ISO/IEC 14543-3-2

Edition 1.0 2006-09

INTERNATIONAL STANDARD

Information technology – Home electronic system (HES) architecture – Part 3-2: Communication layers – Transport, network and general parts of data link layer for network based control of HES Class 1

ISO/IEC 14543-3-2:2006-09(en)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

ISO/IEC 14543-3-2

Edition 1.0 2006-09

INTERNATIONAL STANDARD

Information technology – Home electronic system (HES) architecture – Part 3-2: Communication layers – Transport, network and general parts of data link layer for network based control of HES Class 1

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 35.240.67

ISBN 2-8318-8804-2

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

INFORMATION TECHNOLOGY – HOME ELECTRONIC SYSTEM (HES) ARCHITECTURE –

Part 3-2: Communication layers – Transport, network and general parts of data link layer for network based control of HES Class 1

FOREWORD

1) ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards. Their preparation is entrusted to technical committees; any ISO and IEC National Committee interested in the subject dealt with may participate in this preparatory work. International governmental and non-governmental organizations liaising with ISO and IEC also participate in this preparation.

2) In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

3) The formal decisions or agreements of IEC or ISO on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC and ISO National Committees.

4) IEC, ISO or ISO/IEC Publications have the form of recommendations for international use and are accepted by IEC and ISO National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC, ISO or ISO/IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

5) In order to promote international uniformity, IEC and ISO National Committees undertake to apply IEC, ISO or ISO/IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any ISO/IEC Publication and the corresponding national or regional publication should be clearly indicated in the latter.

6) ISO or IEC provide no marking procedure to indicate their approval and cannot be rendered responsible for any equipment declared to be in conformity with an ISO/IEC Publication.

7) All users should ensure that they have the latest edition of this publication.

8) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC or ISO member bodies for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication of, use of, or reliance upon, this ISO/IEC publication or any other IEC, ISO or ISO/IEC publications

9) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

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

International Standard ISO/IEC 14543-3-2 was prepared by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

This International Standard together with ISO/IEC 14543-3-1 cancels and replaces ISO/IEC TR 14543-3, published in 2000. It constitutes a complete revision of the principles outlined in ISO/IEC TR 14543-3 and provides the specifications essential for an international standard.

This International Standard has been approved by vote of the member bodies, and the voting results may be obtained from the address given on the title page.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

INTRODUCTION

This standard specifies the Media independent requirements for the data link layer and the requirements for the network layer and the transport layer for Home Electronic Systems.

This standard provides the communication stack targeted for providing the services specified in ISO/IEC 14543-3-3 (EN 50090-3-2) “User Process” and ISO/IEC 14543-3-1 “Application Layer for networked based control of HES Class 1”. It can be used as communication stack on the physical layers as specified in ISO/IEC 14543-3-5, ISO/IEC 14543-3-6 and ISO/IEC 14543-3-7 (EN 50090-5-x).

Currently, ISO/IEC 14543, *Information technology – Home Electronic System (HES) architecture*, consists of the following parts:

Part 2-1:	*Introduction and device modularity*
Part 3-1:	*Communication layers – Application layer for network based control of HES Class 1*
Part 3-2:	*Communication layers – Transport, network and general parts of data link layer for network based control of HES Class 1*
Part 3-3:	*User process for network based control of HES Class 1 (under consideration)*
Part 3-4:	*System management – Management procedures for network based control of HES Class 1 (under consideration)*
Part 3-5:	*Media and media dependent layers – Power line for network based control of HES Class 1 (under consideration)*
Part 3-6:	*Media and media dependent layers – Twisted pair for network based control of HES Class 1 (under consideration)*
Part 3-7:	*Media and media dependent layers – Radio frequency for network based control of HES Class 1 (under consideration)*
Part 4:	*Home and building automation in a mixed-use building (technical report)*
Part 5-1:	*Intelligent grouping and resource sharing for HES Class 2 and Class 3 – Core ptotocol*
Part 5-2:	*Intelligent grouping and resource sharing for HES Class 2 and Class 3 – Device certification*
	Additional parts may be added later.

INFORMATION TECHNOLOGY –
HOME ELECTRONIC SYSTEM (HES) ARCHITECTURE –

Part 3-2: Communication layers – Transport, network and general parts of data link layer for network based control of HES Class 1

1 Scope

This part of ISO/IEC 14543 specifies the services and protocol in a physical layer independent way for the data link layer and for the network layer and the transport layer for usage in Home Electronic Systems.

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 (all parts), *Information technology – Open Systems Interconnection – Basic reference model*

ISO/IEC 14543-2-1, *Information technology – Home Electronic System (HES) Architecture – Part 2-1: Introduction and device modularity*

ISO/IEC 14543-3-1 *Information technology – Home Electronic System (HES) Architecture – Part 3-1: Communication layers – Application layer for network based control of HES Class 1*

NOTE 1 The provisions of the referenced specifications, as identified in this subclause, are valid within the context of this International Standard. The reference to a specification within this International Standard does not give it any further status within ISO/IEC; in particular, it does not give the referenced specification the status of an International Standard.

EN 50090-5 (all parts), *Home and Building Electronic Systems (HBES) – Part 5: Media and media dependent layers*

NOTE 2 Reference to EN 50090-5-x is to be considered as deleted as soon as ISO/IEC 14543-3-5, ISO/IEC 14543-3-6 and ISO/IEC 14543-3-7 are approved.

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this part the terms and definitions given in ISO/IEC 14543-2-1 and the following apply.

3.1.1
individual address
IA
unique identifier for every device in a network

NOTE The individual address is a 2-octet value that consists of an 8-bit subnetwork address and an 8-bit device address.

3.1.2
subnetwork address
SNA
part of the individual address; consists of a 4-bit line address and a 4-bit area address, that specifies the subnetwork in which the device is mounted

3.1.3
area address
part of the individual address that specifies the area in which the device is mounted

3.1.4
line address
part of the individual address that specifies the line in which the device is mounted

3.1.5
device address
unique identifier for every device in a subnetwork; the device address is an 8-bit value

NOTE Figure 1 shows the relationship between individual address, subnetwork address, area address, line address and device address.

<table>
<tr><td colspan="16">Individual address</td></tr>
<tr><td colspan="8">Octet 0</td><td colspan="8">Octet 1</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">Area address</td><td colspan="4">Line address</td><td colspan="8" rowspan="2">Device address</td></tr>
<tr><td colspan="8">Subnetwork address</td></tr>
</table>

Figure 1 – Individual address

3.1.6
group address
GA
2-octet value

NOTE Figure 2 shows the group address, consisting of main group and sub-group address.

<table>
<tr><td colspan="16">Group address</td></tr>
<tr><td colspan="8">Octet 0</td><td colspan="8">Octet 1</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Main group</td><td colspan="8">Sub-group</td></tr>
</table>

Figure 2 – Group address

3.1.7
datagram
full sequence of elements (physical symbols) transporting a frame on the physical medium

3.1.8
frame
sequence of octets exchanged between data link layers through the physical layer

3.2 Abbreviations

ACK	Acknowledge
APDU	Application layer Protocol Data Unit
con	confirmation
GA	Group Address
HES Class 1	refers to simple control and command.
HES Class 2	refers to Class 1 plus simple voice and stable picture transmission
HES Class 3	refers to Class 2 plus complex video transfers
IA	Individual Address
ind	indication
IACK	Immediate Acknowledge
LPDU	Link layer Protocol Data Unit
LSDU	Link layer Service Data Unit
NACK	Negative Acknowledge
NPDU	Network layer Protocol Data Unit
NSDU	Network layer Service Data Unit
PDU	Protocol Data Unit
req	request
SNA	Sub-Network Address
TSAP	Transport layer Service Access Point
TPDU	Transport layer Protocol Data Unit
UART	Universal Asynchronous Receiver Transmitter

4 Conformance

An entity of operational exchange conforming to this International Standard shall contain a physical layer and data link layer in accordance with the requirements of clause 5, a network layer in accordance with the requirements of clause 6 and a transport layer in accordance with the requirements of clause 7.

5 Requirements for the physical layer and independent data link layer

5.1 Functions of the data link layer

The data link layer (also called "Layer-2") is the layer between the data link layer user and the physical layer. The data link layer conforms to the definitions of the ISO/OSI model (ISO/IEC 7498) data link layer. It provides medium access control and logical link control.

The data link layer is concerned with reliable transport of single frames between two or more devices on the same subnetwork.

- When transmitting, it is responsible for
  - building up a complete frame from the information passed to it by the network layer,

- gaining access to the medium according to the particular medium access protocol in use, and
- transmitting the frame to the data link layer in the peer entity or entities, using the services of the physical layer.

If the transmission fails, the transmitting data link layer may decide to try again after a certain interval. In particular, if the remote device signals that its buffers are temporarily full, the data link layer will wait for a pre-determined time and then attempt to re-transmit the frame (flow control).

- When receiving, data link layer is responsible for
  - determining whether the frame is intact or corrupted,
  - deciding after destination address check to pass the frame to upper layers and
  - issuing positive or negative acknowledgements back to the transmitting data link layer.

The data link layer shall provide some means to prevent from service duplication (in case of repetitions because of corrupted acknowledgement frames).

The services provided include individual, group and broadcast addressing options.

The data link layer uses the services of the physical layer and provides services to the data link layer user (see Figure 3).

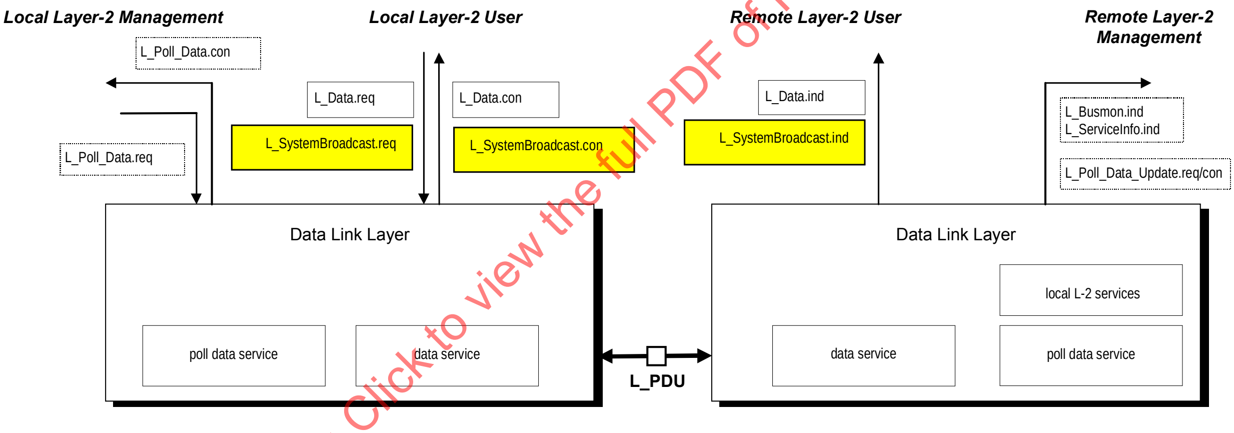

Figure 3 – Interaction of the data link layer

5.2 Possible media and their impact on layer-2

The data link layer is defined for the following media:

Twisted pair 0;
Twisted pair 1;
Powerline 110;
Powerline 132;
Radio-frequency.

Data link layer will also be defined for the following media:

Infra-red;
Ethernet.

The data link layer is open for new media in the future.

Each medium needs a dedicated medium access control and a logical link control that adapts to the medium access control. This clause focuses on medium independent features, for example on the provided service interface to network layer.

The physical layer dependent requirements are specified in ISO/IEC 14543-3-5, ISO/IEC 14543-5-6 and ISO/IEC 14543-3-7 (EN 50090-5 series).

5.3 Data link layer services

5.3.1 Data link layer modes

The data link layer mode defines which data link layer services shall be available to the data link layer user. There shall be 2 data link layer modes:

1) the normal mode;

2) the busmonitor mode.

In normal mode the remote L_Data service, the remote L_SystemBroadcast service, the remote L_Poll_Data service and the local L_Service_Information service shall be available to the data link layer user. In busmonitor mode only the local L_Busmon service shall be available. The data link layer mode is a parameter of layer-2.

The frame effectively sent on the physical medium link layer protocol data unit (LPDU) is medium dependent. Therefore, it is described in ISO/IEC 14543-3-5, ISO/IEC 14543-5-6 and ISO/IEC 14543-3-7 (EN 50090-5 series), respectively.

5.3.2 L_Data service

5.3.2.1 General

The L_Data service is a frame transfer service. It transmits a single link layer service data unit (LSDU) to data link layer of one or several devices connected to the same subnetwork. The destination address may be an individual address or a group address (multicast or broadcast). The service is acknowledged or not, depending on the quality of service requested.

There shall be three service primitives, as shown in Figure 4.

a) L_Data.Req shall be used to transmit a frame.

b) L_Data.Ind shall be used to receive a frame.

c) L_Data.Con shall be used as a local primitive generated by the local layer-2 for its own client to indicate that it is satisfied with the transmission.

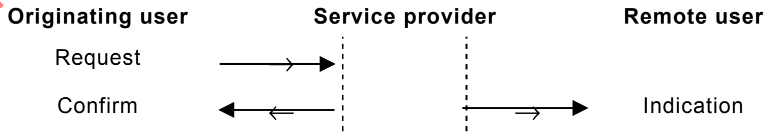

Figure 4 – Exchange of primitives for the L_Data-Service

If the local user of Layer-2 prepares an LSDU for the remote user it shall apply the L_Data.req primitive to pass the LSDU to the local layer-2. The local layer-2 shall accept the service request and try to send the LSDU to the remote layer-2 with the relevant frame format.

The local layer-2 shall pass an L_Data.con primitive to the local user that indicates either a correct or erroneous data transfer. Depending on whether an L2-acknowledgement is requested or not, this confirmation is related to the reception of the L2-acknowledgement or only to the transmission of the frame on the medium.

