INTERNATIONAL STANDARD

ISO 20794-3

First edition
2020-02

Road vehicles — Clock extension peripheral interface (CXPI) —

Part 3:
Transport and network layer

Véhicules routiers — Interface du périphérique d'extension d'horloge (CXPI) —

Partie 3: Transport et couche réseau

Reference number
ISO 20794-3:2020(E)

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

Published in Switzerland

Contents

Page

Foreword

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

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

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

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

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

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

A list of all parts in the ISO 20794 series can be found on the ISO website.

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

Introduction

ISO 20794 (all parts) specifies the application (partly), application layer, transport layer, network layer, data link layer, and physical layer requirements of an in-vehicle network called "clock extension peripheral interface (CXPI)".

CXPI is an automotive low-speed single-wire network. It is an enabler for reducing vehicle weight and fuel consumption by reducing wire counts to simple devices like switches and sensors.

CXPI serves as and is designed for automotive control applications, for example door control group, light switch, and HVAC (Heating Ventilation and Air Conditioning) systems.

The CXPI services, protocols, and their key characteristics are specified in different parts according to the OSI layers.

- Application and application layer
  - application measurement and control data communication to exchange information between applications in different nodes based on message communication;
  - wake-up and sleep functionality;
  - two kinds of communication methods can be selected at system design by each node:
    i) the event-triggered method, which supports application measurement- and control-based (event-driven) slave node communication, and

    ii) the polling method, which supports slave node communication based on a periodic master schedule;
  - performs error detection and reports the result to the application;
  - application error management.
- Transport layer and network layer
  - transforms a message into a single packet;
  - adds protocol control information for diagnostic and node configuration into each packet;
  - adds packet identifier for diagnostic and node configuration into each packet;
  - performs error detection and reports the result to higher OSI layers.
- Data link layer and physical layer
  - provides long and short data frames;
  - adds a frame identifier into the frame;
  - adds frame information into the frame;
  - adds a cyclic redundancy check into the frame;
  - performs byte-wise arbitration and reports the arbitration result to higher OSI layers;
  - performs frame type detection in reception function;
  - performs error detection and reports the result to higher OSI layers.
  - performs Carrier Sense Multiple Access (CSMA);
  - performs Collision Resolution (CR);

- generates a clock, which is transmitted with each bit to synchronise the connected nodes on the CXPI network;
- supports bit rates up to 20 kbit/s.

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

Figure 1 illustrates an overview of communication frameworks beyond the scope of this document including related standards:

- vehicle normal communication framework, which is composed of ISO 20794-2, and ISO 20794-5;
- vehicle diagnostic communication framework, which is composed of ISO 14229-1, ISO 14229-2[3], and ISO 14229-8[4];
- presentation layer standards, e.g. vehicle manufacturer specific or ISO 22901-1 ODX[7];
- lower OSI layers framework, which is composed of ISO 20794-3, ISO 20794-4, ISO 20794-6, and ISO 20794-7 conformance testing.

ISO 20794 (all parts) and ISO 14229-8[4] are based on the conventions specified in the OSI Service Conventions (ISO/IEC 10731)[1] as they apply for all layers and the diagnostic services.

