) **Frame indicators ... 65**

**Bibliography ... 68**

© ISO 2020 – All rights reserved



Documents
Document Sharing Hub

Full document details: BS ISO 21806-6:2020

# Foreword

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

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

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

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

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

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

A list of all parts in the ISO 21806 series can be found on the ISO website.

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


© ISO 2020 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21806-6:2020

# Introduction

The Media Oriented Systems Transport (MOST) communication technology was initially developed at the end of the 1990s in order to support complex audio applications in cars. The MOST Cooperation was founded in 1998 with the goal to develop and enable the technology for the automotive industry. Today, MOST[1)] enables the transport of high quality of service (QoS) audio and video together with packet data and real-time control to support modern automotive multimedia and similar applications. MOST is a function-oriented communication technology to network a variety of multimedia devices comprising one or more MOST nodes.

Figure 1 shows a MOST network example.

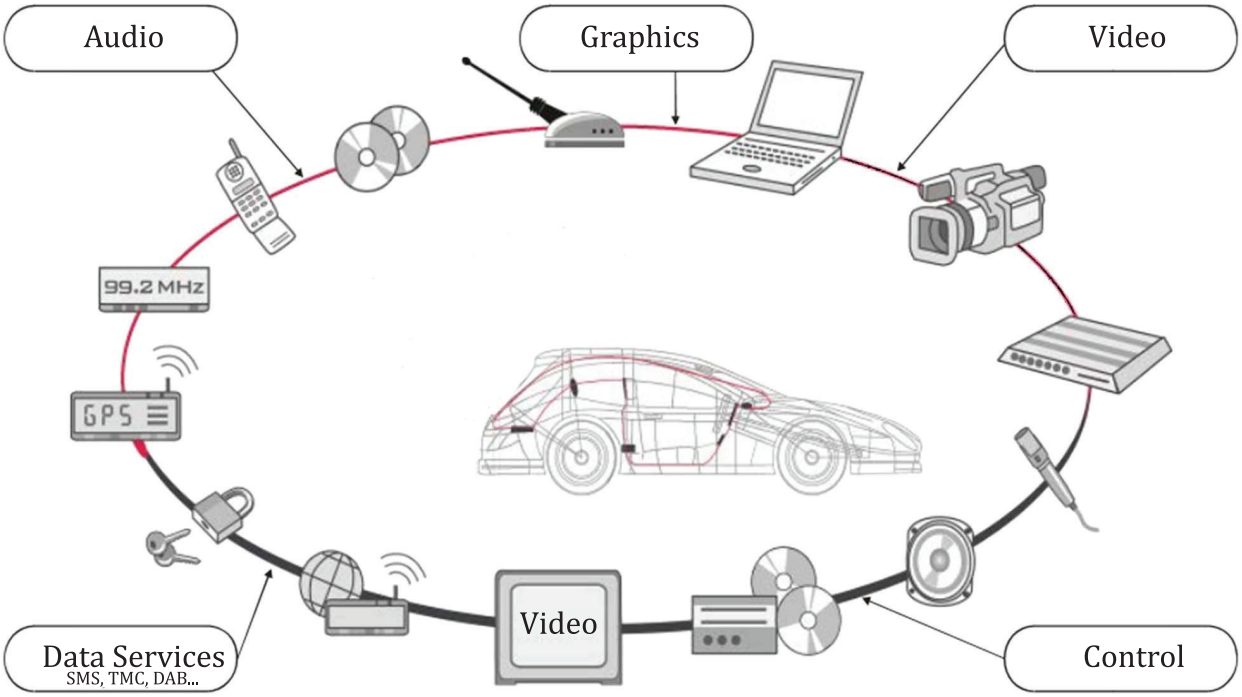


**Figure 1 — MOST network example**

The MOST communication technology provides

— synchronous and isochronous streaming,

— small overhead for administrative communication control,

— a functional and hierarchical system model,

— API standardization through a function block (FBlock) framework,

— free partitioning of functionality to real devices,

— service discovery and notification, and

— flexibly scalable automotive-ready Ethernet communication according to ISO/IEC/IEEE 8802-3[4].

MOST is a synchronous time-division-multiplexing (TDM) network that transports different data types on separate channels at low latency. MOST supports different bit rates and physical layers. The network clock is provided with a continuous data signal.

1) MOST® is the Registered Trademark of Microchip Technology Inc. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO.

© ISO 2020 – All rights reserved


Documents
Document Sharing Hub
Full document details: BS ISO 21806-6:2020

Within the synchronous base data signal, the content of multiple streaming connections and control data is transported. For streaming data connections, bandwidth is reserved to avoid interruptions, collisions, or delays in the transport of the data stream.

MOST specifies mechanisms for sending anisochronous, packet-based data in addition to control data and streaming data. The transmission of packet-based data is separated from the transmission of control data and streaming data. None of them interfere with each other.

A MOST network consists of devices that are connected to one common control channel and packet channel.