L_Data.req(source_address, destination_address, address_type, priority, octet_count, ack_request, frame_format, lsdu)

source_address this parameter shall be used to indicate the source address of the requested frame; it shall be the individual address of the device that requests the service primitive

destination_address: this parameter shall be used to indicate the destination address of the requested frame; it shall be either an individual address or a group address

address_type: this parameter shall be used to indicate whether the destination_address of the requested frame is an individual address or a group address

priority: this parameter shall be used to indicate the priority that shall be used to transmit the requested frame; it shall be “system”, “urgent”, “normal” or “low”

octet_count: this parameter shall be used to indicate the length information of the requested frame

ack_request: this parameter shall be used to indicate whether a layer-2 acknowledge is mandatory or optional

frame_format: standard or extended frame format

lsdu: this parameter shall be used to contain the user data to be transferred by layer-2

L_Data.con(destination_address, address_type, priority, frame_format, lsdu, l_status)

destination_address: this parameter shall be used to indicate the destination address of the transmitted frame; it shall be either an individual address or a group address

address_type: this parameter shall be used to indicate whether the destination_address of the transmitted frame is an individual address or a group address

priority: this parameter shall be used to indicate the priority that has been used to transmit the transmitted frame; it shall be “system”, “urgent”, “normal” or “low”

lsdu: this parameter shall be used to indicate the length information of the transmitted frame

frame_format: standard or extended frame format

l_status: ok: the value of this parameter shall be used to indicate that the transmission of the frame has been successful

not_ok: the value of this parameter shall be used to indicate that the transmission of the frame did not succeed

L_Data.ind(source_address, destination_address, address_type, priority, ack_request, octet_count, frame_format, lsdu)

source_address: this parameter shall be used to indicate the source address of the received frame; it shall be the individual address of the device that has transmitted the service primitive

destination_address: this parameter shall be used to indicate the destination address of the received frame; it shall be either an individual address or a group address

address_type: this parameter shall be used to indicate whether the destination_address of the received frame is an individual address or a group address

priority: this parameter shall be used to indicate the priority of the received frame; it shall be “system”, “urgent”, “normal” or “low”

ack_request: this parameter shall be used to indicate whether a layer-2 acknowledge is mandatory or optional

octet_count: this parameter shall be used to indicate the length information of the received frame

frame_format: standard or extended frame format

lsdu: this parameter shall be used to contain the user data that has been received by layer-2

5.3.2.2 Usage of priority

Table 1 – Usage of priority

Priority value	Priority	Usage
11	low	shall be used for long frames, burst traffic, etc.
01	normal	shall be used as the default for short frames
10	urgent	shall be used exclusively for urgent frames
00	system	shall be used for high priority, system configuration and management procedures

The usage conditions for these priorities, see Table 1, are specified in ISO/IEC 14543–3-1.

In a network, the frame traffic using urgent priority shall not exceed 5 % of the total traffic (integration period: 1 min maximum).

5.3.2.3 Octet count

This service parameter shall contain the number of octets of the transported application layer protocol data unit (APDU).

The Octet Count parameter shall be used on each medium to encode the LPDU length field as follows.

For standard frames, the length field shall contain the number of octets in the APDU coded in 4 bit.

For extended frames, the length field shall contain the number of octets in the APDU coded in 8 bit except the value FFh. The value FFh (255) is used as an escape-code.

The escape-code (“ESC”) shall be available for future high speed media to enable larger lengths.

5.3.2.4 Ack_request

This service parameter shall be used to indicate whether a link layer acknowledge is requested or not.

5.3.2.5 Frame_format

This parameter shall be used to select the Standard or Extended Frame Format for Data Link Layer and shall include information for the used extended frame type, see Figure 5.

If the frame_format parameter is 0 the Standard Frame Format shall be used. If this parameter is different from 0, it shall be used as the frame_format in the extended control field.

For the definition of the extended control field, see the medium dependent layer description in ISO/IEC 14543-3-5, ISO/IEC 14543-5-6 and ISO/IEC 14543-3-7 (EN 50090-5 series) respectively.

Octet 3								
7	6	5	4	3	2	1	0	
Frame type				Extended Frame Format				FT = Frame type (0 = Standard, 1 = Extended) (for, standard the frame type bit in the control field is 1) EFF = Frame Format in case of FT=1 = Extended Frame Format
FT	0	0	0	t	t	t	t	
0	0	0	0	0	0	0	0	Standard Frame Format Standard Group or Individual
1	0	0	0	0	0	0	0	Extended Frame Format Standard Group or Individual
1	0	0	0	0	1	x	x	LTE-HEE extended address type
								All other codes are reserved for future use

Figure 5 – Frame_format parameter

The Extended Frame Format from the frame_format parameter shall be placed in the extended control field. The position of the extended frame type is medium dependent, see Figure 6.

The decision whether to use Standard or Extended Frame Format shall be made in the Application Layer and selected by the frame_format parameter in T_Data_.... services. The remote Application Layer shall be tolerant towards usage of long frames if short frames would be sufficient:
Example: A_PropertyValue_Read-PDU shall fit into Standard (short) Frame Format. But if received using Extended (long) Frame Format it shall be accepted anyway by the remote Application Layer and the corresponding A_PropertyValue_Response-PDU shall be transported using the appropriate short or long format.

Extended Frame Format (EFF)				
b_3 CtrlE_3	b_2 CtrlE_2	b_1 CtrlE_1	b_0 CtrlE_0	Usage
0	0	0	0	Standard messages enabling long APDU > 15 octets Standard usage of DA for peer to peer or group messages
0 0 0	0 0 0	0 1 1	1 0 1	Reserved
0	1	X	X	LTE-HEE extended message format CtrlE_1, CtrlE_0 containing extension of DA group address
1 1 1 1 1 1 1	0 0 0 0 1 1 1	0 0 1 1 0 0 1	0 1 0 1 0 1 0	Reserved
1	1	1	1	Escape

Figure 6 – Coding of Extended Frame Format

The Extended Frame Format shall not be used instead of Standard Frame Format if encoding capabilities of L_Data-Standard frame are sufficient (e.g. for short frames).

5.3.3 L_SystemBroadcast service

The L_SystemBroadcast service is a frame transfer service. It shall transmit a single link layer service data unit (LSDU) to the data link layer of all devices within the network. The destination address shall be the system broadcast address (Domain Address = 0000h and destination address = 0000h and address_type = multicast). The service may be acknowledged or not, depending on the transmission medium.

There shall be three service primitives:

1) L_SystemBroadcast.req shall be used to transmit a frame;

2) L_SystemBroadcast.ind shall be used to receive a frame;

3) L_SystemBroadcast.con shall be a local primitive generated by the local layer-2 for its own client to indicate the success of the transmission.

If the local user of layer-2 prepares a LSDU for the remote user it shall apply the L_SystemBroadcast.req primitive to pass the LSDU to the local layer-2. The local layer-2 shall accept the service request and shall try to send the LSDU to the remote layer-2 with the relevant frame format.

The local layer-2 shall pass a L_SystemBroadcast.con primitive to the local user that shall indicate either a correct or erroneous data transfer. Depending on whether a L2-acknowledgement is requested or not, this confirmation shall be related to the reception of the L2-acknowledgement or only to the transmission of the frame on the medium.

L_SystemBroadcast.req(destination_address, address_type, priority, octet_count, ack_request, lsdu)

destination_address: this parameter shall be used to indicate the destination address of the requested frame; it shall be the system broadcast address 0000h

address_type: this parameter shall be set to multicast

priority: this parameter shall be used to indicate the priority that shall be used to transmit the requested frame; it shall be “system”, “urgent”, “normal” or “low”

octet_count: this parameter shall be used to indicate the length information of the requested frame

ack_request: this parameter shall be used to indicate whether a layer-2 acknowledge is mandatory or optional

lsdu: this parameter shall be used to contain the user data to be transferred by layer-2

L_SystemBroadcast.con(source_address, destination_address, address_type, priority, octet_count, lsdu, l_status)

source_address this parameter shall be used to indicate the source address of the requested frame; it shall be the individual address of the device that requests the service primitive

destination_address: this parameter shall be used to indicate the destination address of the requested frame; it shall be the system broadcast address 0000h

address_type: this parameter shall be set to multicast

priority: this parameter shall be used to indicate the priority that shall be used to transmit the requested frame; it shall be “system”, “urgent”, “normal” or “low” octet_count: this parameter shall be used to indicate the length information of the requested frame

ack_request: this parameter shall be used to indicate whether a layer-2 acknowledge is mandatory or optional

l_status: ok: the value of this parameter shall be used to indicate that the transmission of the L_SystemBroadcast.req service has been successful

not_ok: the value of this parameter shall be used to indicate that the transmission of the L_SystemBroadcast.req service did not succeed

L_SystemBroadcast.ind(source_address, destination_address, address_type, priority, ack_request, octet_count, lsdu)

source_address: this parameter shall be used to indicate the source address of the received frame; it shall be the individual address of the device that has transmitted the service primitive

destination_address: this parameter shall be used to indicate the destination address of the received frame; it shall be the system broadcast address 0000h

address_type: this parameter shall be set to multicast

priority: this parameter shall be used to indicate the priority of the received frame; it shall be “system”, “urgent”, “normal” or “low”

ack_request: this parameter shall be used to indicate whether a layer-2 acknowledge is mandatory or optional

octet_count: this parameter shall be used to indicate the length information of the received frame

lsdu: this parameter shall be used to contain the user data that has been received by layer-2

5.3.4 L_Poll_Data service and protocol

The L_Poll_Data service is a confirmed multicast service. The local user of layer-2 shall use the L_Poll_Data.req primitive to request data from one or more remote users. The local layer-2 shall accept the service request and try to send the L_Poll_Data.req to the remote layer-2 with frame format 3. The destination address shall always be a poll group address. The poll group address shall be a parameter of layer-2.

L_Poll_Data request frames that are not correctly received shall be discarded.

After receiving a correct L_Poll_Data request frame with a poll_group_address equal to its own poll group addresses, the remote layer-2 shall respond with a single Poll_Data character. The remote layer-2 shall obtain the Poll_Data octet from its user with the L_Poll_Update.req primitive. The Poll_Data character shall be transmitted in the response slot associated with this device. The device’s response slot shall be a defined time slot in which the Poll_Data slave device shall transmit the Poll_Data character. The duration of a response slot shall be an idle time of 5 times followed by a single Universal Asynchronous Receiver Transmitter (UART) character.

EXAMPLE: If a device has the third response slot then the device has to wait for two Poll_Data characters transmitted by other devices, until the device is transmitting its Poll_Data character in the third response.

The response slot number shall be a parameter of layer-2.

A device shall not respond if its response slot number is larger than the number of expected poll data (nr_of_expected_poll_data) in the request frame.

The local layer-2 shall expect a number of Poll_Data characters from the poll group. If an expected Poll_Data character has not started after five bit times, the local layer-2 shall send a FILL (FEh) after six bit times. The remote layer-2 can therefore still count Poll_Data characters even if a member of the poll group doesn't respond.

The local layer-2 shall pass a L_Poll_Data.con primitive to the local user that contains the received Poll_Data and FILL octets or an information that the service failed.

The L_Poll_Data Service can only be applied between devices on a single physical segment. The number of expected Poll_Data characters is limited to 16.

L_Poll_Data.req(destination, nr_of_expected_poll_data)

destination: this parameter shall be used to indicate the destination address of the requested frame; it shall be a poll group address

nr_of_expected_poll_data: this parameter shall be used to indicate the number of expected poll data cycles

L_Poll_Data.con(l_status, poll_data_sequence)

l_status: ok: the value of this parameter shall be used to indicate that a valid poll_data_sequence has been composed

not_ok: the value of this parameter shall be used to indicate that it was not possible to compose a valid poll_data_sequence, i.e. collision occurred during transmission of a FILL, or at least one Poll_Data not correct

poll_data_sequence: this parameter shall be used to contain the sequence of Poll_Data octets and FILL octets

L_Poll_Update.req(Poll_Data)

Poll_Data: this parameter shall be used to contain the value of the Poll_Data octet that shall be transmitted in the L_Poll_Data_Response frame

L_Poll_Update.con(): indicates that the L_Poll_Update.req has been accepted by the local Layer-2

5.3.5 L_Busmon service

The L_Busmon service is a local Data Link service available only in busmonitor mode. It consists of the L_Busmon.ind primitive that transfers each received frame from the local layer-2 to the local layer-2 user.

L_Busmon.ind(L_Status, time_stamp, lpdu)

L_Status: this parameter shall be used to contain information whether a frame error, bit error or a parity error has been detected in the received frame. Additional information about the number of already received frames may also be contained

time_stamp: this parameter shall be used to contain the time information of the time when the first start bit of the frame has been received

lpdu: this parameter shall be used to contain all octets of the received frame

5.3.6 L_Service_Information service

The L_Service_Information service is a local Data Link service available in Data Link Normal Mode. It consists of the L_Service_Information.ind primitive.

L_Service_Information.ind():
this service primitive shall be used to indicate that a frame has been received which contained the individual address of the local layer-2 as source address

5.4 Data link layer protocol

5.4.1 Protocol

The data link layer shall offer a reliable frame transfer service between devices on the same subnetwork. This means that corrupted frames shall be detected and retransmitted (i.e. repeated) for a number of times and that only information of correctly received frames shall be presented to the data link layer user.

The maximum frame length that is supported may be adapted to the needs of a device.

Frames that are corrupted or that exceed the reception capacity of a device shall be discarded.

Some means may be provided to prevent from information duplication (i.e. that this information is not presented several times to the data link layer user).

Most of these functions are done in a medium dependent way, as specified in ISO/IEC 14543-3-5, ISO/IEC 14543-5-6 and ISO/IEC 14543-3-7 (EN 50090-5 series) respectively.

5.4.2 Recommendations for duplication prevention

Some means to prevent duplication are available on certain media. However, there is always a remaining risk of duplication.

The following guide-lines to handle this can be taken into account.

– Noise at the medium should be reduced as much as necessary to avoid corrupted acknowledgements.