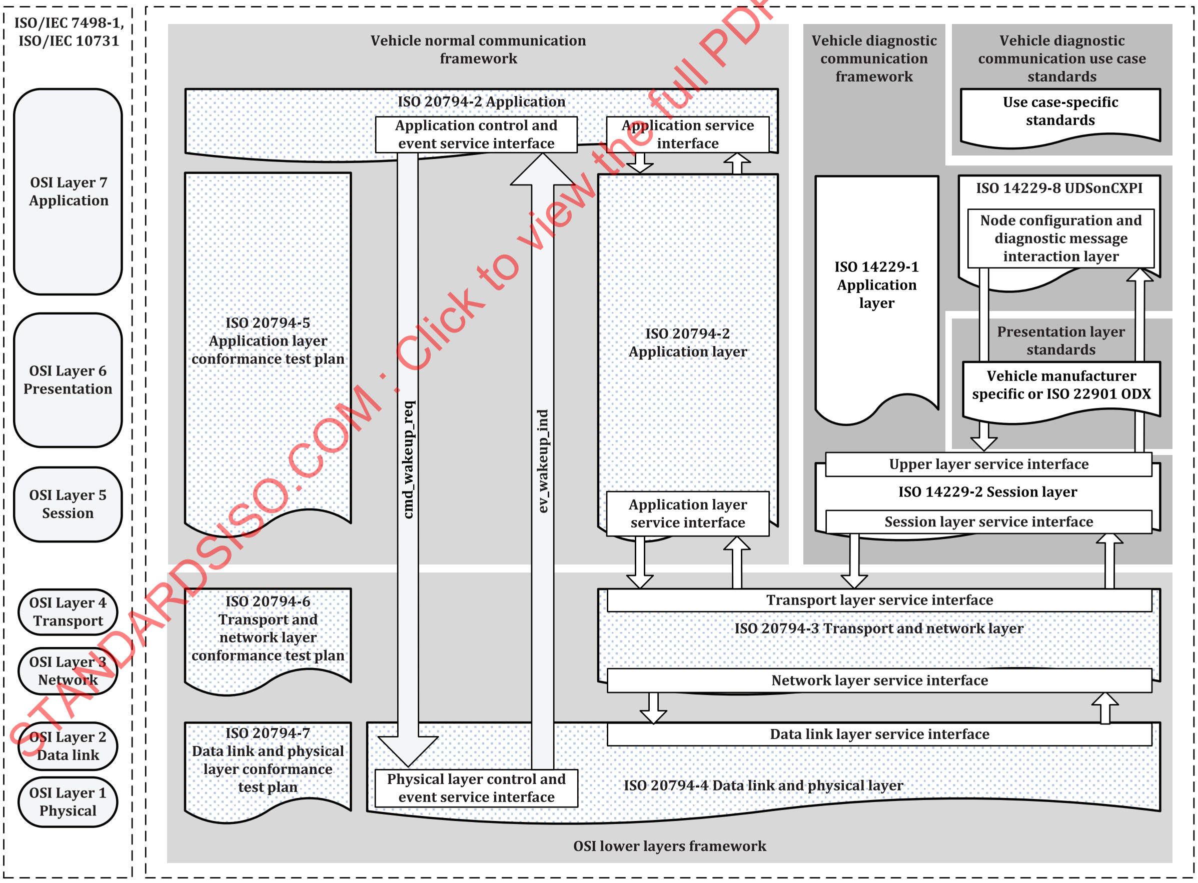

ISO/IEC 7498-1,
ISO/IEC 10731
OSI Layer 7 Application
OSI Layer 6 Presentation
OSI Layer 5 Session
OSI Layer 4 Transport
OSI Layer 3 Network
OSI Layer 2 Data link
OSI Layer 1 Physical
Vehicle normal communication framework
ISO 20794-2 Application
Application control and event service interface
Application service interface
ISO 20794-5 Application layer conformance test plan
ISO 20794-2 Application layer
Application layer service interface
cmd_wakeup_req
ev_wakeup_ind
Vehicle diagnostic communication framework
Vehicle diagnostic communication use case standards
Use case-specific standards
ISO 14229-8 UDSonCXPI
Node configuration and diagnostic message interaction layer
ISO 14229-1 Application layer
Presentation layer standards
Vehicle manufacturer specific or ISO 22901 ODX
Upper layer service interface
ISO 14229-2 Session layer
Session layer service interface
ISO 20794-6 Transport and network layer conformance test plan
Transport layer service interface
ISO 20794-3 Transport and network layer
Network layer service interface
ISO 20794-7 Data link and physical layer conformance test plan
Data link layer service interface
Physical layer control and event service interface
ISO 20794-4 Data link and physical layer
OSI lower layers framework

Figure 1 — ISO 20794 documents reference according to OSI model

Road vehicles — Clock extension peripheral interface (CXPI) —

Part 3:
Transport and network layer

1 Scope

This document specifies the OSI transport layer and network layer by means of services and protocols. They can be used by different applications.

The transport layer:

- transforms a message into a single packet;
- adds protocol control information for diagnostic and node configuration into each packet;
- adds packet identifier for diagnostic and node configuration into each packet; and
- performs transport protocol error detection.

The transport layer protocol is not used for normal communication except to map the service interface parameters between lower and upper OSI layers.

The network layer:

- adds a node address for diagnostic and node configuration into each segment; and
- observes timing for diagnostic and node configuration of each segment.

The network layer is not used for normal communication except to map the service interface parameters between lower and upper OSI layers.

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

3 Terms and definitions

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

3.1
normal communication
NormalCom
measurement and control data

3.2
higher OSI layers
application, presentation, session, transport, and network layers according to ISO/IEC 7498-1:1994

3.3
lower OSI layers
data link and physical layers according to ISO/IEC 7498-1:1994

4 Symbols and abbreviated terms

4.1 Symbols

ms	millisecond
|	vertical bar indicates choice; either the left-hand side or the right-hand side of the vertical bar appears

4.2 Abbreviated terms

A_	prefix for application layer
AL	application layer
DiagNodeCfg	diagnostic and node configuration
Mtype	message type
N_	prefix for network layer
NAD	node address
NormalCom	normal communication
NL	network layer
OSI	open systems interconnection
PDU	protocol data unit
Ptype	packet type
ReqId	request identifier
SA	source address
SAP	service access point
SF	single frame
SIP	service interface parameter
T_	prefix for transport layer
T_Data	transport data service primitive

TA	target address
TL	transport layer
w/	with
w/o	without

5 Conventions

This document is based on the conventions discussed in the OSI Service Conventions as specified in ISO/IEC 10731.

6 Overview of transport layer and network layer

6.1 Properties

The transport layer and network layer have the following properties:

- addition (transmission) and elimination (reception) of transport- and network-specific header information to A_PDU;
- creation of T_PDU and N_PDU;
- mapping of service interface parameters between application and transport layer;
- mapping of service interface parameters between transport and network layer;
- determination of packet types to identify normal communication measurement and/or control data, diagnostic and node configuration data; and
- target address type and address handling of packet transmission and reception.

6.2 Packet types

The transport layer only supports single packets (no segmentation of application layer messages).

6.3 Error detection and indication

For detecting errors, the following measures are specified:

- transport layer packet type error;
- transport protocol control information single frame data length error;
- transport protocol control information single frame data length extension error; and
- node related error detection and error signalling.

To confirm an error occurrence during data transmission and reception, higher OSI layers are notified by an error indication.

7 Service interface parameters (SIP)

7.1 SIP — General

The following subclauses specify the service interface parameters and data types, which are used by the transport and network layer services.

7.2 SIP — Data type definitions

This requirement specifies the data type definitions of the CXPI service interface parameters.

REQ	0.1 SIP — Data type definitions
The data types shall be in accordance to: — Enum = 8-bit enumeration — Unsigned Byte = 8-bit unsigned numeric value — Unsigned Word = 16-bit unsigned numeric value — Byte Array = sequence of 8-bit aligned data — 2-bit Bit String = 2-bit binary coded — 8-bit Bit String = 8-bit binary coded — 16-bit Bit String = 16-bit binary coded	

7.3 SIP — TA, target address

This requirement specifies the target address parameter values of the CXPI service interface.

REQ	0.2 SIP — TA, target address
The TA parameter shall be of data type Unsigned Byte and shall be used to identify the target address of the information to be transmitted. Range: [01_{16} to FF_{16}]	

7.4 SIP — SA, source address

This requirement specifies the source address parameter values of the CXPI service interface.

REQ	0.3 SIP — SA, source address
The SA parameter shall be of data type Unsigned Byte and shall be used to identify the source address of the received information. Range: [01_{16} to FF_{16}]	

7.5 SIP — Ptype, packet type

This requirement specifies the packet type parameter values of the CXPI service interface.

REQ	0.4 SIP — Ptype, packet type
The Ptype parameter shall be of data type Enum and shall be used to identify the packet type and range of address information included in a service call. Range: [NormalCom, DiagNodeCfg]	

7.6 SIP — ReqId, request identifier

This requirement specifies the request identifier parameter values of the CXPI service interface.

REQ	0.5 SIP — ReqId, request identifier
The ReqId parameter shall be of data type Unsigned Byte and shall contain the request identifier. Range: [01_{16} to $7\mathrm{F}_{16}$]	

7.7 SIP — ReqTypeId, request type identifier

This requirement specifies the request type identifier parameter value of the CXPI service interface.