In summary, MOST is a network that has mechanisms to transport the various signals and data streams that occur in multimedia and infotainment systems.

The ISO standards maintenance portal (https://standards.iso.org/iso/) provides references to MOST specifications implemented in today's road vehicles because easy access via hyperlinks to these specifications is necessary. It references documents that are normative or informative for the MOST versions 4V0, 3V1, 3V0, and 2V5.

The ISO 21806 series has been established in order to specify requirements and recommendations for implementing the MOST communication technology into multimedia devices and to provide conformance test plans for implementing related test tools and test procedures.

To achieve this, the ISO 21806 series is based on the open systems interconnection (OSI) basic reference model in accordance with ISO/IEC 7498-1[1] and ISO/IEC 10731[2], which structures communication systems into seven layers as shown in Figure 2. Stream transmission applications use a direct stream data interface (transparent) to the data link layer.


© ISO 2020 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21806-6:2020

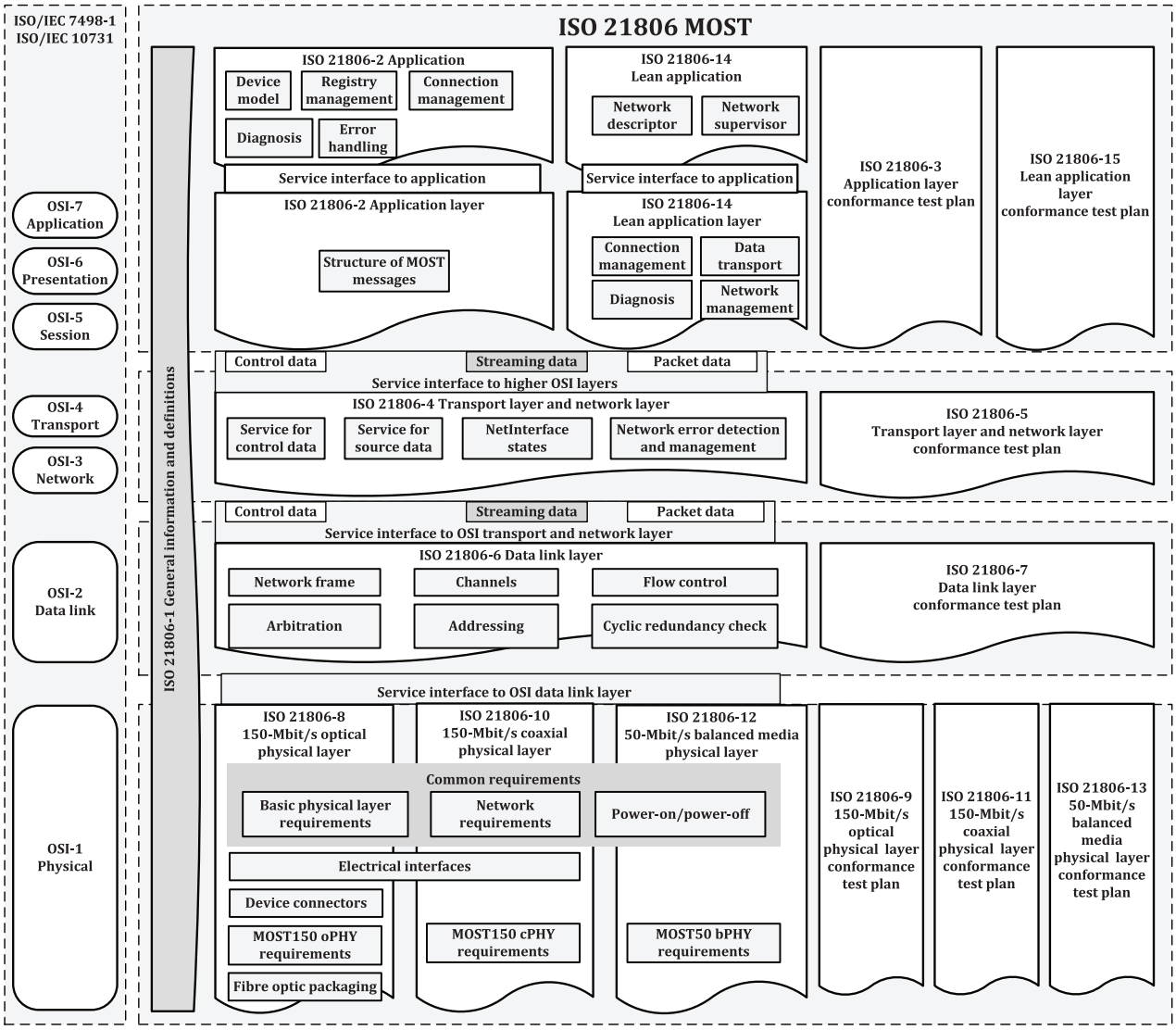


**Figure 2 — The ISO 21806 series reference according to the OSI model**

The International Organization for Standardization (ISO) draws attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO takes no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO. Information may be obtained from the patent database available at www.iso.org/patents.