– The medium, transmission method and transmitter and receiver technology shall provide basic noise immunity. [See the media descriptions in ISO/IEC 14543-3-5, ISO/IEC 14543-5-6 and ISO/IEC 14543-3-7 (EN 50090-5 series) respectively]

– During internal or external user application programming, be aware of the possibility that in some cases a duplicated L_Data service may occur. This will be rare on media with collision avoidance but far more common on media without collision prevention or detection or media that are more noise sensitive.

– Excessive use of the priority ‘urgent’ is not recommended on certain media.

5.5 Parameters of layer-2

The following parameters influence the behaviour of layer-2 and shall be implemented inside layer-2 in order to operate correctly:

– domain address	domain address of the network the device belongs to (on open media);
– individual address	unique individual address of this device;
– group address table	table with the group address(es) of this device;
– nack_retry	defines the number of retries in case of a negative acknowledge (NACK) response or a acknowledgement time-out;
– busy_retry	defines the number of retries in case of a BUSY/FULL response;
– poll group address	the poll group address of this device;

– response slot number	the response slot number of this device (for polling mode);
– data link layer mode	either the normal mode or the busmonitor mode of the data link layer.

5.6 Specific devices

5.6.1 Layer-2 of a bridge

A bridge connects two segments of one single subnetwork.

NOTE 1 The notion of bridge provides only limited interest. It is not able to correctly handle the L2-acknowledgement, without detailed knowledge of the network.

A bridge shall forward frames and L2-acknowledgements shall be sent on each side of the bridge depending on the destination address. The bridge shall know which device addresses are used on each side of the bridge. All other layer-2 services shall be ignored.

NOTE 2 Sending back an L2-acknowledgement cannot be done by the devices through the bridge. Delays will be incompatible with layer-2 protocol on most media. Systematic sending of L2-acknowledgement and forwarding of frames through the bridge (i.e. as a repeater with delay) would inhibit the L2-ack functionality. Mechanisms relying on this function to check addresses available would be fooled. That is why such bridges are not allowed.

A bridge does not need to have an individual address. A bridge may have an individual address to allow setting parameters.

5.6.2 Layer-2 of a router

A router is a device that interconnects a hierarchically higher subnetwork and a hierarchically lower subnetwork.

The layer-2 of a router shall respond to a correct L_Data.request frame on either one of the three following conditions.

a) The value of the destination address is listed in the routing_table ().

b) The destination address is an individual address that indicates that the destination is on the other side of the router.

c) The destination address is equal to the individual address of the router.

In these cases, the L_Data.ind service primitive shall be passed to the layer-3. All other layer-2 services shall be ignored.

6 Requirements for the network layer

6.1 Functions of the network layer

The network layer (layer-3) is the layer between the data link layer and the network layer user. This network layer shall comply to the definitions of the ISO/OSI model (ISO/IEC 7498) network layer.

The network layer shall use the L_Data and the L_SystemBroadcast service of the Data link layer and shall provide N_Data_Individual, N_Data_Group, N_Data_Broadcast and N_Data_SystemBroadcast services to the network layer user, see Figure 7.

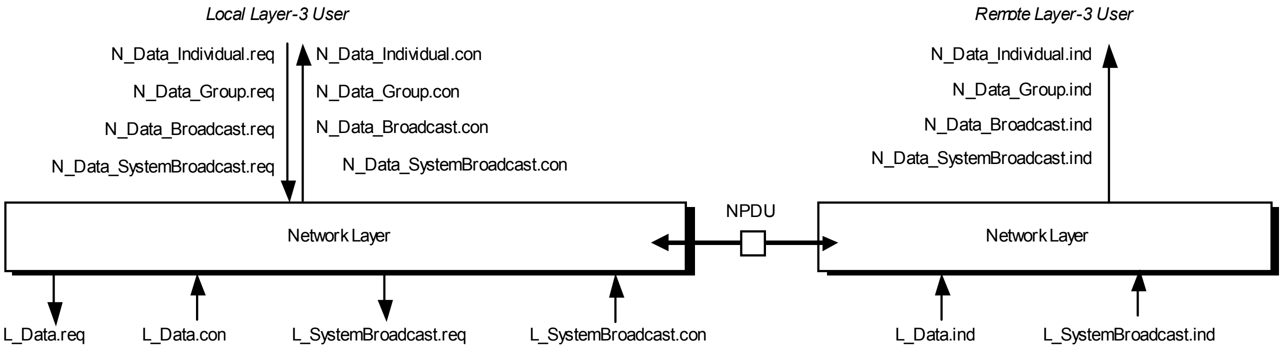

Figure 7 – Interaction of the network layer (not for bridges or routers)

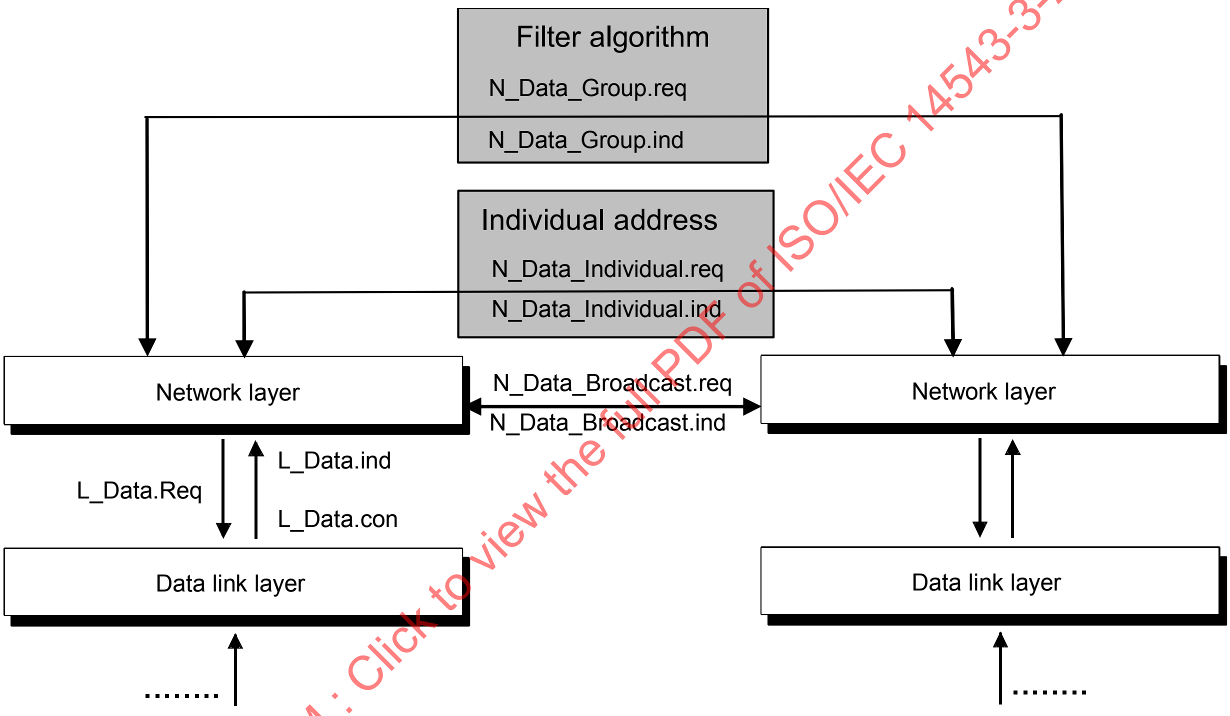

Figure 8 – General functioning of a router or a bridge

Communication across subnetworks with filter characteristics shall use devices called routers, as specified in 6.4.4. Routers are devices that allow to couple two link layer protocol instances together, which are connected to different subnetworks. For routing frames from one subnetwork to the other the router shall use a filter algorithm. Furthermore, a router shall remove misrouted messages so that they cannot flood the network. All the filter algorithms of a network together define the allowed communication paths between any two devices.

Communication across subnetworks without filter characteristics needs devices called bridges, see also Figure 8. Like a router a bridge allows to couple two Data link layer protocol instances together, which are connected to different subnetworks. But a bridge shall not have the filter property of a router and therefore shall not require any filter algorithm.

Two different mechanisms for routing are used. For group addressing a filter algorithm shall be used. For individual addressing the routing shall be done by interpreting the individual address of the received frame.

Two different network layer users shall be distinguished:

a) the network layer user in a standard device; this is the transport layer, as specified in Clause 7;

b) the network layer user in a router; this is the filter algorithm.

6.2 Network layer services and protocol

6.2.1 Network layer protocol data unit (NPDU)

The NPDU shall correspond to the LPDU of an L_Data-Frame without the control field, source address, destination address, address type flag and the octet count.

An example of a NPDU that can be used is shown in Figure 9 below.

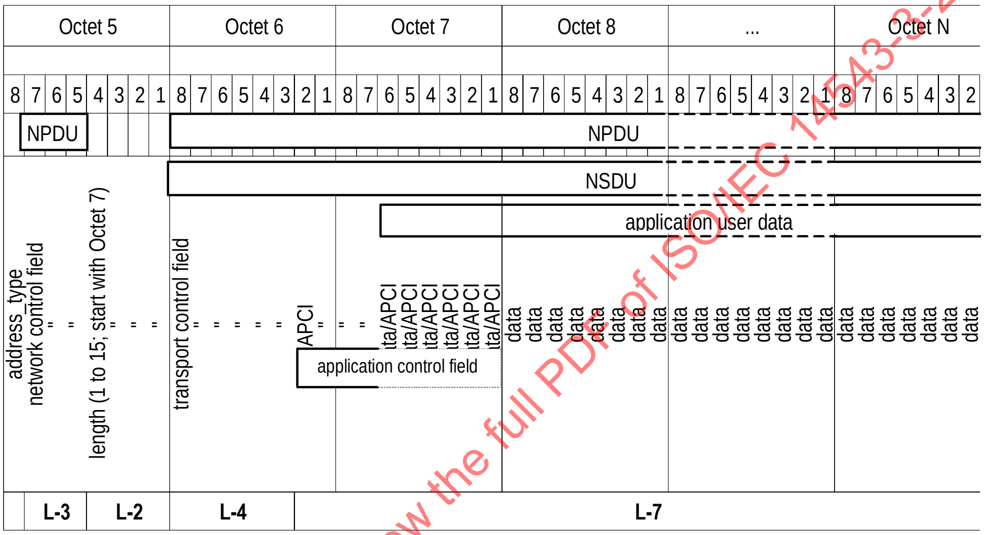

Figure 9 – Format of the NPDU (example)

6.2.2 Network layer services

6.2.2.1 N_Data_Individual service

The local user of network layer shall prepare a network layer service data unit (NSDU) for the remote user of network layer. The destination shall be addressed with an individual address. The local user of network layer shall apply the N_Data_Individual.req primitive to pass the NSDU to the local network layer. The local network layer shall accept the service request and shall pass it with a L_Data.req with address_type = 'individual' to the local link layer.

The local network layer shall encode the NSDU to the NPDU by adding the hop_count with the value according to the parameter hop_count_type and mapping the arguments ack_request, destination_address, octet_count and priority and to the corresponding arguments ack_request, destination_address, octet_count and priority of the L_Data.req primitive.

The remote network layer shall map an L_Data.ind primitive with address_type = 'individual' to an N_Data_Individual.ind primitive. It shall remove the hop_count and shall generate the parameter hop_count_type according to its value. The argument lsdu shall be decoded to the argument nsdu. The argument's octet_count, priority and source_address shall be mapped to the corresponding argument's octet_count, priority and source_address of the N_Data_Individual.ind primitive.

The local network layer shall map the L_Data.con primitive to the N_Data_Individual.con primitive. The argument l_status shall be mapped to the corresponding argument n_status of the N_Data_Individual.con primitive.

N_Data_Individual.req(ack_request, destination_address, hop_count_type, octet_count, priority, nsdu)

ack_request: this parameter shall be used to indicate whether a layer-2 acknowledge is mandatory or optional

destination_address: this parameter shall be used to indicate the destination address of the requested frame; it shall be the individual address of the destination

hop_count_type: this parameter shall be used to indicate whether the hop_count shall be set to 7 or if the network layer parameter shall be used

octet_count: this parameter shall be used to indicate the length information of the requested frame

priority: this parameter shall be used to indicate the priority that shall be used to transmit the requested frame; it shall be “system”, “urgent”, “normal” or “low”

nsdu: this parameter shall be used to contain the user data that shall be transferred by the network layer

N_Data_Individual.con(ack_request, destination_address, hop_count_type, octet_count, priority, nsdu, n_status)

ack_request: this parameter shall be used to indicate whether a layer-2 acknowledge has been indicated as mandatory or optional in the transmitted frame

destination_address: this parameter shall be used to indicate the destination address of the transmitted frame; it shall be the individual address of the destination

hop_count_type: this parameter shall be used to indicate whether the hop_count of the transmitted frame has been set to 7 or if the network layer parameter has been used

octet_count: this parameter shall be used to indicate the length information of the transmitted frame

priority: this parameter shall be used to indicate the priority that has been used to transmit the requested frame; it shall be “system”, “urgent”, “normal” or “low”

nsdu: this parameter shall be used to contain the user data that has been transferred by the network layer

n_status: ok: the value of this parameter shall be used to indicate that the transmission of the N_Data_Individual has been successful

not_ok: the value of this parameter shall be used to indicate that the transmission of the N_Data_Individual did not succeed

N_Data_Individual.ind(destination_address, hop_count_type, octet_count, priority, source_address, nsdu)

destination_address: this parameter shall be used to contain the destination address of the received frame; it shall be the individual address of this device

hop_count_type: this parameter shall be used to indicate whether the hop count of the received frame equals to 7 or not

octet_count: this parameter shall be used to contain the length information of the received frame

priority: this parameter shall be used to indicate the priority of the received frame; it shall be “system”, “urgent”, “normal” or “low”

source_address: this parameter shall be used to indicate the source address of the received frame; it shall be the individual address of the device that has transmitted the N_Data_Individual service

nsdu: this parameter shall be used to contain the user data that has been received by the network layer

6.2.2.2 N_Data_Group service

The N_Data_Group service shall be confirmed locally. The local user of network layer shall prepare a NSDU for the remote user of network layer, the destination shall be addressed with a group address. The local user of the network layer shall apply the N_DATA_GROUP.req primitive to pass the NSDU to the local network layer. The local network layer shall accept the service request and shall pass it with a L_Data.req with address_type = multicast to the local Data link layer.