REQ	0.6 SIP — ReqTypeId, request type identifier
The ReqTypeId parameter shall be of data type Unsigned Byte and shall contain the request type identifier. Range: [00_{16}] NOTE ReqTypeId is used by the application to enable the polling method. It has a fixed value.	

7.8 SIP — PDU, protocol data unit

This requirement specifies the protocol data unit parameter values of the CXPI service interface.

REQ	0.7 SIP — PDU, protocol data unit
The PDU parameter shall be of data type Byte Array and shall contain the packet data (PDU) content of the request or response packet to be transmitted/received. Range: [00_{16} to FF_{16}]	

7.9 SIP — Length, length of PDU

This requirement specifies the length of PDU parameter value of the CXPI service interface.

REQ	0.8 SIP — Length, length of PDU
The Length parameter shall be of data type Unsigned Byte and shall contain the length of the PDU to be transmitted/received. Range: [00_{16} to FF_{16}]	

7.10 SIP — NMInfo, network management information

This requirement specifies the network management information parameter values of the CXPI service interface.

REQ	0.9 SIP — NMInfo, network management information
The NMInfo parameter shall be of data type Enum and shall contain the NM information in the response field. Type: Enum Range: 00_2 = [no request for wakeup_ind, sleep_ind prohibition] 01_2 = [no request for wakeup_ind, sleep_ind permission] 10_2 = [request for wakeup_ind, sleep_ind prohibition] 11_2 = [request for wakeup_ind, sleep_ind permission]	

7.11 SIP — SCT, sequence count

This requirement specifies the sequence count parameter values of the CXPI service interface.

REQ	0.10 SIP — SCT, sequence count
The SCT parameter shall be of data type Unsigned Byte and shall contain the sequence count information in the response field to be transmitted/received. Type: 2-bit Unsigned Byte Range: [00_2 to 11_2]	

7.12 SIP — Result, result

This requirement specifies the result parameter values of the CXPI service interface.

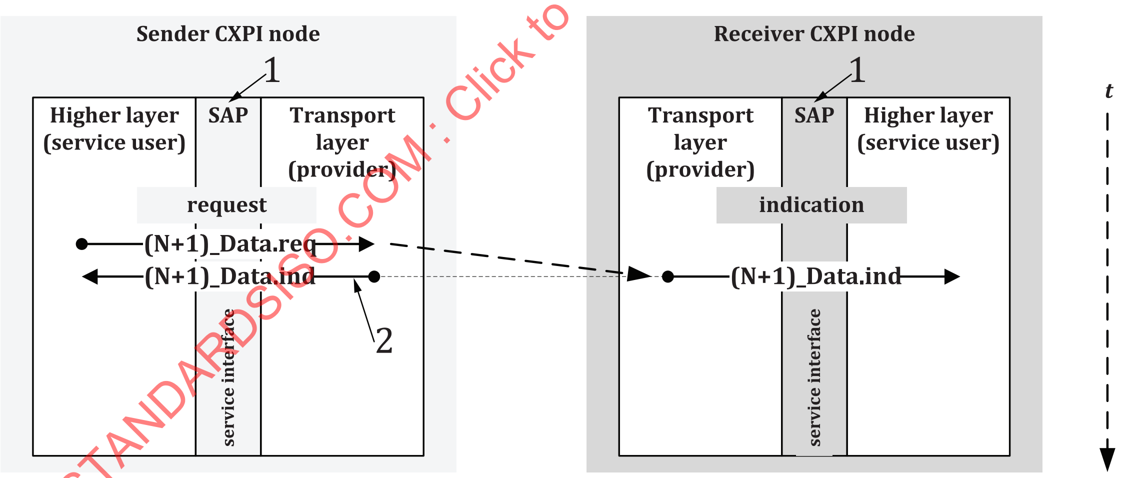
REQ	0.11 SIP — SIP — Result, result
The Result parameter shall be of data type 16-bit Bit String and shall contain the status relating to the outcome of a service execution. If two or more errors are discovered at the same time, then the transport layer or network layer entity shall set the appropriate error bit in the Result parameter. Range: [OK, DLL_Arb_Lost, Err_DLL_Byte, Err_DLL_CRC, Err_DLL_DLC, Err_DLL_DLCext_Value, Err_DLL_Parity, Err_DLL_Framing, Err_DLL_Value, Err_NL_TIMEOUT_A, Err_TL_Ptype, Err_TL_PCI_DL_Value, Err_TL_PCI_DLext_Value] The result OK shall be issued to the service user when the service execution completed without error. The result OK shall be issued to a service user on both, the sender and receiver side. The Err_... shall be issued to the service user when an error is detected. The Err_... shall be issued to the service user on both, the sender and receiver side. Length: 16-bit; (1-bit per result) Range: [0_2 to 1_2]	

8 Transport layer (TL)

8.1 SI — T_Data.req and T_Data.ind service interface

The service interface defines the service primitives and parameter mapping to the TL.

Figure 2 shows the TL T_Data.req and T_Data.ind service interface.



Key

1 service access point

2 read back from CXPI network provided by lower OSI layer

t time

Figure 2 — TL T_Data.req and T_Data.ind service interface

For normal communication (NormalCom) the sender node transmits, if master node, either a request protected type identifier field (polling method), or a request protected identifier field (T_Data.req). All nodes receive the request protected identifier field of type NormalCom (T_Data.ind). The node, which has corresponding PDU data, transmits the response PDU (T_Data.req) and all nodes receive the response PDU (T_Data.ind).

For diagnostic and node configuration (DiagNodeCfg) the sender node transmits a request message including a protected identifier field and the A_PDU (T_Data.req) onto the CXPI network. All nodes receive the DiagNodeCfg request message (T_Data.ind). The node(s), which is (are) targeted by the request message, transmit a DiagNodeCfg response message (T_Data.req) onto the CXPI network.

8.2 SI — T_Data.req and T_Data.ind service interface parameter mapping

This requirement specifies the transport layer service interface parameter mapping to lower OSI layers.

REQ	**4.1 SI — T_Data.req and T_Data.ind service interface parameter mapping**
The T_Data.req and T_Data.ind service interface parameter mapping is specified in Table 1.	

The normal communication packet (NormalCom) consists of a T_ReqId (request field) or a T_PDU (response field). The request field has no T_PDU and therefore, the T_Length = NULL. The response field has a T_PDU and therefore, the T_Length ≥ 0.