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

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21806-6:2020

Documents
Document Sharing Hub

# Road vehicles — Media Oriented Systems Transport (MOST) —

## Part 6:
## Data link layer

## 1 Scope

This document specifies technical requirements related to the MOST data link layer functionality.

A MOST network is comprised of two or more nodes connected through a physical layer. The data link layer functionality is provided by each node. On each network, all nodes are synchronised and one node provides the system clock. This node is the TimingMaster, while all other nodes are TimingSlaves. The timing configuration of the node (TimingMaster or TimingSlave) determines the tasks that need to be performed on the data link layer.

The data link layer specifies the following subjects:

— the service interface to the network layer;

— the network frame, its areas and indicators;

— the different network channels;

— the different flow control mechanisms;

— the load-adaptive arbitration and the round-robin arbitration;

— the different addressing options;

— the different cyclic redundancy checks, their usage and the CRC acknowledge;

— the frame indicators.

## 2 Normative references

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

ISO 21806-1, *Road vehicles — Media Oriented Systems Transport (MOST) — Part 1: General information and definitions*

## 3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 21806-1 and the following apply.

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

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

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

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21806-6:2020

**3.1**
**diagnosis flag**
flag that determines whether diagnosis is active

**3.2**
**END**
indicator for the end of a channel frame

**3.3**
**Ethernet frame**
frame according to ISO/IEC/IEEE 8802-3

**3.4**
**isochronous frame**
frame that consists of isochronous data

**3.5**
**new allocation flag**
flag that is set for newly allocated bytes

**3.6**
**packet frame**
frame that transports packet data with 16-bit addressing

**3.7**
**PREAMBLE**
indicator for the start of the network frame

**3.8**
**protected system channel**
channel that transports network status information

**3.9**
**ring lock flag**
flag that is set when the TimingMaster locks onto the incoming data stream

**3.10**
**START**
indicator for the start of a channel frame

**3.11**
**static master flag**
flag that determines whether the TimingMaster continuously sends network frames

**3.12**
**synchronous frame**
frame that is synchronous to the network clock and consists of unformatted data

**3.13**
**timestamp channel**
channel that is used to transport a CRC-protected timestamp

**3.14**
**WAIT**
indicator that is used for different purposes


© ISO 2020 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21806-6:2020

# 4 Symbols and abbreviated terms

## 4.1 Symbols

| | |
|---|---|
| <...> | range of bits, e.g. bit 7 to bit 0: <7:0> |
| ... | and so on |
| --- | empty cell/undefined |
| $N_{PBC}$ | packet bandwidth control |
| $N_{PBC_max}$ | maximum value of packet bandwidth control |
| $N_{SDBPFREST}$ | remaining number of source data bytes per frame |
| $N_{TNABPF}$ | total number of administrative bytes per frame |
| $N_{TNBPF}$ | total number of bytes per frame |
| $N_{TNSDBPF}$ | total number of source data bytes per frame |

## 4.2 Abbreviated terms

| | |
|---|---|
| alloc | allocation |
| Arb[X] | 8-bit arbitration value |
| ARBVAL | arbitration value |
| CACK | CRC acknowledge |
| CF | channel frame |
| CL | connection label |
| CPos | calculated position of the END indicator |
| CRC | cyclic redundancy check |
| DLL | data link layer |
| LSb | least significant bit |
| MOST | Media Oriented Systems Transport |
| MSb | most significant bit |
| NC | node counter (used in tables and figures) |
| NF | network frame (used in tables and figures) |
| NOFFAD | number of frames for auto deblock |
| PACK | pre-emptive acknowledge |
| PDU | protocol data unit |
| SDBPF | source data bytes per frame |

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21806-6:2020

| | |
|---|---|
| SOAF | start of allocation frame |
| TM | TimingMaster (used in figures) |
| TS | TimingSlave (used in figures) |

# 5 Conventions

This document is based on OSI service conventions as specified in ISO/IEC 10731[2].

# 6 DLL — Service interface to upper layers

## 6.1 DLL — Overview

The DLL service interface defines the abstract interface to the OSI transport layer and network layer (see ISO 21806-4[3]).

Figure 3 shows the service interface to upper layers.