The local network layer shall encode the NSDU to the LSDU by adding the hop_count with the value according to the parameter hop_count_type and mapping the arguments ack_request, destination_address, octet_count and priority to the corresponding arguments ack_request, destination_address, octet_count and priority of the L_Data.req primitive.

The remote network layer shall map a L_Data.ind primitive with address_type = multicast and destination_address<>‘0’ to a N_Data_Group.ind primitive. It shall remove the hop_count and shall generate the parameter hop_count_type according to its value. The arguments destination_address, octet_count and priority shall be mapped to the corresponding arguments destination_address, octet_count and priority of the N_Data_Group.ind primitive.

The local network layer shall map the L_Data.con primitive to the N_Data_Group.con primitive. The argument l_status shall be mapped to the corresponding argument n_status of the N_Data_Group.con primitive.

N_Data_Group.req(ack_request, destination_address, hop_count_type, octet_count, priority, nsdu)

ack_request: this parameter shall be used to indicate whether a layer-2 acknowledge is mandatory or optional

destination_address: this parameter shall be used to indicate the destination address of the requested frame; it shall be a group address

hop_count_type: this parameter shall be used to indicate whether the hop_count shall be set to 7 or if the network layer parameter shall be used

octet_count: this parameter shall be used to indicate the length information of the requested frame

priority: this parameter shall be used to indicate the priority that shall be used to transmit the requested frame; it shall be “system”, “urgent”, “normal” or “low”

nsdu: this parameter shall be used to contain the user data that shall be transferred by the network layer

N_Data_Group.con(ack_request, destination_address, hop_count_type, octet_count, priority, nsdu, n_status)

ack_request: this parameter shall be used to indicate whether a layer-2 acknowledge has been indicated as mandatory or optional in the transmitted frame

destination_address: this parameter shall be used to indicate the destination address of the transmitted frame; it shall be a group address

hop_count_type: this parameter shall be used to indicate whether the hop_count of the transmitted frame has been set to 7 or if the network layer parameter has been used octet_count: this parameter shall be used to indicate the length information of the transmitted frame

priority: this parameter shall be used to indicate the priority that has been used to transmit the requested frame; it shall be “system”, “urgent”, “normal” or “low”

nsdu: this parameter shall be used to contain the user data that has been transferred by the network layer

n_status: ok: the value of this parameter shall be used to indicate that the transmission of the N_Data_Group has been successful

not_ok: the value of this parameter shall be used to indicate that the transmission of the N_Data_Group did not succeed

N_Data_Group.ind(destination_address, hop_count_type, octet_count, priority, nsdu)

destination_address: this parameter shall be used to contain the destination address of the received frame; it shall be the addressed group address of this device

hop_count_type: this parameter shall be used to indicate whether the hop count of the received frame equals 7 or not

octet_count: this parameter shall be used to contain the length information of the received frame

priority: this parameter shall be used to indicate the priority of the received frame; it shall be “system”, “urgent”, “normal” or “low”

nsdu: this parameter shall be used to contain the user data that has been received by the network layer

6.2.2.3 N_Data_Broadcast service

The local user of network layer shall prepare a NDSU for all the remote network layer users within the same domain, the destination shall be addressed with the broadcast address (destination address = ´0´ and address_type = multicast). The local user of the network layer shall apply the N_Data_Broadcast.req primitive to pass the NSDU to the local network layer. The local network layer shall accept the service request and shall pass it with a L_Data.req with address_type = multicast to the local Data link layer.

The local network layer shall encode the NSDU to the LSDU by adding the hop_count with the value according to the parameter hop_count_type and mapping the arguments ack_request octet_count and priority to the corresponding arguments ack_request, octet_count and priority of the L_Data.req primitive and setting the L_Data.req parameter destination_address to ‘0’.

The remote network layer shall map a L_Data.ind primitive with address_type = multicast and destination address = ´0´ to a N_Data_Broadcast.ind primitive. It shall remove the hop_count and shall generate the parameter hop_count_type according to its value. The argument lsdu shall be mapped to the argument nsdu. The argument priority shall be mapped to the corresponding argument priority of the N_Data_Broadcast.ind primitive.

The local network layer shall map the L_Data.con primitive to the N_Data_Broadcast.con primitive. The argument l_status shall be mapped to the corresponding argument n_status of the N_Data_Broadcast.con primitive.

N_Data_Broadcast.req(ack_request, hop_count_type, octet_count, priority, nsdu)

ack_request: this parameter shall be used to indicate whether a layer-2 acknowledge is mandatory or optional

hop_count_type: this parameter shall be used to indicate whether the hop_count shall be set to 7 or if the network layer parameter shall be used octet_count: this parameter shall be used to indicate the length information of the requested frame

priority: this parameter shall be used to indicate the priority that shall be used to the transmit the requested frame; it shall be “system”, “urgent”, “normal” or “low”

nsdu: this parameter shall be used to contain the user data that shall be transferred by the network layer

N_Data_Broadcast.con(ack_request, hop_count_type, octet_count, priority, nsdu, n_status)

ack_request: this parameter shall be used to indicate whether a layer-2 acknowledge has been indicated as mandatory or optional in the transmitted frame

hop_count_type: this parameter shall be used to indicate whether the hop_count of the transmitted frame has been set to 7 or if the network layer parameter has been used

octet_count: this parameter shall be used to indicate the length information of the transmitted frame

priority: this parameter shall be used to indicate the priority that has been used to transmit the requested frame; it shall be “system”, “urgent”, “normal” or “low”

nsdu: this parameter shall be used to contain the user data that has been transferred by the network layer

n_status: ok: the value of this parameters shall be used to indicate that the transmission of the N_Data_Broadcast has been successful

not_ok: the value of this parameters shall be used to indicate that the transmission of the N_Data_Broadcast did not succeed

N_Data_Broadcast.ind(hop_count_type, octet_count, priority, source_address, nsdu)

hop_count_type: this parameter shall be used to indicate whether the hop count of the received frame is equal to 7 or not

octet_count: this parameter shall be used to contain the length information of the received frame

priority: this parameter shall be used to indicate the priority of the received frame; it shall be “system”, “urgent”, “normal” or “low”

source_address: this parameter shall be used to indicate the source address of the received frame; it shall be the individual address of the device that has transmitted the N_Data_Broadcast service

nsdu: this parameter shall be used to contain the user data that has been received by the network layer

6.2.2.4 N_Data_SystemBroadcast service

The local user of Network Layer shall prepare a NSDU for all the remote Network Layer users. The destination shall be addressed with the system broadcast address (destination address = 0000h and address_type = multicast). The local user of network layer shall apply the N_Data_SystemBroadcast.req primitive to pass the NSDU to the local network layer. The local network layer shall accept the service request and pass it with a L_SystemBroadcast.req with address_type = multicast to the local data link layer.

The local network layer shall encode the NSDU to the LSDU by adding the hop_count with the value according to the parameter hop_count_type and mapping the arguments ack_request octet_count and priority to the corresponding arguments ack_request, octet_count and priority of the L_SystemBroadcast.req primitive and setting the L_SystemBroadcast.req parameter destination_address to 0000h.

The remote network layer shall map a L_SystemBroadcast.ind primitive with address_type = multicast and destination_address = 0000h to a N_Data_SystemBroadcast.ind primitive. It shall remove the hop_count and generate the parameter hop_count_type according to its value. The argument lsdu shall be mapped to the argument nsdu. The argument priority shall be mapped to the corresponding argument priority of the N_Data_SystemBroadcast.ind primitive.

The local network layer shall map the L_SystemBroadcast.con primitive to the N_Data_SystemBroadcast.con primitive. The argument l_status shall be mapped to the corresponding argument n_status of the N_Data_SystemBroadcast.con primitive.

N_Data_SystemBroadcast.req(ack_request, hop_count_type, octet_count, priority, nsdu)

ack_request: this parameter shall be used to indicate whether a layer-2 acknowledge is mandatory or optional

hop_count_type: this parameter shall be used to indicate whether the hop_count shall be set to 7 or whether the network layer parameter shall be used

octet_count: this parameter shall be used to indicate the length information of the requested frame

priority: this parameter shall be used to indicate the priority that shall be used to transmit the requested frame; it shall be “system”, “urgent”, “normal” or “low”

nsdu: this parameter shall be used to contain the user data that shall be transferred by the network layer

N_Data_SystemBroadcast.con(ack_request, hop_count_type, octet_count, priority, nsdu, n_status)

ack_request: this parameter shall be used to indicate whether a layer-2 acknowledge has been indicated as mandatory or optional in the transmitted frame

hop_count_type: this parameter shall be used to indicate whether the hop_count of the transmitted frame has been set to 7 or whether the network layer parameter has been used

octet_count: this parameter shall be used to indicate the length information of the transmitted frame

priority: this parameter shall be used to indicate the priority that has been used to transmit the requested frame; it shall be “system”, “urgent”, “normal” or “low”

nsdu: this parameter shall be used to contain the user data that has been transferred by the network layer

n_status: ok: the value of this parameters shall be used to indicate that the transmission of the N_Data_SystemBroadcast has been successful

not_ok: the value of this parameters shall be used to indicate that the transmission of the N_Data_SystemBroadcast.req did not succeed

N_Data_SystemBroadcast.ind(hop_count_type, octet_count, priority, source_address, nsdu)

hop_count_type: this parameter shall be used to indicate whether the hop count of the received frame equals 7 or not

octet_count: this parameter shall be used to contain the length information of the received frame

priority: this parameter shall be used to indicate the priority of the received frame; it shall be “system”, “urgent”, “normal” or “low”

source_address: this parameter shall be used to indicate the source address of the received frame; it shall be the individual address of the device that has transmitted the N_Data_SystemBroadcast service

nsdu: this parameter shall be used to contain the user data that has been received by the network layer

6.3 Parameters of the network layer

The following parameters influence the behaviour of network layer and shall be implemented inside the network layer in order to operate correctly:

hop_count shall be added to the frame by the network layer;

device_type information about the device; either normal device or bridge or router.

6.4 Network layer state machines

6.4.1 Overview

Bridges and routers do also have a network layer but their network layer state machine differs from normal devices.

6.4.2 State machine of network layer for normal devices

The state machine of network layer for normal devices shall map services as described in 6.2.2. The value of the hop count shall be added when the transport layer applies a network layer request primitive.

In sending direction, the network layer shall set the value of the hop_count field in the NSDU to the value according to the value service primitive parameter ‘hop_count_type’ of the request service primitive as follows:

- hop_count_type = ‘7’ the hop_count shall be set to the value 7;
- hop_count_type = ‘network layer Parameter’ the hop_count shall be set to the value of the network layer parameter ‘hop_count’ (see 6.3).

In reception direction, the network layer shall set the service parameter hop_count_type of the indication- and confirmation service primitives according to the value of the hop_count field in the received NSDU:

- hop_count = 7 the hop_count_type shall be set to ‘equal to 7’;
- hop_count ≠ 7 the hop_count_type shall be set to ‘not equal to 7’.

6.4.3 State machine of network layer for bridges

If an L_Data.ind with a hop_count in [1...6] is received, the bridge shall decrement the hop_count and shall transmit the service parameters of the L_Data.ind with the corresponding service parameters (source address, destination_address, address_type, priority, ack_request, octet_count, lsdu) of an L_Data.req to the other side.

If an L_Data.ind with a hop_count value of seven is received, the bridge shall transmit the service parameters of the L_Data.ind with the corresponding service parameters of a L_Data.req to the other side.

Otherwise the network layer of the bridge shall discard the L_Data.ind.

6.4.4 State machine of network layer for routers

6.4.4.1 General

If an L_Data.ind with address_type = multicast and hop_count in [1...6] is received and the filter condition for the destination address is true, the router shall decrement the hop_count and shall transmit the service parameters of the L_Data.ind with the corresponding service parameters of a L_Data.req to the other side.

If an L_Data.ind with address_type = multicast and hop_count equal to seven is received, the router shall transmit the service parameters of the L_Data.ind with the corresponding service parameters of a L_Data.req to the other side.

If an L_Data.ind with address_type = 'individual' and destination address equal to the individual address of the router is received, the router shall process the L_Data.ind identical to a normal device, see 6.2.2.1.

If a L_Data.ind with address_type = 'individual' is received and the destination address matches the conditions for routing, the router shall transmit the service parameters of the L_Data.ind with the corresponding service parameters of a L_Data.req to the other side. Additionally, if the routing counter value was in [1..6] the router shall decrement it.

Otherwise the Network layer of the router shall discard the L_Data.ind.

An N_Data_Individual.req service invoked by the network layer user at the router shall be processed as described in 6.2.2.1.

Symbols for the following subclauses within 5.4.4:

C hop count value contained in the N-protocol header

D low order octet of the destination address, i.e. device address part

G group address

SD low nibble of high order octet plus low order octet, i.e. Line Address + Device Address

Z high nibble of high order octet of the destination address, i.e. Area Address

ZS high order octet of the destination address, i.e. hierarchy information part: Area Address + Line Address

6.4.4.2 Detailed routing algorithm

For a router there shall be five possible courses of action in response to a received LSDU:

a) ROUTE_UNMODIFIED: The LSDU shall be routed from the first subnetwork to another subnetwork without modification of the hop count value. The LSDU shall be routed to the second subnetwork after an Acknowledge (ACK) character shall be sent back to the originator of the LSDU;

b) ROUTE_DECREMENTED: The LSDU shall be routed from the first subnetwork to another subnetwork after the hop count value shall be decremented. The LSDU shall be routed to the second subnetwork after an ACK character shall be sent back to the originator of the LSDU;

c) FORWARD_LOCALLY: The LSDU shall be processed to an NSDU and shall be given to the local network layer user after an ACK character shall be sent back to the originator of the LSDU;

d) IGNORE_TOTALLY: The LSDU shall be ignored; no acknowledgement shall be sent back to the originator of the LSDU;

e) IGNORE_ACKED: The LSDU shall be ignored, nevertheless an ACK character shall be sent back to the originator of the LSDU.