The diagnostic and node configuration packet (DiadeCfg) always consists of a T_ReqId (request field) and a T_PDU (response field). Therefore, the T_Length > 0.

Table 1 — T_Data.req and T_Data.ind service interface parameter mapping

Higher OSI layers **(service user)**	**Transport layer** **(service provider)**	**T_Data.req and T_Data.ind parameter validity**					
		NormalCom with T_Length				**DiagNodeCfg with T_Length**	
		= NULL		**≥'0'**		**>'0'**	
		.req	**.ind**	**.req**	**.ind**	**.req**	**.ind**
A_Mtype	T_Ptype	X[a]	X	X	X	X	X
A_Length	T_Length	X	X	X	X	X	X
A_ReqId	T_ReqId	X	X	X	X	X	X
A_ReqTypeId[b]	T_ReqTypeId[b]	X	X	—[c]	—[c]	—[c]	—[c]
A_TA	T_TA	—[c]	—[c]	—[c]	—[c]	X	X
A_SA	T_SA	—[c]	—[c]	—[c]	—[c]	X	X
A_PDU	T_PDU	—[c]	—[c]	X	X	X	X
A_NMInfo	T_NMInfo	—[c]	—[c]	X	X	X	X
A_SCT	T_SCT	—[c]	—[c]	X	X	X	X
A_Result	T_Result	—[c]	X	—[c]	X	—[c]	X

NOTE 1 A service interface call either includes a ReqId or a ReqTypeId parameter.

NOTE 2 If T_Ptype = DiagNodeCfg then the service provider packet consists of T_ReqId and T_PDU.

[a] Supported "X".

[b] Not used if T_Ptype = DiagNodeCfg.

[c] Not supported "—".

8.3 TL — Service interface with T_Ptype parameter mapping

8.3.1 TL — T_Data.req and T_Data.ind with T_Ptype = NormalCom (T_Length = NULL)

This requirement specifies the transport layer packet type NormalCom with T_Length = NULL via the T_Data.req and T_Data.ind interface.

REQ	**4.2 TL — T_Data.req and T_Data.ind with T_Ptype = NormalCom (T_Length = NULL)**

The T_Data.req and T_Data.ind service interface with T_Ptype = NormalCom with T_Length = NULL shall use the service interface parameters specified in Table 1 and Figure 3.
If the TL provider is the sender then the following mapping shall be implemented for a T_Data.req:
— T_Ptype = A_Mtype(NormalCom);
— T_ReqTypeId = A_ReqTypeId;
— T_ReqId = A_ReqId.
If the TL provider is the receiver then the following mapping shall be implemented for a T_Data.ind:
— A_Mtype = T_Ptype(NormalCom);
— A_ReqTypeId = T_ReqTypeId;
— A_ReqId = T_ReqId;
— A_Result = T_Result.	

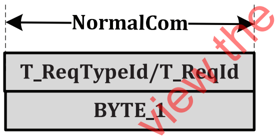


Figure 3 — TL T_Data.req and T_Data.ind with T_Ptype = NormalCom (T_Length = NULL)

8.3.2 TL — T_Data.req and T_Data.ind with T_Ptype = NormalCom (T_Length ≥ '0')

This requirement specifies the transport layer packet type NormalCom with T_Length ≥ '0' via the T_Data.req and T_Data.ind interface.

REQ **4.3 TL — T_Data.req and T_Data.ind with T_Ptype = NormalCom (T_Length ≥ '0')**

The T_Data.req and T_Data.ind service interface with T_Ptype = NormalCom with T_Length ≥ '0' shall use the service interface parameters specified in Table 1 and Figure 4.

If the TL provider is the sender then the following mapping shall be implemented for a T_Data.req:

— T_Ptype = A_Mtype(NormalCom);

— T_Length = A_Length;

— T_PDU = A_PDU;

— T_MNInfo = A_MNInfo;

— T_SCT = A_SCT.

If the TL provider is the receiver then the following mapping shall be implemented for a T_Data.ind:

— A_Mtype = T_Ptype(NormalCom);

— A_Length = T_Length;

— A_PDU = T_PDU;

— A_MNInfo = T_MNInfo;

— A_SCT = T_SCT;

— A_Result = T_Result.

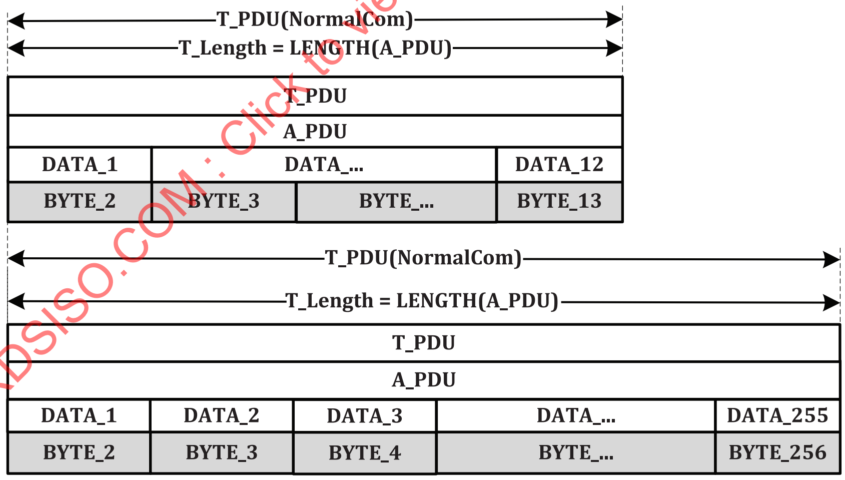


Figure 4 — TL T_Data.req and T_Data.ind with T_Ptype = NormalCom (T_Length ≥ '0')

8.3.3 TL — T_Data.req and T_Data.ind interface with T_Ptype = DiagNodeCfg

This requirement specifies the transport layer packet type DiagNodeCfg via the T_Data.req and T_Data.ind interface.

REQ	4.4 TL — T_Data.req and T_Data.ind interface with T_Ptype = DiagNodeCfg

The T_Data.req and T_Data.ind service interface with T_Ptype = DiagNodeCfg shall use the service interface parameters specified in Table 1 and in Figure 5.