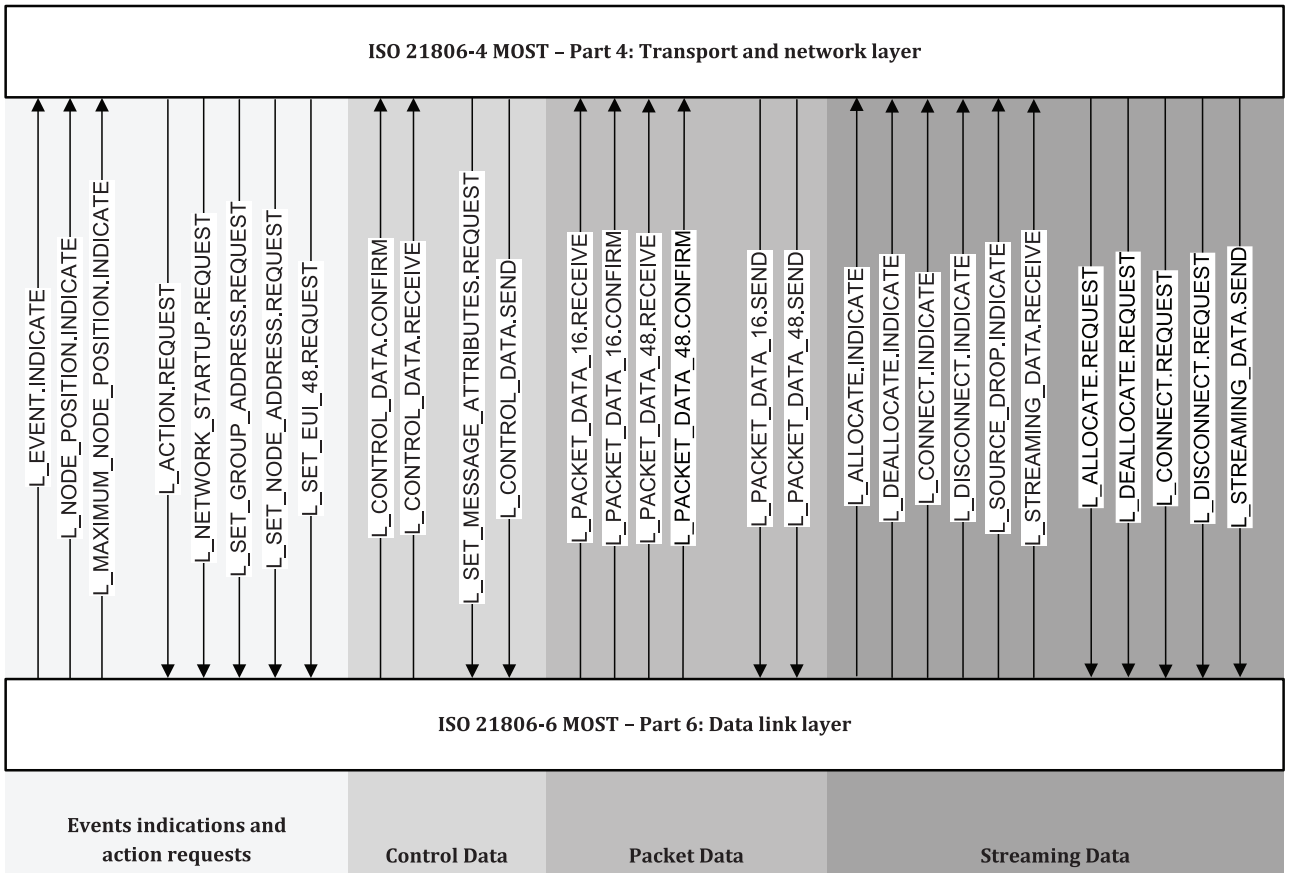


**Figure 3 — Service interface to upper layers**

## 6.2 DLL — Data type definitions

| REQ | 2.1 Service interface – DLL – Data type definitions |
|---|---|
| The data types shall be in accordance to: | |


© ISO 2020 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21806-6:2020

| REQ | 2.1 Service interface – DLL – Data type definitions |
|---|---|
| — `Enum`: 8-bit enumeration;<br>— `Unsigned Byte`: 8-bit unsigned numeric value;<br>— `Unsigned Word`: 16-bit unsigned numeric value;<br>— `EUI-48`: 48-bit address value;<br>— `Byte Array`: sequence of 8-bit aligned data. | |

## 6.3 DLL — Parameters

### 6.3.1 DLL — Parameters – DLL to TL/NL

#### 6.3.1.1 DLL — Overview

Table 1 provides an overview of the parameters that are used in the specified service interface and passed from DLL to TL/NL.

**Table 1 — Parameters passed from DLL to TL/NL**

| Parameter | Data type | Description |
|---|---|---|
| `Network_Event` | `Enum {Unlock, Lock, Lock_Flag, Network_Change_Event, Shutdown_Flag, MOST_Output_Off, Network_Activity}` | An event that is reported to TL/NL. |
| `Node_Position` | `Unsigned Byte` | Node counter |
| `Maximum_Position` | `Unsigned Byte` | Visible nodes |
| `Transmission_Status` | `Enum {Success, Buffer_Full, CRC_Error, Wrong_Target}` | Transmission status that is reported back to the sender. |

#### 6.3.1.2 DLL — Network_Event

The `Network_Event` lists events that are used to notify TL/NL about changes in the DLL, which require no additional information.

| REQ | 2.2 Service interface – DLL – Parameters – DLL to TL/NL – DLL – Network_Event |
|---|---|
| The `Network_Event` parameter shall be of data type `Enum` and shall use the values defined in Table 2. | |