The following subclauses specify the routing algorithm for a router, which can either be a line coupler or a backbone coupler, depending on its position in the topology.

6.4.4.3 Routing in case of a group destination address

```
if      routing condition = TRUE and $0_H < C < 7_H$ then ROUTE_DECREMENTED
if      routing condition = TRUE and $C = 0_H$        then IGNORE_ACKED[1]
elsif   $C = 7_H$  then ROUTE_UNMODIFIED
else    IGNORE_TOTALLY
```

6.4.4.4 Routing in case of an individual destination address: line coupler

6.4.4.4.1 Main line to subline routing

```
if      ZS = own subline address then
        if D <> $00_H$ then
             if      $C = 7_H$            then ROUTE_UNMODIFIED
             elsif   $0_H < C < 7_H$      then ROUTE_DECREMENTED
             else    IGNORE_ACKED
        else FORWARD_LOCALLY
else    IGNORE_TOTALLY
```

6.4.4.4.2 Subline to main line routing

```
if      ZS <> own subline address then
        if      $C = 7_H$          then ROUTE_UNMODIFIED
        elsif   $0_H < C < 7_H$    then ROUTE_DECREMENTED
        else    IGNORE_ACKED
elseif D = $00_H$   then FORWARD_LOCALLY
else    IGNORE_TOTALLY
```

6.4.4.5 Routing in case of the destination address is an individual address: backbone router

6.4.4.5.1 Backbone line to main line routing

```
if      Z = own area address then
        if      SD <> $00_H$ then
                if     $C = 7_H$            then ROUTE_UNMODIFIED
                elsif  $0_H < C < 7_H$      then ROUTE_DECREMENTED
                else   IGNORE_ACKED
        else FORWARD_LOCALLY
else    IGNORE_TOTALLY
```

[1] The ACK is sent by the Data link layer.

6.4.4.5.2 Main line to backbone routing

```
if      Z <> own area address then
            if      C = 7H                then ROUTE_UNMODIFIED
            elsif   0H < C < 7H           then ROUTE_DECREMENTED
            else    IGNORE_ACKED
elseif SD = 00H then FORWARD_LOCALLY
else  IGNORE_TOTALLY
```

6.4.4.6 Routing in case the destination address is a broadcast address

```
if      C = 7H                            then ROUTE_UNMODIFIED
elsif   0H < C < 7H                           then ROUTE_DECREMENTED
else    IGNORE_ACKED
```

7 Requirements for the transport layer

7.1 Functionality of the transport layer

The transport layer (Layer-4) shall provide data transmission over different communication modes. These modes shall connect transport layer users with each other. The transport layer shall provide five different communication modes:

a) point-to-multipoint, connection-less (multicast);

b) point-to-domain, connection-less (broadcast);

c) point-to-all-points, connection-less (system broadcast);

d) point-to-point, connection-less;

e) point-to-point, connection-oriented.

Every communication mode shall provide specific transport layer Service Access Points (TSAPs) accessible via different transport layer services, see Figure 10. Each of these services shall consist of three service primitives, the request (req), the confirm (con) and the indication (ind).

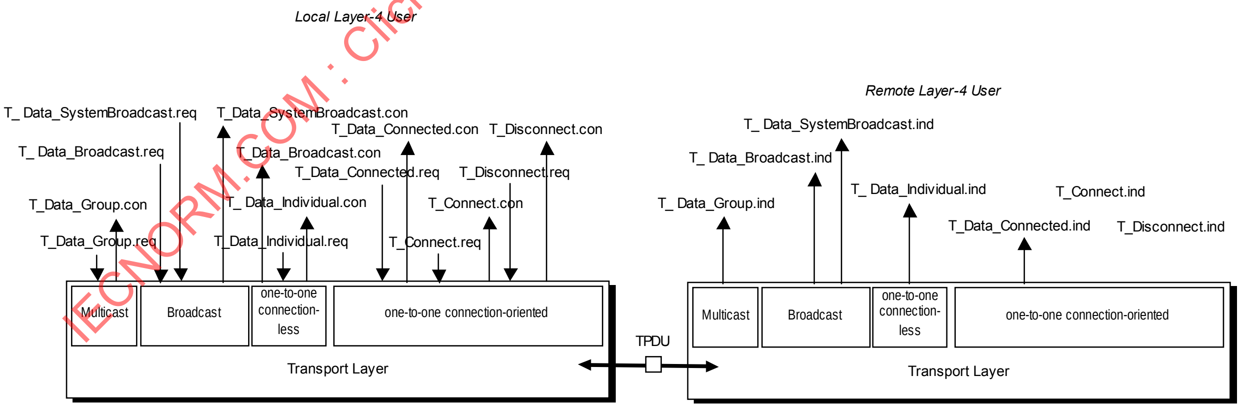

Figure 10 – Interaction of the transport layer

7.2 Transport layer Protocol Data Unit (TPDU)

An example of a TPDU that can be used is shown in Figure 11.

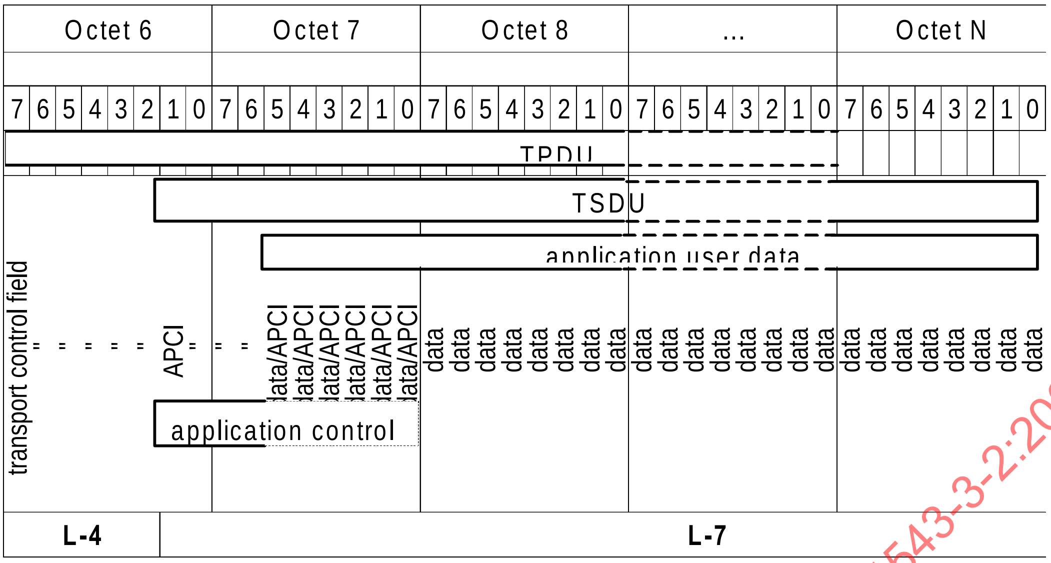

Figure 11 – Format of the TPDU (Example)

The TPDU shall correspond to the NPDU, but reduced by the network control field. The transport control field shall be encoded and decoded by transport layer and shall contain the transport layer service codes and the sequence_number, see Figure 12:

Octet 5								Octet 6								
								transport ctrl field								
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
Address Type (AT)								Data/Control Flag	Numbered							
1								0	0	0	0	0	0			T_DATA_BROADCAST_REQ_PDU (Destination_Address=0)
1								0	0	0	0	0	0			T_DATA_GROUP_REQ_PDU (Destination_Address<>0)
1								0	0	0	0	0	1			T_DATA_TAG_GROUP_PDU
0								0	0	0	0	0	0			T_DATA_INDIVIDUAL_REQ_PDU
0								0	1	SeqNo	SeqNo	SeqNo	SeqNo			T_DATA_CONNECTED_REQ_PDU
0								1	0	0	0	0	0	0	0	T_CONNECT_REQ_PDU
0								1	0	0	0	0	0	0	1	T_DISCONNECT_REQ_PDU
0								1	1	SeqNo	SeqNo	SeqNo	SeqNo	1	0	T_ACK_PDU

Figure 12 – Transport control field

The encoding 0x0BF is reserved and shall not be used for future extensions.

7.3 Overview communication modes

7.3.1 Point-to-multipoint, connection-less (multicast) communication mode

The multicast communication mode shall allow communication between devices belonging to the same group. A device can be member of multiple groups. Any member of the group may be the initiator of a communication. The group shall be identified by its group address. The multicast communication mode on transport layer shall map TSAPs (group index) to group addresses and vice versa.

In Point-to-Multipoint Connectionless (multicast) Communication Mode exclusively the following transport layer services shall be available:

T_Data_Group

T_Data_Tag_Group

7.3.2 Point-to-domain, connection-less (broadcast) communication mode

The broadcast communication mode shall connect a single device with all other devices within the same domain. The single device shall always be a transmitter, all the other devices shall always be receivers.

The following transport layer services shall exclusively be available on the Point-to-domain Connection-less (Broadcast) Communication Mode:

T_Data_Broadcast.

7.3.3 Point-to-all-points, connection-less (SystemBroadcast) communication mode

The System Broadcast communication mode shall connect a single device with all communication partners. The single end device shall always be a transmitter, the communication partners shall always be receivers.

The following transport layer service shall be available on the Point-to-All-Points System Broadcast communication mode:

T_Data_SystemBroadcast

7.3.4 Point-to-point, connection-less communication mode

The Point-to-Point connectionless communication mode shall enable communication between any two single devices.

The following transport layer services shall exclusively be available on the Point-to-Point, Connection-less Communication Mode:

T_Data_Individual.

7.3.5 Point-to-point, connection-oriented communication mode

The Point-to-point, connection-oriented communication mode shall allow a reliable communication between any two single devices, within a connection.

The following transport layer services shall exclusively be available on the point-to-point, connection-oriented communication mode:

T_Connect;

T_Data_Connected;

T_Disconnect.

The user of this communication mode type shall establish the connection before this connection can be used. The user should release the connection if it is no longer needed. The transport layer shall provide a supervision of the connection with a connection-time-out-timer. If the timer expires or if an unrecoverable error occurs, the transport layer shall release the connection immediately. The transport layer shall also provide a reliable end-to-end transmission over bridges and routers on the connection-oriented communication mode. T_Data_Connected services shall be repeated up to three times if the T_Data_Connected.req is not acknowledged from the remote transport layer entity within an acknowledgement-time-out-time. Repetitions of T_Data_Connected services shall be detected using a sequence number. Parallel services shall not be allowed on a connection-oriented communication mode.

The connection-oriented communication mode shall be processed according to the transport layer state machine described in 7.6.

7.3.6 Algorithm for the identifier of communication

The TSAP used in the T_Data_Group service shall be a reference to a group address, see 7.4.2.

The TSAP used in the T_Data_Individual service shall be the individual address of the communication partner.

The TSAP used in connection-oriented transport communication shall be the identifier of the connection address. The number of possible connections at a time depends on the available internal resources.

7.4 Transport layer services

7.4.1 General

All the transport layer services shall provide a confirmation to the requester (user of transport layer). The confirmation of the T_Data_Connected service shall indicate that the remote transport layer entity did acknowledge the T_Data_Connected.req. The confirmation of the other transport layer services shall be caused by the local confirmation of the network layer.

The user of transport layer shall not request a service primitive before the preceding request is confirmed by transport layer, i.e. no parallel services shall be allowed.

7.4.2 T_Data_Group service

The T_Data_Group service shall be applied by the user of the transport layer, to transmit a TSDU over a multicast communication mode to one or more remote partners. The T_Data_Group service shall neither be acknowledged nor confirmed by the remote transport layer entity. The confirmation shall be a local confirmation that shall be caused by the N_Data_Group.con of the network layer.

The local user of transport layer shall prepare a TSDU for the remote user. The local user of transport layer shall apply the T_Data_Group.req primitive to pass the TSDU to the local transport layer. The destination shall be defined by the TSAP.

The local transport layer shall accept the service request, shall map the TSAP to the destination group address, shall map the arguments to the corresponding arguments of the N_Data_Group.req primitive, shall encode the TSDU to the NSDU and shall pass it with a N_Data_Group.req to the local network layer.

The remote transport layer shall map a N_Data_Group.ind primitive to a T_Data_Group.ind primitive. The remote transport layer shall map the destination_address to the TSAP. The argument NSDU shall be mapped to the argument TSDU, the argument priority shall be mapped to the corresponding argument priority of the T_Data_Group.ind primitive.

If the local transport layer receives an N_Data_Group.con from the local network layer it shall pass a T_Data_Group.con to the local transport layer user. If the confirmation is positive (n_status = ok), the local transport layer shall pass a positive T_Data_Group.con (t_status = ok) to the local user. If the confirmation is negative (n_status = not_ok), the local transport layer shall pass a T_Data_Group.con (t_status = not_ok) to the local user indicating that the transmission of the associated N_Data_Group.req did not succeed.

T_Data_Group.req(ack_request, hop_count_type, octet_count, priority, TSAP, tsdu)

ack_request: Data link layer Ack requested or don't care

hop_count_type: this parameter shall be used to indicate whether the hop_count shall be set to 7 or if the network layer parameter shall be used

octet_count: length information as described in Data link layer

priority: system, urgent, normal or low priority

TSAP: identifier of the service access point

tsdu: this is the user data to be transferred by transport layer

T_Data_Group.con(ack_request, hop_count_type, octet_count, priority, source_address, TSAP, tsdu, t_status)

ack_request: Data link layer Ack requested or don't care

hop_count_type: this parameter shall be used to indicate whether the hop_count shall be set to 7 or if the network layer parameter shall be used

octet_count: length information as described in Data link layer

priority: system, urgent, normal or low priority

source_address: the individual address of the originator of the message

TSAP: identifier of the service access point

tsdu: this is the user data that has been transferred by the transport layer

t_status: ok: T_Data_Group sent successfully with N_Data_Group service

not_ok: transmission of the associated N_Data_Group request frame didn't succeed

T_Data_Group.ind(hop_count_type, octet_count, priority, source_address, TSAP, tsdu)

hop_count_type: hop count equals 7 or not

octet_count: length information as described in Data link layer

priority: system, urgent, normal or low priority

source_address: the individual address of the originator of the message

TSAP: identifier of the service access point

tsdu: this is the user data that has been transferred by the transport layer

7.4.3 T_Data_Tag_Group service

The T_Data_Tag_Group service shall be applied by the user of Transport Layer, to transmit a TSDU over a multicast communication mode to one or more remote partners. The T_Data_Tag_Group service shall be neither acknowledged nor confirmed by the remote Transport Layer entity. The confirmation shall be a local confirmation caused by the N_Data_Group.con of the Network Layer.