If the TL provider is the sender then the following mapping shall be implemented for a T_Data.req:

— T_Ptype = A_Mtype(DiagNodeCfg);

— T_ReqId = A_ReqId;

— T_Length = LENGTH(A_PDU);

— T_TA = A_TA;

— T_PDU = [T_PCI, A_PDU];

— T_MNInfo = A_MNInfo;

— T_SCT = A_SCT.

If the TL provider is the receiver then the following mapping shall be implemented for a T_Data.ind:

— A_Mtype = T_Ptype(DiagNodeCfg);

— A_ReqId = T_ReqId;

— A_Length = LENGTH(T_PDU);

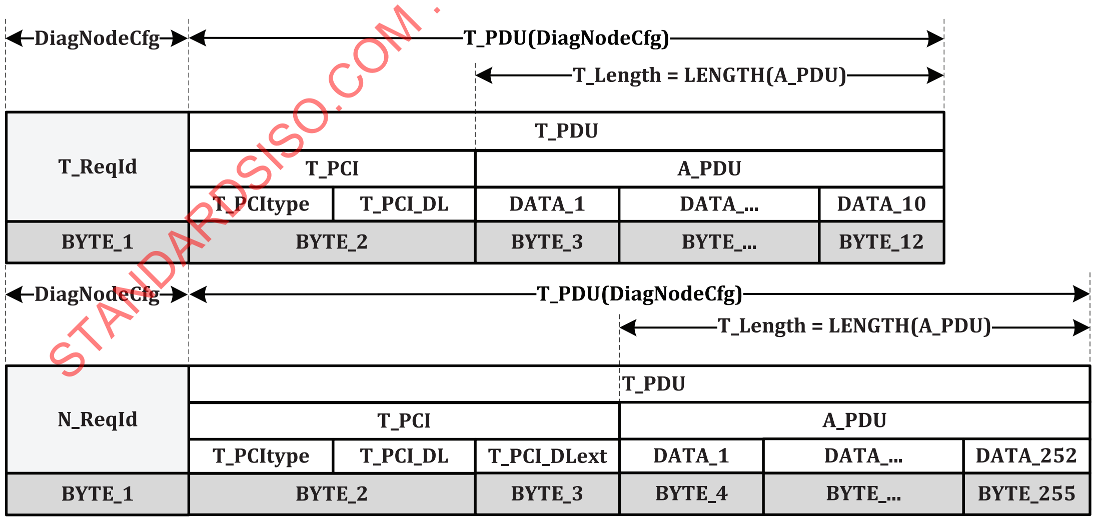
— A_SA = T_SA;

— A_PDU = [T_PDU w/o T_PCI];

— A_MNInfo = T_MNInfo;

— A_SCT = T_SCT;

— A_Result = T_Result.



T_ReqId	T_PDU				
	T_PCI		A_PDU		
	T_PCItype	T_PCI_DL	DATA_1	DATA_...	DATA_10
BYTE_1	BYTE_2		BYTE_3	BYTE_...	BYTE_12

N_ReqId	T_PDU					
	T_PCI			A_PDU		
	T_PCItype	T_PCI_DL	T_PCI_DLext	DATA_1	DATA_...	DATA_252
BYTE_1	BYTE_2		BYTE_3	BYTE_4	BYTE_...	BYTE_255

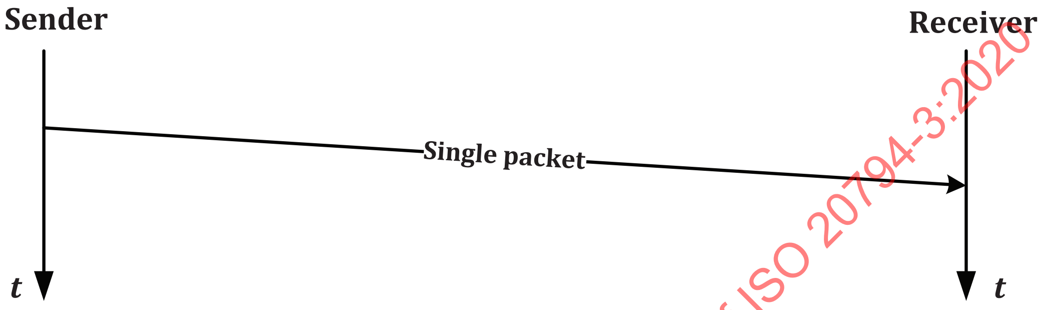
Figure 5 — TL T_Data.req and T_Data.ind with T_Ptype = DiagNodeCfg

8.4 TL — Internal operation

The transport layer transports application layer messages as packets. The TL supports two kinds of A_PDU length up to 12 and of length greater than 12. The packets type are single packet.

REQ	4.5 TL — Internal operation
The internal operation of the CXPI transport layer shall support unsegmented (single packet) unacknowledged data transmission.	

Figure 6 shows unsegmented unacknowledged data transmission.



Key

t time

Figure 6 — TL unsegmented unacknowledged data transmission

8.5 TL — T_PDU

8.5.1 TL — T_PDU definition

The T_PDU represents the TL header and application protocol data unit (A_PDU).

REQ	4.6 TL — T_PDU — TL — T_PDU definition
If T_Ptype = DiagNodeCfg then the T_PDU shall consist of [T_PCI, A_PDU].	
If T_Ptype = NormalCom then the T_PDU shall consist of [A_PDU].	
Type: `Byte Array`	
Range: [00_{16} to FF_{16}]	

8.5.2 TL — T_PCI definition

8.5.2.1 TL — General

The T_PCI parameter consists of a T_PCItype and T_PCI_DL. The T_PCI_DLext parameter depends on the T_Length value (see Table 2).

8.5.2.2 TL — T_PCItype definition

The T_PCItype is used by the TL to identify which type of frame (e.g. single frame) is supported.

REQ	4.7 TL — T_PDU — TL — T_PCItype definition
The T_PCI byte and type values shall only be included if T_Ptype = DiagNodeCfg and shall be in accordance with Table 2 and Table 3.	