**Table 2 — Network_Event values**

| Enum value | Description |
|---|---|
| `Unlock` | Unlock event occurred |
| `Lock` | Lock reached |
| `Lock_Flag` | Lock flag detected |
| `Network_Change_Event` | The visible nodes value that is distributed by the TimingMaster has changed. Consequently, a network change event (NCE) is generated. |
| `Shutdown_Flag` | Shutdown flag detected |
| `Network_Activity_End` | Network activity ended |
| `Network_Activity` | Network activity detected |

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21806-6:2020

#### 6.3.1.3 DLL — Node_Position

| REQ | 2.3 Service interface – DLL – Parameters – DLL to TL/NL – DLL – Node_Position |
|---|---|
| The `Node_Position` parameter shall be of data type `Unsigned Byte` and shall contain the node counter (NC) value of the respective node. | |

#### 6.3.1.4 DLL — Maximum_Position

| REQ | 2.4 Service interface – DLL – Parameters – DLL to TL/NL – DLL – Maximum_Position |
|---|---|
| The `Maximum_Position` parameter shall be of data type `Unsigned Byte` and shall contain the valid number of nodes visible in the MOST network. | |

#### 6.3.1.5 DLL — Transmission_Status

The `Transmission_Status` lists the possible outcomes of a transmission.

| REQ | 2.5 Service interface – DLL – Parameters – DLL to TL/NL – DLL – Transmission_Status |
|---|---|
| The `Transmission_Status` values shall be of data type `Enum` and shall contain the value specified according to Table 3. | |

**Table 3 — Transmission_Status values**

| Enum value | Description |
|---|---|
| `Success` | The node is ready to receive and a valid CRC is received. |
| `Buffer_Full` | The node is not ready to receive. |
| `CRC_Error` | Incorrect CRC received. |
| `Wrong_Target` | There is no such target. The PACK and CACK bytes remain unaltered. |

### 6.3.2 DLL — Parameters – TL/NL to DLL

#### 6.3.2.1 DLL — Overview

Table 4 provides an overview of the parameters that are used in the specified service interface and passed from TL/NL to DLL.

**Table 4 — Parameters passed from TL/NL to DLL**

| Parameter | Data type | Description |
|---|---|---|
| `Network_Request` | `Enum {`<br>`  cmd_Set_Shutdown_Flag,`<br>`  cmd_Set_Lock_Flag,`<br>`  cmd_Clear_Lock_Flag,`<br>`  cmd_MOST_Output_Off,`<br>`  cmd_MOST_Output_On,`<br>`  cmd_Emergency_Shutdown,`<br>`  cmd_Open_Bypass,`<br>`  cmd_Static_Master,`<br>`  cmd_Ring_Lock`<br>`}` | A request from TL/NL |
| `Network_Startup_Type` | `Enum {`<br>`  TimingMaster,`<br>`  TimingSlave`<br>`}` | Determines how a node starts up. |
| `Number_Of_Retries` | `Unsigned Byte` | The number of low-level retries to perform on a control frame. |
| `Priority` | `Unsigned Byte` | The priority for a control frame |


© ISO 2020 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21806-6:2020

**Table 4** *(continued)*

| Parameter | Data type | Description |
|---|---|---|
| Group_Address | Unsigned Word | A group address |
| Node_Address | Unsigned Word | A logical node address |
| EUI_48 | EUI-48 | A 48-bit address |
| Bandwidth | Unsigned Word | Required bandwidth |

#### 6.3.2.2 DLL — Network_Request

The Network_Request lists actions that may be requested from the DLL. These requests require no additional information.

| REQ | 2.6 Service interface – DLL – Parameters – TL/NL to DLL – DLL – Network_Request |
|---|---|
| The Network_Request values shall be of data type Enum and shall use the value specified according to Table 5. | |

**Table 5 — Network_Request values**

| Enum value | Description |
|---|---|
| cmd_Set_Shutdown_Flag | Setting the shutdown flag requested. By default, the shutdown flag is not set. |
| cmd_Set_Lock_Flag | Setting the lock flag requested. By default, the lock flag is cleared. |
| cmd_Clear_Lock_Flag | Clearing the lock flag requested. By default, the lock flag is cleared. |
| cmd_MOST_Output_Off | Switching off the MOST output requested. By default, the MOST output is off. |
| cmd_MOST_Output_On | Switching on the MOST output requested. By default, the MOST output is off. |
| cmd_Emergency_Shutdown | Emergency shutdown requested. By default, emergency shutdown is not active. |
| cmd_Open_Bypass | Opening the bypass requested. By default, the bypass is closed. |
| cmd_Static_Master | Static master mode requested. By default, the TimingMaster is in standard TimingMaster mode. |
| cmd_Ring_Lock | Ring lock requested. By default, the TimingMaster is not locked. |