The local user of Transport Layer shall prepare a TSDU for the remote user. The local user of Transport Layer shall apply the T_Data_Tag_Group.req primitive to pass the TSDU to the local Transport Layer. The destination shall be defined by the Destination Address and the Frame Format Parameter.

The local Transport Layer shall accept the service request, map the arguments to the corresponding arguments of the N_Data_Group.req primitive, encode the TSDU to the NSDU and pass it with a N_Data_Group.req to the local Network Layer.

The remote Transport Layer shall map a N_Data_Group.ind primitive to a T_Data_Tag_Group.ind primitive. The argument NSDU shall be mapped to the argument TSDU, the argument priority shall be mapped to the corresponding argument priority of the T_Data_Tag_Group.ind primitive.

Prior to passing a T_Data_Tag_Group.con primitive to the local user, the local Transport Layer shall receive a N_Data_Group.con from the local Network Layer. If the confirmation is positive (n_status = ok), the local Transport Layer shall pass a positive T_Data_Tag_Group.con (t_status = ok) to the local user. If the confirmation is negative (n_status = not_ok), the local Transport Layer shall pass a T_Data_Tag_Group.con (t_status = not_ok) to the local user indicating that the transmission of the associated N_Data_Group.req did not succeed.

T_Data_Tag_Group.req(ack_request, hop_count_type, octet_count, priority, Destination Address, frame_format, tsdu)

ack_request	Data Link Layer Ack requested or don’t care
hop_count_type:	hop count 7 or Network Layer Parameter
octet_count:	length information as described in Data Link Layer
priority:	system, urgent, normal or low priority
destination_address:	destination address of the receiver
frame_format:	used frame format
tsdu:	this is the user data to be transferred by Transport Layer

T_Data_Tag_Group.con(hop_count_type, octet_count, priority, source_address, Destination Address, frame_format, tsdu, t_status)

hop_count_type:		hop count 7 or Network Layer Parameter
octet_count:	length information as described in Data Link Layer	
priority:	system, urgent, normal or low priority	
source_address:		the Individual Address of the originator of the message
destination_address:		destination address of the receiver
frame_format:		used frame format.
tsdu:		this is the user data that has been transferred by Transport Layer
t_status:	ok:	T_Data_Tag_Group sent successfully with N_Data_Group service
	not_ok:	transmission of the associated N_Data_Group request frame didn’t succeed

T_Data_Tag_Group.ind(hop_count_type, octet_count, priority, source_address, Destination Address, frame_format, tsdu)

hop_count_type:	hop count equals 7 or not
octet_count:	length information as described in Data Link Layer
priority:	system, urgent, normal or low priority
source_address:	the Individual Address of the originator of the message
destination_address:	destination address of the receive
frame_format:	used frame format
tsdu:	this is the user data that has been transferred by Transport Layer

7.4.4 T_Data_Broadcast service

The T_Data_Broadcast service shall be applied by the user of transport layer, to transmit a TSDU over a connection-less communication mode to all remote partners. The T_Data_Broadcast service shall neither be acknowledged nor confirmed by any remote transport layer entity. The confirmation shall be a local confirmation that shall be caused by the N_Data_Broadcast.con of the network layer.

The local user of transport layer shall prepare a TSDU for all the remote users of transport layer. The local user of transport layer shall apply the T_Data_Broadcast.req primitive to pass the TSDU to the local transport layer. The local transport layer shall accept the service request and shall pass it with a N_Data_Broadcast.req to the local network layer.

The local transport layer shall encode the TSDU to the NSDU and shall map the arguments to the corresponding arguments of the N_Data_Broadcast.req primitive.

The remote transport layer shall map a N_Data_Broadcast.ind primitive to a T_Data_Broadcast.ind primitive. The argument NSDU shall be mapped to the argument TSDU, the argument priority shall be mapped to the corresponding argument priority of the T_Data_Broadcast.ind primitive.

If the local transport layer receives an N_Data_Broadcast.con from the local network layer it shall pass a T_Data_Broadcast.con to the local transport layer user. If the confirmation is positive (n_status = ok), the local transport layer shall pass a positive T_Data_Broadcast.con (t_status = ok) to the local user. If the confirmation is negative (n_status = not_ok), the local transport layer shall pass a T_Data_Broadcast.con (t_status = not_ok) to the local user indicating that the transmission of the associated N_Data_Broadcast.req did not succeed.

T_Data_Broadcast.req(ack_request, hop_count_type, octet_count, priority, tsdu)

ack_request: Data link layer Ack requested or don't care

hop_count_type: this parameter shall be used to indicate whether the hop_count shall be set to 7 or if the network layer parameter shall be used

octet_count: length information as described in Data link layer

priority: highest, urgent, normal or low priority

tsdu: this is the user data to be transferred by transport layer

T_Data_Broadcast.con(ack_request, hop_count_type, octet_count, priority, tsdu, t_status)

ack_request: Data link layer Ack requested or don't care

hop_count_type: this parameter shall be used to indicate whether the hop_count shall be set to 7 or if the network layer parameter shall be used

octet_count: length information as described in Data link layer

priority: highest, urgent, normal or low priority

tsdu: this is the user data that has been transferred by transport layer

t_status: ok: T_Data_Broadcast sent successfully with N_Data_Broadcast service

not_ok: transmission of the associated N_Data_Broadcast request frame did not succeed

T_Data_Broadcast.ind(hop_count_type, octet_count, priority, source_address, tsdu)

hop_count_type: hop count equals 7 or not

octet_count: length information as described in Data link layer

priority: highest, urgent, normal or low priority

source_address: individual address of the device that requested the T_Data_Broadcast service

tsdu: this is the user data that has been transferred by transport layer

7.4.5 T_Data_SystemBroadcast service

The T_Data_SystemBroadcast service shall be applied by the user of transport layer, to transmit a TSDU over a connection-less communication mode to all remote partners. The T_Data_SystemBroadcast service shall neither be acknowledged nor confirmed by any remote transport layer entity. The confirmation shall be a local confirmation caused by the L_SystemBroadcast.con of the Layer-2.

The local user of transport layer prepares a TSDU for all the remote users of transport layer. The local user of transport layer shall apply the T_Data_SystemBroadcast.req primitive to pass the TSDU to the local transport layer. The local transport layer shall accept the service request and pass it with a N_Data_SystemBroadcast.req to the local Network Layer.

The local transport layer shall encode the TSDU to the NSDU and map the arguments to the corresponding arguments of the N_Data_SystemBroadcast.req primitive.

The remote transport layer shall map a N_Data_SystemBroadcast.ind primitive to a T_Data_SystemBroadcast.ind primitive. The argument NSDU shall be mapped to the argument TSDU, the argument priority shall be mapped to the corresponding argument priority of the T_Data_SystemBroadcast.ind primitive.

If the local transport layer receives a N_Data_SystemBroadcast.con from the local Network Layer, it shall pass a T_Data_SystemBroadcast.con to the local Transport layer user. If the confirmation is positive (n_status = ok), the local transport layer shall pass a positive T_Data_SystemBroadcast.con (t_status = ok) to the local user. If the confirmation is negative (n_status = not_ok), the local transport layer shall pass a T_Data_Broadcast.con (t_status = not_ok) to the local user indicating that the transmission of the associated N_Data_SystemBroadcast.req did not succeed.

T_Data_SystemBroadcast.req(ack_request, hop_count_type, octet_count, priority, tsdu)

ack_request: this parameter shall be used to indicate whether a layer-2 acknowledge is mandatory or optional

hop_count_type: this parameter shall be used to indicate whether the hop_count shall be set to 7 or if the network layer parameter shall be used

octet_count: this parameter shall be used to indicate the length information of the requested frame

priority: this parameter shall be used to indicate the priority that shall be used to transmit the requested frame; it shall be “system”, “urgent”, “normal” or “low”

tsdu: this parameter shall be used to contain the user data to be transferred by the transport layer

T_Data_SystemBroadcast.con(hop_count_type, octet_count, priority, tsdu, t_status)

hop_count_type: this parameter shall be used to indicate whether the hop_count of the transmitted frame has been set to 7 or if the network layer parameter has been used

octet_count: this parameter shall be used to indicate the length information of the transmitted frame

priority: this parameter shall be used to indicate the priority that has been used to transmit the requested frame; it shall be “system”, “urgent”, “normal” or “low”

tsdu: this parameter shall contain the user data that has been transferred by transport layer

t_status: ok: this value of this parameter shall be used to indicate that the transmission of the T_Data_SystemBroadcast.req has been successful

not_ok: this value of this parameter shall be used to indicate that the transmission of the T_Data_SystemBroadcast.req did not succeed

T_Data_SystemBroadcast.ind(hop_count_type, octet_count, priority, source_address, tsdu)

hop_count_type: this parameter shall be used to indicate whether the hop count of the received frame equals 7 or not

octet_count: this parameter shall be used to contain the length information of the received frame

priority: this parameter shall be used to indicate the priority of the received frame; it shall be “system”, “urgent”, “normal” or “low”

source_address: this parameter shall be used to indicate the source address of the received frame; it shall be the individual address of the device that has transmitted the T_Data_SystemBroadcast service

tsdu: this parameter shall contain the user data that has been transferred by transport layer

7.4.6 T_Data_Individual service

The T_Data_Individual service shall be applied by the user of transport layer, to transmit a TSDU over a connection-less point-to-point communication mode to exactly one remote partner. The T_Data_Individual service shall neither be acknowledged nor confirmed by the remote transport layer entity. The confirmation shall be a local confirmation that shall be caused by the N_Data_Individual.con of the network layer.

The local user of transport layer shall prepare a TSDU for the remote user. The local user of transport layer shall apply the T_Data_Individual.req primitive to pass the TSDU to the local transport layer. The destination shall be defined by the TSAP. The local transport layer shall map the TSAP to the destination individual address.

The local transport layer shall encode the TSDU to the NSDU and shall map the arguments to the corresponding arguments of the N_Data_Individual.req primitive.

The remote transport layer shall map a N_Data_Individual.ind primitive containing a T_DATA_INDIVIDUAL_REQ_PDU to a T_Data_Individual.ind primitive. The remote transport layer shall map the source_address to the TSAP. The argument NSDU shall be mapped to the argument TSDU, the other arguments shall be mapped to the corresponding arguments of the T_Data_Individual.ind primitive.

If the local transport layer receives a N_Data_Individual.con from the local network layer, it shall pass a T_Data_Individual.con to the local transport layer user. If the confirmation is positive (n_status = ok), the local transport layer shall pass a positive T_Data_Individual.con (t_status = ok) to the local user. If the confirmation is negative (n_status = not_ok), the local transport layer shall pass a T_Data_Individual.con (t_status = not_ok) to the local user indicating that the transmission of the associated N_Data_Individual.req did not succeed.

T_Data_Individual.req(ack_request, hop_count_type, octet_count, priority, TSAP, tsdu)

ack_request: Data link layer Ack requested or don't care

hop_count_type: this parameter shall be used to indicate whether the hop_count shall be set to 7 or if the network layer parameter shall be used

octet_count: length information as described in Data link layer

priority: highest, urgent, normal or low priority

TSAP: identifier of the service access point
(may be directly the individual address of the remote partner)

tsdu: this is the user data to be transferred by transport layer

T_Data_Individual.con(ack_request, hop_count_type, octet_count, priority, TSAP, tsdu, t_status)

ack_request: Data link layer Ack requested or don't care

hop_count_type: this parameter shall be used to indicate whether the hop_count shall be set to 7 or if the network layer parameter shall be used

octet_count: length information as described in Data link layer

priority: highest, urgent, normal or low priority

TSAP: identifier of the service access point
(may be directly the individual address of the remote partner)

tsdu: this is the user data that has been transferred by transport layer

t_status: ok: T_Data_Individual sent successfully with N_Data_Individual service

not_ok: transmission of the associated N_Data_Individual request frame did not succeed

T_Data_Individual.ind(hop_count_type, octet_count, priority, TSAP, tsdu)

hop_count_type: hop count equals 7 or not

octet_count: length information as described in Data link layer

priority: highest, urgent, normal or low priority

TSAP: identifier of the service access point
(may be directly the individual address of the remote partner)

tsdu: this is the user data that has been transferred by transport layer

7.4.7 T_Connect service

NOTE The specification for this service as described below only handles the event handling and frame flow under normal conditions. The exact conditions and exceptions and error handling, are specified in 7.6.

The T_Connect service shall be applied by the user of transport layer, to establish a transport connection on a connection-oriented point-to-point communication mode. The T_Connect primitives shall be mapped to N_Data_Individual primitives and vice versa according to the transport layer state machine described in 7.6. If the available resources allow for building up a new connection, the local transport layer shall accept the service request and shall try to send the T_CONNECT_REQ_PDU to the remote transport layer with a N_Data_Individual.req. The destination_address shall be an individual address, the priority shall be 'system'; the ack_request shall be set to true; the octet_count shall be set to 6; the hop_count_type shall be set to 'network layer parameter'.

If the remote transport layer receives a N_Data_Individual.ind primitive containing a T_CONNECT_REQ_PDU and allows for building up a new connection, it shall map the source_address to the TSAP. The other parameters of the N_Data_Individual.ind primitive shall be mapped to the corresponding parameters of the T_Connect.ind primitive. In this case the remote transport layer entity shall neither acknowledge nor confirm the T_Connect service to the local transport layer. The confirmation shall be a local confirmation that shall be caused by the N_Data_Individual.con of network layer.