Table 2 — T_PCI byte definition

T_Ptype	T_Length	T_PCI		
		BYTE_1		BYTE_2
		Bit 7 to 4	Bit 3 to '0'	Bit 7 to '0'
DiagNodeCfg	$(1 \leq T\_Length \leq 10)$	PCItype = 0000_2	DL	N/A
DiagNodeCfg	$(11 \leq T\_Length \leq 252)$	PCItype = 0000_2	0000_2	DLext

Table 3 — T_PCItype value definition

PCItype	Description
0000_2	**Single packet** The TL protocol provides an optimised implementation of the transport protocol in a single packet (T_PCItype = 0000_2). T_Length definition, see Table 2.
0001_2 to 1111_2	**Invalid** This range of values shall not be used in this document and shall be ignored by the receiving node.

8.5.2.3 TL — T_PCI_DL and T_PCI_DLext definition

The T_PCI_DL and T_PCI_DLext parameter contains the length of the A_PDU used by the TL.

REQ	4.8 TL — T_PDU — TL — T_PCI_DL and T_PCI_DLext definition
The T_PCI_DL and T_PCI_DLext parameter in the T_PDU shall be used to transmit the service user data length received in the T_Length parameter in the T_Data.req and T_Data.ind service interface as specified Table 4.	

Table 4 — T_PCI_DL and T_PCI_DLext value definition

Value	Description
00_{16}	**Invalid**
	This value is invalid.
0001_2 to 1010_2	**T_PCI_DL**
	The T_PCI_DL parameter value shall be encoded in the low nibble of the 4 bit of T_PCI parameter value.
$0B_{16}$ to FC_{16}	**T_PCI_DLext**
	The T_PCI_DLext parameter value shall be encoded in the T_PCI parameter value.

8.6 TL — Function models

8.6.1 TL — Transmission logic

The TL transmission function model specifies the logic and the mapping of service interface parameters to the TL-specific packet parameters. The TL transmission logic distinguishes between packet types T_Ptype = NormalCom and T_Ptype = DiagNodeCfg transmissions.

If the T_Ptype = NormalCom, then the length of the A_PDU determines, whether a packet 'A' or a packet 'B' is transmitted to the lower OSI layer. If the T_Ptype = DiagNodeCfg, then a packet 'C' is transmitted to the lower OSI layer.

REQ	4.9 TL — Function models — TL — Transmission logic
The TL transmission function model shall be in accordance with the logic specified in Figure 7.	

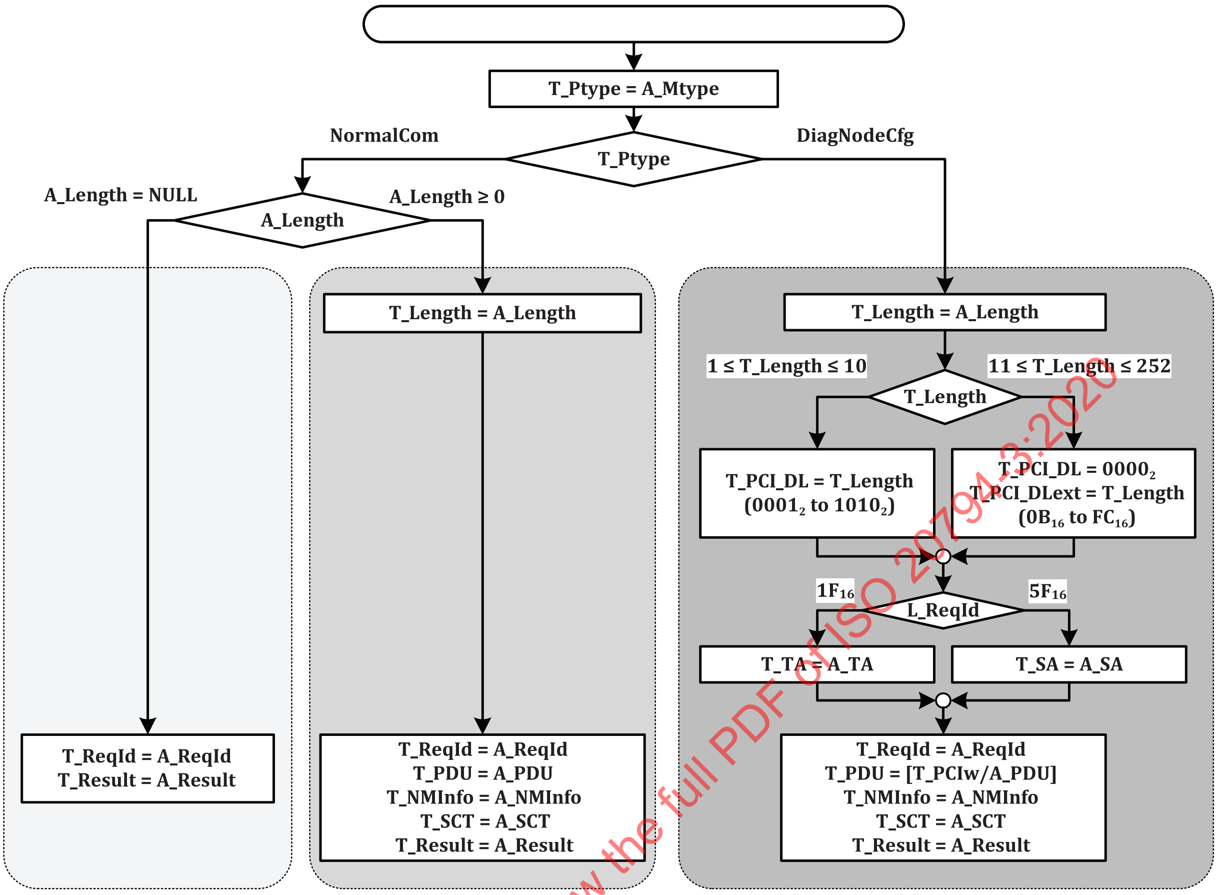


Figure 7 — TL transmission logic

8.6.2 TL — Reception logic

The TL reception function model specifies the logic and the mapping of service interface parameters to the TL-specific packet parameters. The TL reception logic distinguishes between packet type T_Ptype = NormalCom and T_Ptype = DiagNodeCfg reception.

If the T_Ptype = NormalCom then the length of the A_PDU determines, whether a packet 'A' or a packet 'B' is transmitted to the lower OSI layer. If the T_Ptype = DiagNodeCfg then a packet 'C' is transmitted to the lower OSI layer.

REQ	4.10 TL — Function models — TL — Reception logic
The TL reception function model shall be in accordance with the logic specified in Figure 8.	