#### 6.3.2.3 DLL — Network_Startup_Type

The Network_Startup_Type lists the two different startup types, which can be chosen for the node.

| REQ | 2.7 Service interface – DLL – Parameters – TL/NL to DLL – DLL – Network_Startup_Type |
|---|---|
| The Network_Startup_Type values shall be of data type Enum and shall use the value specified according to Table 6. | |

**Table 6 — Network_Startup_Type values**

| Enum value | Description |
|---|---|
| TimingMaster | For startup, configure the node as TimingMaster. |
| TimingSlave | For startup, configure the node as TimingSlave. |

#### 6.3.2.4 DLL — Number_Of_Retries

| REQ | 2.8 Service interface – DLL – Parameters – TL/NL to DLL – DLL – Number_Of_Retries |
|---|---|
| The Number_Of_Retries parameter shall be of data type Unsigned Byte and shall use the maximum permissible number of low-level retries for a particular transmission. | |

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21806-6:2020

#### 6.3.2.5 DLL — Waiting_Period

| REQ | 2.9 Service interface – DLL – Parameters – TL/NL to DLL – DLL – Waiting_Period |
|---|---|
| The `Waiting_Period` parameter shall be of data type `Unsigned Byte` and shall use the waiting period between low-level retries for a particular transmission. | |

#### 6.3.2.6 DLL — Priority

| REQ | 2.10 Service interface – DLL – Parameters – TL/NL to DLL – DLL – Priority |
|---|---|
| The `Priority` parameter shall be of data type `Unsigned Byte` and shall use the priority for a particular transmission. | |

#### 6.3.2.7 DLL — Group_Address

| REQ | 2.11 Service interface – DLL – Parameters – TL/NL to DLL – DLL – Group_Address |
|---|---|
| The `Group_Address` parameter shall be of data type `Unsigned Word` and shall use the group address of the node. | |

#### 6.3.2.8 DLL — Node_Address

| REQ | 2.12 Service interface – DLL – Parameters – TL/NL to DLL – DLL – Node_Address |
|---|---|
| The `Node_Address` parameter shall be of data type `Unsigned Word` and shall use the logical node address. | |

#### 6.3.2.9 DLL — EUI_48

| REQ | 2.13 Service interface – DLL – Parameters – TL/NL to DLL – DLL – EUI_48 |
|---|---|
| The `EUI_48` parameter shall be of data type `EUI-48` and shall use the MAC address of the node. | |

#### 6.3.2.10 DLL — Bandwidth

| REQ | 2.14 Service interface – DLL – Parameters – TL/NL to DLL – DLL – Bandwidth |
|---|---|
| The `Bandwidth` parameter shall be of data type `Unsigned Word` and shall use the value to request the allocation of the corresponding number of bytes in the network frame. | |

### 6.3.3 DLL — Parameters – DLL to TL/NL and TL/NL to DLL

#### 6.3.3.1 DLL — Overview

Table 7 provides an overview of the parameters that are used in the specified service interface for both directions.

**Table 7 — Parameters used both ways**

| Parameter | Data type | Description |
|---|---|---|
| `Media_Interface_ID` | `Unsigned Word` | An identifier for media data output or input |
| `Length` | `Unsigned Word` | Length of the data field that is used in the same service interface. |
| `Data` | `Byte Array` | A data field, whose length is determined by the `Length` parameter. |
| `Session_ID` | `Unsigned Word` | A session identifier to correlate confirmations to send operations. |
| `Target_Address` | `Unsigned Word` | A 16-bit target address |
| `Source_Address` | `Unsigned Word` | A 16-bit source address |


© ISO 2020 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21806-6:2020

**Table 7** *(continued)*

| Parameter | Data type | Description |
|---|---|---|
| `Destination_MAC_Address` | `EUI-48` | A 48-bit target address |
| `Source_MAC_Address` | `EUI-48` | A 48-bit source address |

#### 6.3.3.2 DLL — Media_Interface_ID

| REQ | 2.15 Service interface – DLL – Parameters – DLL to TL/NL and TL/NL to DLL – DLL – Media_Interface_ID |
|---|---|
| The `Media_Interface_ID` parameter shall be of data type `Unsigned Word` and shall be used to unambiguously identify a media interface as source or target of streaming data. | |

#### 6.3.3.3 DLL — Length

| REQ | 2.16 Service interface – DLL – Parameters – DLL to TL/NL and TL/NL to DLL – DLL – Length |
|---|---|
| The `Length` parameter shall be of data type `Unsigned Word` and shall be used to provide the size of the data field that is used in the same service interface. | |

#### 6.3.3.4 DLL — Data

| REQ | 2.17 Service interface – DLL – Parameters – DLL to TL/NL and TL/NL to DLL – DLL – Data |
|---|---|
| The `Data` parameter shall be of data type `Byte Array` and shall be used as a wrapper for payload that requires no interpretation in the context of the service interface that contains it. | |