If the remote transport layer receives a N_Data_Individual.ind primitive containing a T_CONNECT_REQ_PDU and does not allow for building up a new connection, the frame shall not be passed to the remote transport layer user. Instead, the remote transport layer shall send a T_DISCONNECT_REQ_PDU using a N_Data_Individual.req primitive to the local transport layer.

If the local transport layer receives a N_Data_Individual.con from the local network layer, it shall pass a T_Connect.con to the local transport layer user. If the confirmation is positive (n_status = ok), the local transport layer shall pass a positive T_Connect.con (t_status = ok) to the local transport layer user. If the confirmation is negative, (n_status = not_ok) the local transport layer shall pass a T_Disconnect.ind to the local transport layer user indicating that the transmission of the N_Data_Individual.req did not succeed.

T_Connect.req(destination_address, priority)

destination_address: individual address of the device to which the transport connection shall be established

priority: highest, urgent, normal or low priority

T_Connect.con(destination_address, TSAP, t_status)

destination_address: individual address of the device to which the transport connection was requested

TSAP: identifier of the service access point
(may be directly the individual address of the remote partner)

t_status: ok: T_Connect.req sent successfully with N_Data_Individual service

not_ok: transmission of the associated N_Data_Individual request frame did not succeed

T_Connect.ind(TSAP)

TSAP: identifier of the service access point
(may be directly the individual address of the remote partner)

7.4.8 T_Disconnect service

NOTE The specification for this service as described below only handles the event handling and frame flow under normal conditions. For exact conditions and exception and error handling, please refer to 7.6.

The T_Disconnect service shall be applied by the user of transport layer, to release a transport connection on a connection-oriented point-to-point communication mode. The T_Disconnect primitives shall be mapped to N_Data_Individual primitives and vice versa according to the transport layer state machine described in 7.6. The local transport layer shall accept the service request and shall try to send the T_DISCONNECT_REQ-PDU to the remote transport layer with an N_Data_Individual.req. The TSAP shall be mapped to the destination address; the priority shall be set to system; the ack_request shall be set to true; the octet_count shall be set to 6; the hop_count_type shall be set to network layer Parameter.

The remote transport layer shall map a N_Data_Individual.ind primitive containing a T_DISCONNECT_REQ-PDU to a T_Disconnect.ind. The argument source address shall be mapped to the argument TSAP. The T_Disconnect service shall neither be acknowledged nor confirmed by the remote transport layer entity. The confirmation to the local transport layer entity shall be caused by the N_Data_Individual.con of the local network layer.

If the local transport layer receives a N_Data_Individual.con from the local network layer, it shall pass a T_Disconnect.con primitive to the local transport layer user. If the confirmation is positive (n_status = ok), the local transport layer shall pass a positive T_Disconnect.con (t_status = ok) to the local transport layer user. If the confirmation is negative, (n_status = not_ok) the local transport layer shall pass a negative T_Disconnect.con (t_status = not_ok) to the local transport layer user indicating that the transmission of the N_Data_Individual.req did not succeed.

The T_Disconnect.ind primitive may also be caused by the transport layer entity in order to indicate a protocol error.

T_Disconnect.req(priority, TSAP)

priority: highest, urgent, normal or low priority

TSAP: identifier of the service access point to which the connection shall be released

T_Disconnect.con(priority, TSAP, t_status)

priority: highest, urgent, normal or low priority

TSAP: identifier of the service access point to which the release of the connection was requested

t_status: ok: T_Disconnect.req sent successfully with N_Data_Individual service

not_ok: The transmission of the associated N_Data_Individual-frame did not succeed

T_Disconnect.ind(TSAP)

TSAP: identifier of the service access point to which the connection is released

7.4.9 T_Data_Connected service

NOTE The specification for this service as described below only handles the event handling and frame flow under normal conditions. For exact conditions and exception and error handling, please refer to 7.6.

The T_Data_Connected service shall be applied by the user of transport layer, to transmit a TSDU over a transport connection on a connection-oriented communication mode to a remote partner. The T_Data_Connected service shall be acknowledged with a T_ACK_PDU by the remote transport layer entity. The T_Data_Connected primitives shall be mapped to N_Data_Individual primitives and vice versa according to the transport layer state machine described in 7.6.

The local user of transport layer shall prepare a TSDU for the remote user. The local user of transport layer shall apply the T_Data_Connected.req primitive to pass the TSDU to the local transport layer. The local transport layer shall accept the service request only if the connection is established (state OPEN_IDLE). The local transport layer shall then map the TSAP to the destination individual address and shall try to send the TSDU to the remote transport layer with a N_Data_Individual.req; the hop_count_type shall be set to network layer parameter. The local transport layer shall either pass a T_Data_Connected.con primitive to the local user that shall indicate either a correct data transfer or it shall pass a T_Disconnect.ind primitive to the local user that shall indicate an erroneous data transfer.

The remote transport layer shall only accept the N_Data_Individual.ind with the T_DATA_CONNECTED_REQ_PDU received, if the connection is established, i.e. in the states OPEN_IDLE, OPEN_WAIT. Therefore mutual T_Data_Connected services shall be allowed on a connection-oriented communication mode. If the N_Data_Individual.ind is accepted, the remote transport layer shall map the source individual address to the TSAP and shall pass the T_DATA_CONNECTED_REQ_PDU to the remote transport layer user. The remote transport layer shall confirm the reception by sending a N_Data_Individual.req containing a T_ACK-PDU to the local transport layer.

If the local transport layer receives a confirmation from the remote transport layer, it shall pass a confirmation to the local transport layer user. If the acknowledgement is a positive acknowledgement (T_ACK_PDU), the local transport layer shall pass a T_Data_Connected.con to the local user. If no acknowledgement is received or if the acknowledgement is a negative acknowledgement, the local transport layer shall repeat the transmission of the T_DATA_CONNECTED_REQ_PDU up to 3 times with an acknowledgement time-out time of 3 s. If it fails, the local transport layer shall pass a T_Disconnect.ind primitive to the local user indicating that the connection is released (state = CLOSED).

T_Data_Connected.req (octet_count, priority, TSAP, tsdu)

octet_count: length information as described in Data link layer

priority: highest, urgent, normal or low priority

TSAP: identifier of the service access point to which the frame shall be sent

tsdu: this is the user data to be transferred by transport layer

T_Data_Connected.con(TSAP)
transmission successful

TSAP: this shall be the identifier of the service access point to which the frame has been sent

T_Data_Connected.ind(octet_count, priority, TSAP, tsdu)

octet_count: length information as described in Data link layer

priority: highest, urgent, normal or low priority

TSAP: this shall be the identifier of the service access point from which the frame is received

tsdu: this shall be the user data that has been transferred by transport layer

7.5 Parameters of transport layer

The connection number list shall be the only parameter of transport layer. The implementation of this parameter, for example table or algorithm is up to the manufacturer. The latter may be achieved by mapping the address of the communication partner (group address or individual address with address_type flag) to a 13 bit connection_number.

Connection number list: address table
maps connection numbers to destination addresses and vice versa.

Connection timeout: time interval of 6 s;
timeout to breakdown a connection

Acknowledgement timeout: time interval of 3 s;
timer to start a repetition if no acknowledgement was received

Max_rep_count 3;
maximum of T_Connect.req repetitions

7.6 State machine of connection-oriented communication mode

7.6.1 General

This state machine is designed for only one connection at a time. To use more than one connection at a time, for each connection a separate state machine shall be used.

The transport layer state machine shall process the services T_Connect, T_Disconnect and T_Data_Connected. Other transport layer services shall directly be mapped as described for each individual service independent from the actual state of the transport layer state machine. Invalid PDUs shall be ignored.

The transport layer shall have the following local variables.

- connection_address shall be used to store the actual individual address of the partner (for a given TSAP).
- SeqNoSend shall be a 4 bit binary value, that shall be used to handle the sequence number of numbered data package for sent frames.
- SeqNoRcv shall be a 4 bit binary value, that shall be used to handle the sequence number of numbered data package for received frames.
- connection_timeout_timer this shall be a timer that shall use a time interval of 6 s; this timer shall start with the transition CLOSED→CONNECTING or OPEN_IDLE; it shall stop with the transition OPEN_IDLE→CLOSED; it shall restart if a N_Data_Individual.req is applied in the state machine or a correct N_Data_Individual.ind is received.
- acknowledgment_timeout_timer this shall be a timer that shall use a time interval of 3 s; this timer shall start with the transition OPEN_IDLE→OPEN_WAIT; it shall stop if a correct T_ACK-PDU is received or with transition into →CLOSED.
- rep_count shall be used to count the number of T_DATA_CONNECTED_REQ repetitions.

The user of transport layer shall always get a confirmation for a request.

The transport layer user cannot request a second service primitive on the same connection before getting the confirmation to the preceding primitive; parallel services shall thus not be allowed.

7.6.2 States

The state machine shall have the following states.

- CLOSED There is no connection.
- OPEN_IDLE There is a connection open.
- OPEN_WAIT The state machine is waiting for a T_ACK when data have been sent to the remote partner.
- CONNECTING Client only. The state is waiting for an Immediate Acknowledge (IACK) after trying to connect to a remote partner.

7.6.3 Required actions

Table 2 specifies the actions of the connection oriented state machine.

Table 2 – Actions of the connection oriented state machine

Action name	**Specification of the action**
A0	– do nothing
A1	– the connection_address shall be set to source address of received message; – a T_CONNECT_ind shall be sent to the user; – the SeqNoSend shall be set to 0; – the SeqNoRcv shall be set to 0; – the connection timeout timer shall be started.
A2	– a N_Data_Individual.req with T_ACK_PDU, priority = SYSTEM, destination = connection_address, sequence = SeqNoRcv shall be sent to the network layer (remote device); – the SeqNoRcv shall be incremented; – the received buffer shall be sent as a T_Data_Connected.ind to the user; – the connection timeout timer shall be restarted.
A3	– a N_Data_Individual.req with T_ACK_PDU, priority = SYSTEM, destination = connection_address, sequence = sequence of received message shall be sent to the network layer (remote device); – the connection timeout timer shall be restarted.
A4	– an N_Data_Individual.req with T_NACK_PDU, priority = SYSTEM, destination = connection_address, sequence = sequence of received message shall be sent to the network layer (remote device); – the connection timeout timer shall be restarted.
A5	– a T_Disconnect.ind shall be sent to the user; – the acknowledge timeout timer shall be stopped; – the connection timeout timer shall be stopped.
A6	– a N_Data_Individual.req with *T_DISCONNECT_REQ_PDU, priority = SYSTEM, destination = connection_address, sequence = 0* shall be sent to the network layer (remote device); – a T_Disconnect.ind shall be sent to the user; – the acknowledge timeout timer shall be stopped; – the connection timeout timer shall be stopped.
A7	– the received T_Data_Connected.req shall be stored and shall be sent as a N_Data_Individual.req with *T_DATA_CONNECTED_REQ_PDU, destination = connection_address, sequence = SeqNoSend* to the network layer (remote device); – the rep_count shall be set to 0; – the acknowledge timeout timer shall be started; – the connection timeout timer shall be started.

Table 2 (*continued*)

Action name	Specification of the action
A8	– the acknowledge timeout timer shall be stopped; – the SeqNoSend shall be incremented; – the stored buffer shall be sent to the user as a T_Data_Connected.con with *cleared error-bits, connection number = 0;* – the connection timeout timer shall be restarted.
A8b	– the acknowledge timeout timer shall be stopped; – the SeqNoSend shall be incremented; – the connection timeout timer shall be restarted.
A9	– the stored message shall be sent as a N_Data_Individual.req to the network layer (remote device); – the rep_count shall be incremented; – the acknowledge timeout timer shall be started; – the connection timeout timer shall be restarted.
A10	– a N_Data_Individual.req shall be sent back to the sender with *T_DISCONNECT_REQ_PDU Priority = SYSTEM, Destination = source (rbuffer), Sequence = 0.*
A11	– the event shall be stored back and shall be handled after the next event has been handled; – the order of T_Data_Connected.req events shall not be changed.
A12	– the connection_address shall be set to the address of the T_CONNECT_req; – a N_Data_Individual.req shall be sent with T_CONNECT_REQ_PDU; – the SeqNoSend shall be set to 0; – the SeqNoRcv shall be set to 0; – the connection timeout timer shall be started.
A13	– a T_Connect.con shall be sent to the user.
A14	– a N_Data_Individual.req shall be sent to the network layer with a *T_DISCONNECT_REQ_PDU, priority = SYSTEM, destination = connection_address, sequence = 0;* – a T_Disconnect.con shall be sent to the user; – the acknowledge timeout timer shall be stopped; – the connection timeout timer shall be stopped.
A14b	– a N_Data_Individual.req shall be sent to the network layer with a *T_DISCONNECT_REQ_PDU, priority = SYSTEM, destination = connection_address, sequence = 0;* – the acknowledge timeout timer shall be stopped; – the connection timeout timer shall be stopped.
A15	– a T_Disconnect.con shall be sent to the management user; – the acknowledge timeout timer shall be stopped; – the connection timeout timer shall be stopped.

7.6.4 Transition table of the connection oriented transport layer state machine

7.6.4.1 Style 1

Table 3 specifies for each combination of every possible event and every possible actual state the new state that the Style 1 state machine shall go to and the action that shall be done.

As an alternative, Table 4 specifies for each combination of every possible event and every possible actual state the new state that the Style 1-rationalized state machine shall go to and the action that shall be done.