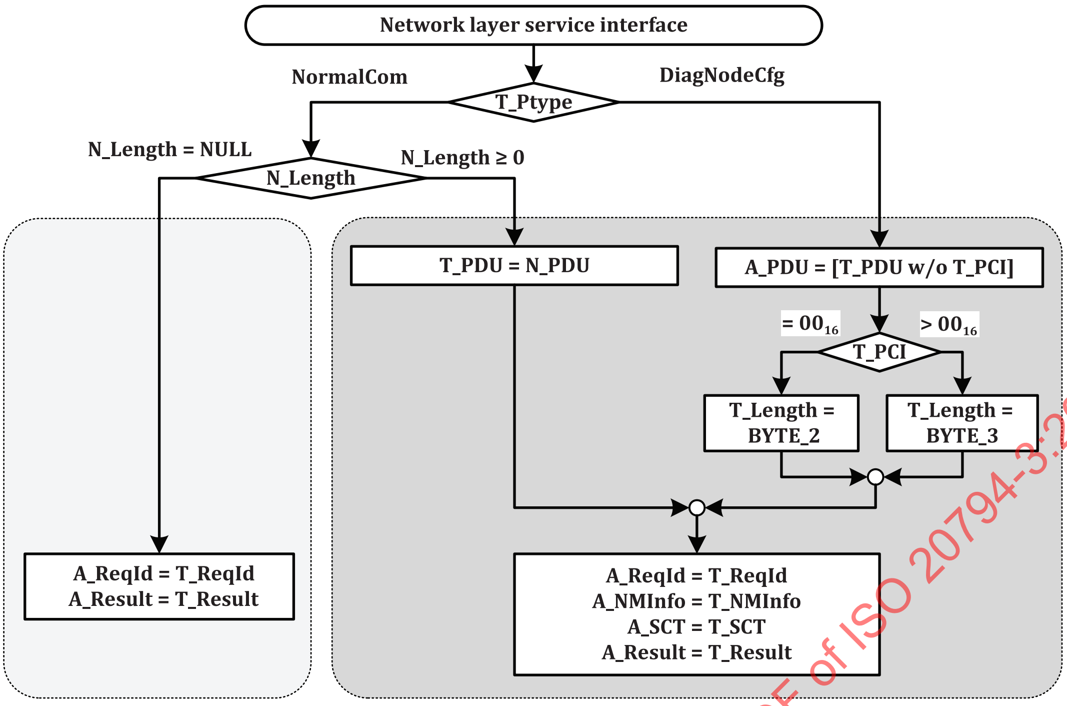


Figure 8 — TL reception function model

8.7 TL — Error detection

The TL error management specifies requirements related to transport protocol error detection.

REQ	**4.11 TL — Error detection — T_Data.ind and T_Data.conf with T_PCItype ≠ 0000_2**
Reception of a T_PDU with T_PCItype ≠ 0000_2 shall be ignored by the receiver.	

REQ	**4.12 TL — Error detection — T_Data.conf frame reception with T_PCI_DL = [0_{16}, B_{16} to F_{16}]**
The receiver shall ignore any reception of packet with T_Length = T_PCI_DL including a value — 0_{16}, — B_{16} to F_{16}.	

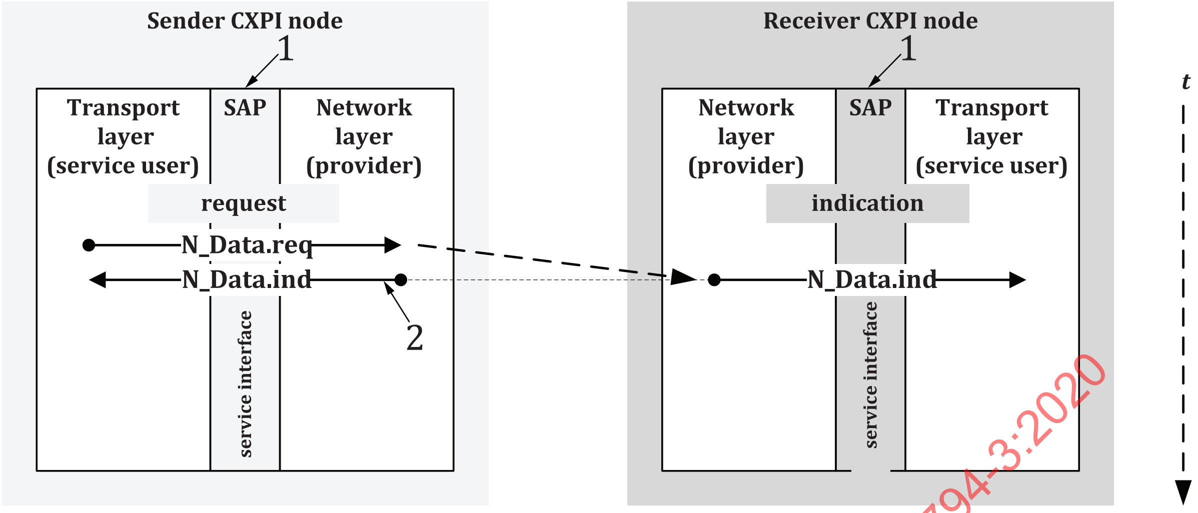
REQ	**4.13 TL — Error detection — T_Data.conf frame reception with T_PCI_DLext = [01_{16} to A_{16}, FD_{16} to FF_{16}]**
The receiver shall ignore any reception of packet with T_Length = T_PCI_DLext including a value — 01_{16} to $0A_{16}$, — FD_{16} to FF_{16}.	

9 Network layer (NL)

9.1 SI — N_Data.req and N_Data.ind service interface

The service interface defines the service primitives and parameter mapping to the NL.

Figure 9 shows the NL N_Data.req and N_Data.ind service interface.



Key

1 service access point

2 read back from CXPI network provided by lower OSI layer

t time

Figure 9 — NL N_Data.req and N_Data.ind service interface

For normal communication (NormalCom) the sender node transmits, if master node, either a request protected type identifier field (polling method), or a request protected identifier field (N_Data.req). All nodes receive the request protected identifier field of type NormalCom (N_Data.ind). The node, which has corresponding PDU data, transmits the response PDU (N_Data.req) and all nodes receive the response PDU (N_Data.ind).