#### 6.3.3.5 DLL — Session_ID

| REQ | 2.18 Service interface – DLL – Parameters – DLL to TL/NL and TL/NL to DLL – DLL – Session_ID |
|---|---|
| The `Session_ID` parameter shall be of data type `Unsigned Word` and shall be used to unambiguously identify an instance of a control frame that is passed down. The session ID is used when determining the outcome of a transmission attempt. | |

#### 6.3.3.6 DLL — Target_Address

| REQ | 2.19 Service interface – DLL – Parameters – DLL to TL/NL and TL/NL to DLL – DLL – Target_Address |
|---|---|
| The `Target_Address` parameter shall be of data type `Unsigned Word` and shall be used to fill the target address field for 16-bit addressing for control data and packet data. | |

#### 6.3.3.7 DLL — Source_Address

| REQ | 2.20 Service interface – DLL – Parameters – DLL to TL/NL and TL/NL to DLL – DLL – Source_Address |
|---|---|
| The `Source_Address` parameter shall be of data type `Unsigned Word` and shall be used to fill the source address field for 16-bit addressing for control data and packet data. | |

#### 6.3.3.8 DLL — Destination_MAC_Address

| REQ | 2.21 Service interface – DLL – Parameters – DLL to TL/NL and TL/NL to DLL – DLL – Destination_MAC_Address |
|---|---|
| The `Destination_MAC_Address` parameter shall be of data type `EUI-48` and shall be used to fill the destination address field for 48-bit addressing. | |

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21806-6:2020

#### 6.3.3.9 DLL — Source_MAC_Address

| REQ | 2.22 Service interface – DLL – Parameters – DLL to TL/NL and TL/NL to DLL – DLL – Source_MAC_Address |
|---|---|
| The `Source_MAC_Address` parameter shall be of data type `EUI-48` and shall be used to fill the source address field for 48-bit addressing. | |

## 6.4 DLL — Event indications and action requests

### 6.4.1 DLL — L_EVENT.INDICATE

| REQ | 2.23 Service interface – DLL – Event indications and action requests – DLL – L_EVENT.INDICATE – Structure |
|---|---|
| The `L_EVENT.INDICATE` shall be passed from DLL to TL/NL to indicate that an event has occurred.<br>`L_EVENT.INDICATE` shall have the following structure:<br>`L_EVENT.INDICATE{`<br>`  Network_Event`<br>`}` | |

### 6.4.2 DLL — L_NODE_POSITION.INDICATE

| REQ | 2.24 Service interface – DLL – Event indications and action requests – DLL – L_NODE_POSITION.INDICATE – Structure |
|---|---|
| The `L_NODE_POSITION.INDICATE` shall be passed from DLL to TL/NL to indicate the node position.<br>`L_NODE_POSITION.INDICATE` shall have the following structure:<br>`L_NODE_POSITION.INDICATE{`<br>`  Node_Position`<br>`}` | |

### 6.4.3 DLL — L_MAXIMUM_NODE_POSITION.INDICATE

| REQ | 2.25 Service interface – DLL – Event indications and action requests – DLL – L_MAXIMUM_NODE_POSITION.INDICATE – Structure |
|---|---|
| `L_MAXIMUM_NODE_POSITION.INDICATE` shall be passed from DLL to TL/NL to indicate the maximum node position.<br>`L_MAXIMUM_NODE_POSITION.INDICATE` shall have the following structure:<br>`L_MAXIMUM_NODE_POSITION.INDICATE{`<br>`  Maximum_Position`<br>`}` | |

### 6.4.4 DLL — L_ACTION.REQUEST

| REQ | 2.26 Service interface – DLL – Event indications and action requests – DLL – L_ACTION.REQUEST – Structure |
|---|---|
| `L_ACTION.REQUEST` shall be passed from TL/NL to DLL to trigger the execution of a request.<br>`L_ACTION.REQUEST` shall be specified according to the following structure:<br>`L_ACTION.REQUEST {`<br>`  Network_Request`<br>`}` | |


© ISO 2020 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21806-6:2020

### 6.4.5 DLL — L_NETWORK_STARTUP.REQUEST

| REQ | 2.27 Service interface – DLL – Event indications and action requests – DLL – L_NETWORK_STARTUP.REQUEST – Structure |
|---|---|
| `L_NETWORK_STARTUP.REQUEST` shall be passed from TL/NL to DLL to trigger the network start up.<br>`L_NETWORK_STARTUP.REQUEST` shall have the following structure:<br>`L_NETWORK_STARTUP.REQUEST{`<br>`  Network_Startup_Type`<br>`}` | |

### 6.4.6 DLL — L_SET_GROUP_ADDRESS.REQUEST

| REQ | 2.28 Service interface – DLL – Event indications and action requests – Structure |
|---|---|
| `L_SET_GROUP_ADDRESS.REQUEST` shall be passed from TL/NL to DLL to set the group address of the node.<br>`L_SET_GROUP_ADDRESS.REQUEST` shall have the following structure:<br>`L_SET_GROUP_ADDRESS.REQUEST{`<br>`  Group_Address`<br>`}` | |