Table 3 – Transition table – Style 1

Event		Actual state		
		CLOSED	**OPEN_IDLE**	**OPEN_WAIT**
00	N_DATA_INDIVIDUAL_ind, T_CONNECT_REQ_PDU (source_address == connection_address)	OPEN_IDLE A1	CLOSED A6	CLOSED A6
		CLOSED A10	OPEN_IDLE A0	OPEN_WAIT A0
01	N_DATA_INDIVIDUAL_ind, T_CONNECT_REQ_PDU (source_address != connection_address)	OPEN_IDLE A1	OPEN_IDLE A10	OPEN_WAIT A10
		CLOSED A10	OPEN_IDLE A0	OPEN_WAIT A0
02	N_DATA_INDIVIDUAL_ind, T_DISCONNECT_REQ_PDU (source_address == connection_address)	CLOSED A0	CLOSED A5	CLOSED A5
03	N_DATA_INDIVIDUAL_ind, T_DISCONNECT_REQ_PDU (source_address != connection_address)	CLOSED A0	OPEN_IDLE A0	OPEN_WAIT A0
04	N_DATA_INDIVIDUAL_ind, T_DATA_CONNECTED_REQ_PDU (source_address == connection_address) and (SeqNo_of_PDU == SeqNoRcv)	CLOSED A10	OPEN_IDLE A2	OPEN_WAIT A2
05	N_DATA_INDIVIDUAL_ind, T_DATA_CONNECTED_REQ_PDU (source_address == connection_address) and (SeqNo_of_PDU == ((SeqNoRcv –1)&0xF))	CLOSED A10	OPEN_IDLE A3	OPEN_WAIT A3
06	N_DATA_INDIVIDUAL_ind, T_DATA_CONNECTED_REQ_PDU (source_address == connection_address) and (SeqNo_of_PDU != SeqNoRcv) and (SeqNo_of_PDU !=((SeqNoRcv-1)&0xF))	CLOSED A10	OPEN_IDLE A4	OPEN_WAIT A4
07	N_DATA_INDIVIDUAL_ind, T_DATA_CONNECTED_REQ_PDU (source_address != connection_address)	CLOSED A10	OPEN_IDLE A10	OPEN_WAIT A10
08	N_DATA_INDIVIDUAL_ind, T_ACK_PDU (source_address == connection_address) and (SeqNo_of_PDU == SeqNoSend)	CLOSED A10	CLOSED A6	OPEN_IDLE A8
09	N_DATA_INDIVIDUAL_ind, T_ACK_PDU (source_address == connection_address) and (SeqNo_of_PDU != SeqNoSend))	CLOSED A10	CLOSED A6	CLOSED A6
10	N_DATA_INDIVIDUAL_ind, T_ACK_PDU (source_address != connection_address)	CLOSED A10	OPEN_IDLE A10	OPEN_WAIT A10

Table 3 (*continued*)

Event		Actual state		
		CLOSED	**OPEN_IDLE**	**OPEN_WAIT**
11	N_DATA_INDIVIDUAL_ind, T_NACK_PDU (source_address == connection_address) and (SeqNo_of_PDU != SeqNoSend)	CLOSED A10	CLOSED A6	CLOSED A6
12	N_DATA_INDIVIDUAL_ind, T_NACK_PDU (source_address == connection_address)and (SeqNo_of_PDU == SeqNoSend) and (rep_count < max_rep_count)	CLOSED A10	CLOSED A6	OPEN_WAIT A9
13	N_DATA_INDIVIDUAL_ind, T_NACK_PDU (source_address == connection_address) and (SeqNo_of_PDU == SeqNoSend) and (rep_count >= max_rep_count)	CLOSED A10	CLOSED A6	CLOSED A6
14	N_DATA_INDIVIDUAL_ind, T_NACK_PDU (source_address != connection_address)	CLOSED A10	OPEN_IDLE A10	OPEN_WAIT A10
15	T_DATA_CONNECTED_req	CLOSED A5	OPEN_WAIT A7	CLOSED A6
16	CONNECTION_TIME_OUT_ind	CLOSED A0	CLOSED A6	CLOSED A6
17	ACKNOWLEDGE_TIME_OUT_ind (rep_count < max_rep_count)	CLOSED A0	OPEN_IDLE A0	OPEN_WAIT A9
18	ACKNOWLEDGE_TIME_OUT_ind (rep_count >= max_rep_count)	CLOSED A0	OPEN_IDLE A0	CLOSED A6
19	N_DATA_INDIVIDUAL_con T_CONNECT_REQ_PDU IAK = OK (CLIENT ONLY)	CLOSED A0	OPEN_IDLE A13	OPEN_WAIT A13
20	N_DATA_INDIVIDUAL_con T_CONNECT_REQ_PDU IAK = NOT OK (CLIENT ONLY)	CLOSED A0	CLOSED A5	CLOSED A5
21	N_DATA_INDIVIDUAL_con T_DISCONNECT_REQ_PDU	CLOSED A0	OPEN_IDLE A0	OPEN_WAIT A0
22	N_DATA_INDIVIDUAL_con T_DATA_CONNECTED_REQ_PDU	CLOSED A0	OPEN_IDLE A0	OPEN_WAIT A0
23	N_DATA_INDIVIDUAL_con T_ACK_PDU	CLOSED A0	OPEN_IDLE A0	OPEN_WAIT A0
24	N_DATA_INDIVIDUAL_con T_NACK_PDU	CLOSED A0	OPEN_IDLE A0	OPEN_WAIT A0
25	T_CONNECT_req (CLIENT ONLY)	OPEN_IDLE A12	CLOSED A6	CLOSED A6
26	T_DISCONNECT_req (CLIENT ONLY)	CLOSED A15	CLOSED A14	CLOSED A14
27	All other, here not mentioned, messages EXAMPLE – not yet defined TPCI	CLOSED A0	OPEN_IDLE A0	OPEN_WAIT A0

□ Requirements for devices that accept a connection from the bus.

░ Requirements for devices that do not accept a connection from the bus.

▒ This event should for this actual state only occur in case of an internal error.

Table 4 – Transition table – Style 1 – rationalized

Event		Actual state		
		CLOSED	**OPEN_IDLE**	**OPEN_WAIT**
00	N_DATA_INDIVIDUAL_ind, T_CONNECT_REQ_PDU (source_address == connection_address)	OPEN_IDLE A1	CLOSED A6	CLOSED A6
		CLOSED A10	OPEN_IDLE A0	OPEN_WAIT A0
01	N_DATA_INDIVIDUAL_ind, T_CONNECT_REQ_PDU (source_address != connection_address)	OPEN_IDLE A1	OPEN_IDLE A10	OPEN_WAIT A10
		CLOSED A10	OPEN_IDLE A0	OPEN_WAIT A0
02	N_DATA_INDIVIDUAL_ind, T_DISCONNECT_REQ_PDU (source_address == connection_address)	CLOSED A0	CLOSED A5	CLOSED A5
03	N_DATA_INDIVIDUAL_ind, T_DISCONNECT_REQ_PDU (source_address != connection_address)	CLOSED A0	OPEN_IDLE A0	OPEN_WAIT A0
04	N_DATA_INDIVIDUAL_ind, T_DATA_CONNECTED_REQ_PDU (source_address == connection_address) and (SeqNo_of_PDU == SeqNoRcv)	CLOSED A10	OPEN_IDLE A2	OPEN_WAIT A2
05	N_DATA_INDIVIDUAL_ind, T_DATA_CONNECTED_REQ_PDU (source_address == connection_address) and (SeqNo_of_PDU == ((SeqNoRcv –1)&0xF))	CLOSED A10	OPEN_IDLE A3	OPEN_WAIT A3
06	N_DATA_INDIVIDUAL_ind, T_DATA_CONNECTED_REQ_PDU (source_address == connection_address) and (SeqNo_of_PDU != SeqNoRcv) and (SeqNo_of_PDU !=((SeqNoRcv-1)&0xF))	CLOSED A10	OPEN_IDLE A6	OPEN_WAIT A6
07	N_DATA_INDIVIDUAL_ind, T_DATA_CONNECTED_REQ_PDU (source_address != connection_address)	CLOSED A10	OPEN_IDLE A10	OPEN_WAIT A10
08	N_DATA_INDIVIDUAL_ind, T_ACK_PDU (source_address == connection_address) and (SeqNo_of_PDU == SeqNoSend)	CLOSED A10	CLOSED A6	OPEN_IDLE A8
09	N_DATA_INDIVIDUAL_ind, T_ACK_PDU (source_address == connection_address) and (SeqNo_of_PDU != SeqNoSend))	CLOSED A10	CLOSED A6	CLOSED A6
10	N_DATA_INDIVIDUAL_ind, T_ACK_PDU (source_address != connection_address)	CLOSED A10	OPEN_IDLE A10	OPEN_WAIT A10
11	N_DATA_INDIVIDUAL_ind, T_NACK_PDU (source_address == connection_address)	CLOSED A10	CLOSED A6	CLOSED A6

Table 4 (*continued*)

Event		Actual state		
		CLOSED	**OPEN_IDLE**	**OPEN_WAIT**
12	Not supported			
13	Not supported			
14	N_DATA_INDIVIDUAL_ind, T_NACK_PDU (source_address != connection_address)	CLOSED A10	OPEN_IDLE A10	OPEN_WAIT A10
15	T_DATA_CONNECTED_req	CLOSED A5	OPEN_WAI T A7	OPEN_WAIT A11
16	CONNECTION_TIME_OUT_ind	CLOSED A0	CLOSED A6	CLOSED A6
17	Not supported			
18	Not supported			
19	N_DATA_INDIVIDUAL_con T_CONNECT_REQ_PDU IAK = OK (CLIENT ONLY)	CLOSED A0	OPEN_IDLE A13	OPEN_WAIT A13
20	N_DATA_INDIVIDUAL_con T_CONNECT_REQ_PDU IAK = NOT OK (CLIENT ONLY)	CLOSED A0	CLOSED A5	CLOSED A5
21	N_DATA_INDIVIDUAL_con T_DISCONNECT_REQ_PDU	CLOSED A0	OPEN_IDLE A0	OPEN_WAIT A0
22	N_DATA_INDIVIDUAL_con T_DATA_CONNECTED_REQ_PDU	CLOSED A0	OPEN_IDLE A0	OPEN_WAIT A0
23	N_DATA_INDIVIDUAL_con T_ACK_PDU	CLOSED A0	OPEN_IDLE A0	OPEN_WAIT A0
24	Not supported			
25	T_CONNECT_req (CLIENT ONLY)	OPEN_IDLE A12	CLOSED A6	CLOSED A6
26	T_DISCONNECT_req (CLIENT ONLY)	CLOSED A15	CLOSED A14	CLOSED A14
27	All other, here not mentioned, messages EXAMPLE - not yet defined TPCI	CLOSED A0	OPEN_IDLE A0	OPEN_WAIT A0

□ Requirements for devices that accept a connection from the bus.

░ Requirements for devices that do not accept a connection from the bus.

▒ This event should for this actual state only occur in case of an internal error.

7.6.4.2 Style 2

Table 5 specifies for each combination of every possible event and every possible actual state the new state that the Style 2 state machine shall go to and the action that shall be done.

Table 5 – Transition table – Style 2

Event		Actual state		
		CLOSED	**OPEN_IDLE**	**OPEN_WAIT**
00	N_DATA_INDIVIDUAL_ind,T_CONNECT_REQ_PDU (source_address == connection_address)	OPEN_IDLE A1	OPEN_IDLE A0	OPEN_IDLE A0
01	N_DATA_INDIVIDUAL_ind, T_CONNECT_REQ_PDU (source_address != connection_address)	OPEN_IDLE A1	OPEN_IDLE A0	OPEN_WAIT A0
02	N_DATA_INDIVIDUAL_ind, T_DISCONNECT_REQ_PDU (source_address == connection_address)	CLOSED A0	CLOSED A5	CLOSED A5
03	N_DATA_INDIVIDUAL_ind, T_DISCONNECT_REQ_PDU (source_address != connection_address)	CLOSED A0	OPEN_IDLE A0	OPEN_WAIT A0
04	N_DATA_INDIVIDUAL_ind, T_DATA_CONNECTED_REQ_PDU (source_address == connection_address) and (SeqNo_of_PDU == SeqNoRcv)	CLOSED A0	OPEN_IDLE A2	OPEN_WAIT A2
05	N_DATA_INDIVIDUAL_ind, T_DATA_CONNECTED_REQ_PDU (source_address == connection_address) and (SeqNo_of_PDU == ((SeqNoRcv -1)0xF))	CLOSED A0	OPEN_IDLE A3	OPEN_WAIT A3
06	N_DATA_INDIVIDUAL_ind, T_DATA_CONNECTED_REQ_PDU (source_address == connection_address) and (SeqNo_of_PDU != SeqNoRcv) and (SeqNo_of_PDU != ((SeqNoRcv -1)0xF))	CLOSED A0	OPEN_IDLE A4	OPEN_WAIT A4
07	N_DATA_INDIVIDUAL_ind, T_DATA_CONNECTED_REQ_PDU (source_address != connection_address)	CLOSED A0	OPEN_IDLE A0	OPEN_WAIT A0
08	N_DATA_INDIVIDUAL_ind, T_ACK_PDU (source_address == connection_address) and (SeqNo_of_PDU == SeqNoSend)	CLOSED A0	CLOSED A6	OPEN_IDLE A8b
09	N_DATA_INDIVIDUAL_ind, T_ACK_PDU (source_address == connection_address) and (SeqNo_of_PDU != SeqNoSend))	CLOSED A0	CLOSED A6	OPEN_WAIT A0
10	N_DATA_INDIVIDUAL_ind, T_ACK_PDU (source_address != connection_address)	CLOSED A0	OPEN_IDLE A0	OPEN_WAIT A0
11	N_DATA_INDIVIDUAL_ind, T_NACK_PDU (source_address == connection_address) and (SeqNo_of_PDU != SeqNoSend)	CLOSED A0	CLOSED A6	OPEN_WAIT A0
12	N_DATA_INDIVIDUAL_ind, T_NACK_PDU (source_address == connection_address)and (SeqNo_of_PDU == SeqNoSend) and (rep_count < max_rep_count)	CLOSED A0	OPEN_IDLE A0	OPEN_WAIT A9
13	N_DATA_INDIVIDUAL_ind, T_NACK_PDU (source_address == connection_address) and (SeqNo_of_PDU == SeqNoSend) and (rep_count >= max_rep_count)	CLOSED A0	OPEN_IDLE A0	CLOSED A6