For diagnostic and node configuration (DiagNodeCfg) the sender node transmits a request protected identifier field DiagNodeCfg (N_Data.req) onto the CXPI network. All nodes receive the request protected identifier field of type DiagNodeCfg (N_Data.ind). The node, which has corresponding PDU data, transmits the response PDU (N_Data.req) and all nodes receive the response PDU (N_Data.ind).

9.2 SI — N_Data.req and N_Data.ind service interface parameter mapping

This requirement specifies the network layer service interface parameter mapping to lower OSI layers.

REQ	3.1 SI — N_Data.req and N_Data.ind service interface parameter mapping
The N_Data.req and N_Data.ind service interface parameter mapping is specified in Table 5.	

The normal communication packet (NormalCom) consists of a N_ReqId (request field) or a N_PDU (response field). The request field has no N_PDU and therefore, the N_Length = NULL. The response field has a N_PDU and therefore, the N_Length ≥ '0'.

The diagnostic and node configuration packet (DiagNodeCfg) always consist of a N_ReqId (request field) and a N_PDU (response field). Therefore, the N_Length > '0'.

Table 5 — N_Data.req and N_Data.ind service interface parameter mapping

Transport layer (service user)	Network layer (service provider)	N_Data.req and N_Data.ind parameter validity					
		NormalCom with N_Length				DiagNodeCfg with N_Length	
		= NULL		≥'0'		>'0'	
		.req	.ind	.req	.ind	.req	.ind
T_Ptype	N_Ptype	X[a]	X	X	X	X	X
T_Length	N_Length	X	X	X	X	X	X
T_ReqId	N_ReqId	X	X	X	X	X	X
T_ReqTypeId[b]	N_ReqTypeId[b]	X	X	—[c]	—[c]	—[c]	—[c]
T_TA	N_TA	—[c]	—[c]	—[c]	—[c]	X	X
T_SA	N_SA	—[c]	—[c]	—[c]	—[c]	X	X
T_PDU	N_PDU	—[c]	—[c]	X	X	X	X
T_NMInfo	N_NMInfo	—[c]	—[c]	X	X	X	X
T_SCT	N_SCT	—[c]	—[c]	X	X	X	X
T_Result	N_Result	—[c]	X	—[c]	X	—[c]	X

NOTE 1 A service interface call either includes a ReqId or a ReqTypeId parameter.

NOTE 2 If N_Ptype = DiagNodeCfg then the service provider packet consists of N_ReqId and N_PDU.

[a] Supported "X".

[b] Not used if N_Ptype = DiagNodeCfg.

[c] Not supported "—".

9.3 NL — Service interface with N_Ptype parameter mapping

9.3.1 NL — N_Data.req and N_Data.ind with N_Ptype = NormalCom (N_Length = NULL)

This requirement specifies the transport layer packet type NormalCom with N_Length = NULL via the N_Data.req and N_Data.ind interface.

REQ 3.2 NL — N_Data.req and N_Data.ind with N_Ptype = NormalCom (N_Length = NULL)

The N_Data.req and N_Data.ind service interface with N_Ptype = NormalCom with N_Length = NULL shall use the service interface parameters specified in Table 5 and Figure 10.

If the NL provider is the sender then the following mapping shall be implemented for a N_Data.req:

- N_Ptype = T_Mtype(NormalCom);
- N_ReqTypeId = T_ReqTypeId;
- N_ReqId = T_ReqId.

If the NL provider is the receiver then the following mapping shall be implemented for a N_Data.ind:

- T_Mtype = N_Ptype(NormalCom);
- T_ReqTypeId = N_ReqTypeId;
- T_ReqId = N_ReqId;
- T_Result = N_Result.

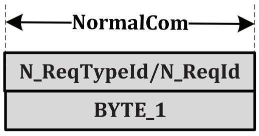


Figure 10 — NL N_Data.req and N_Data.ind with N_Ptype = NormalCom (N_Length = NULL)

9.3.2 TL — N_Data.req and N_Data.ind with N_Ptype = NormalCom (N_Length ≥ '0')

This requirement specifies the transport layer packet type NormalCom with N_Length ≥ '0' via the N_Data.req and N_Data.ind interface.

REQ	3.3 NL — N_Data.req and N_Data.ind with N_Ptype = NormalCom (N_Length ≥ '0')

The N_Data.req and N_Data.ind service interface with N_Ptype = NormalCom with N_Length ≥ '0' shall use the service interface parameters specified in Table 5 and Figure 11.
If the NL provider is the sender then the following mapping shall be implemented for a N_Data.req:
— N_Ptype = T_Mtype(NormalCom);
— N_Length = T_Length;
— N_PDU = T_PDU;
— N_MNInfo = T_MNInfo;
— N_SCT = T_SCT.
If the TL provider is the receiver then the following mapping shall be implemented for a T_Data.ind:
— T_Mtype = N_Ptype(NormalCom);
— T_Length = N_Length;
— T_PDU = N_PDU;
— T_MNInfo = N_MNInfo;
— T_SCT = N_SCT.	

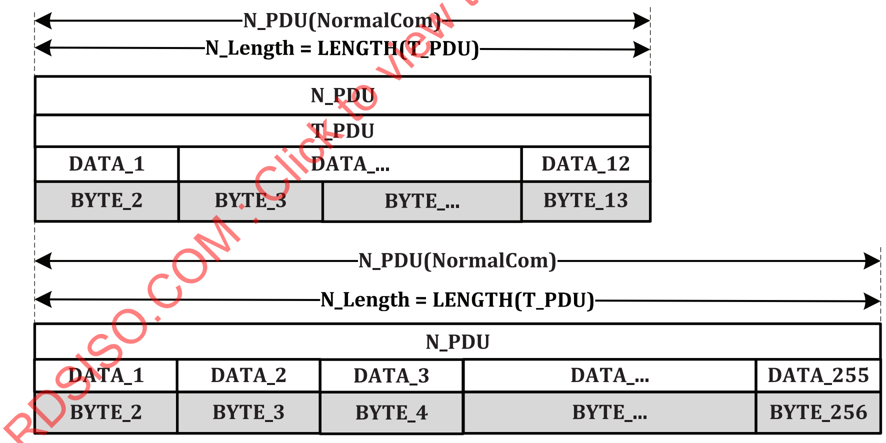


Figure 11 — NL N_Data.req and N_Data.ind with N_Ptype = NormalCom (N_Length ≥ '0')