### 6.4.7 DLL — L_SET_NODE_ADDRESS.REQUEST

| REQ | 2.29 Service interface – DLL – Event indications and action requests – DLL – L_SET_NODE_ADDRESS.REQUEST – Structure |
|---|---|
| `L_SET_NODE_ADDRESS.REQUEST` shall be passed from TL/NL to DLL to set the logical node address.<br>`L_SET_NODE_ADDRESS.REQUEST` shall have the following structure:<br>`L_SET_NODE_ADDRESS.REQUEST{`<br>`  Node_Address`<br>`}` | |

### 6.4.8 DLL — L_SET_EUI_48.REQUEST

| REQ | 2.30 Service interface – DLL – Event indications and action requests – DLL – L_SET_EUI_48.REQUEST – Structure |
|---|---|
| `L_SET_EUI_48.REQUEST` shall be passed from TL/NL to DLL to set the EUI-48 of the node.<br>`L_SET_EUI_48.REQUEST` shall have the following structure:<br>`L_SET_EUI_48.REQUEST{`<br>`  EUI_48`<br>`}` | |

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21806-6:2020

### 6.4.9 DLL — L_SET_TRANSMISSION_ATTRIBUTES.REQUEST

| REQ | 2.31 Service interface – DLL – Event indications and action requests – DLL – L_SET_TRANSMISSION_ATTRIBUTES.REQUEST – Structure |
|---|---|

`L_SET_TRANSMISSION_ATTRIBUTES.REQUEST` shall be passed from TL/NL to DLL to set the attributes for one transmission.

`L_SET_TRANSMISSION_ATTRIBUTES.REQUEST` shall have the following structure:

```
L_SET_TRANSMISSION_ATTRIBUTES.REQUEST{
    Session_ID
    Number_Of_Retries
    Waiting_Period
    Priority
}
```

## 6.5 DLL — Control Data

### 6.5.1 DLL — L_CONTROL_DATA.RECEIVE

| REQ | 2.32 Service interface – DLL – L_CONTROL_DATA.RECEIVE – Structure |
|---|---|

`L_CONTROL_DATA.RECEIVE` shall be passed from DLL to TL/NL to receive control data.

`L_CONTROL_DATA.RECEIVE` shall have the following structure:

```
L_CONTROL_DATA.RECEIVE{
    Length
    Target_Address
    Source_Address
    Data
}
```

### 6.5.2 DLL — L_CONTROL_DATA.CONFIRM

| REQ | 2.33 Service interface – DLL – L_CONTROL_DATA.CONFIRM – Structure |
|---|---|

`L_CONTROL_DATA.CONFIRM` shall be passed from DLL to TL/NL to provide the transmission status for the control data sent.

`L_CONTROL_DATA.CONFIRM` shall have the following structure:

```
L_CONTROL_DATA.CONFIRM{
    Session_ID
    Transmission_Status
}
```

NOTE It is created after successful transmission or after all low-level retries have expired.


© ISO 2020 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21806-6:2020

### 6.5.3 DLL — L_CONTROL_DATA.SEND

| REQ | 2.34 Service interface – DLL – L_CONTROL_DATA.SEND – Structure |
|---|---|
| `L_CONTROL_DATA.SEND` shall be passed from TL/NL to DLL to send control data.<br>`L_CONTROL_DATA.SEND` shall have the following structure:<br>`L_CONTROL_DATA.SEND{`<br>`  Length`<br>`  Session_ID`<br>`  Target_Address`<br>`  Data`<br>`}` | |

## 6.6 DLL — Packet data

### 6.6.1 DLL — 16-bit addressing

#### 6.6.1.1 DLL — L_PACKET_DATA_16.RECEIVE

| REQ | 2.35 Service interface – DLL – L_PACKET_DATA_16.RECEIVE – Structure |
|---|---|
| `L_PACKET_DATA_16.RECEIVE` shall be passed from DLL to TL/NL to receive packet data with 16-bit addressing.<br>`L_PACKET_DATA_16.RECEIVE` shall have the following structure:<br>`L_PACKET_DATA_16.RECEIVE{`<br>`  Length`<br>`  Target_Address`<br>`  Source_Address`<br>`  Data`<br>`}` | |

#### 6.6.1.2 DLL — L_PACKET_DATA_16.CONFIRM

| REQ | 2.36 Service interface – DLL – L_PACKET_DATA_16.CONFIRM – Structure |
|---|---|
| `L_PACKET_DATA_16.CONFIRM` shall be passed from DLL to TL/NL to provide the transmission status for packet data that is sent with 16-bit addressing.<br>`L_PACKET_DATA_16.CONFIRM` shall have the following structure:<br>`L_PACKET_DATA_16.CONFIRM{`<br>`  Session_ID`<br>`  Transmission_Status`<br>`}` | |

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21806-6:2020