

Data communication for HVAC application — Automation net

Part 2: EIBnet

ICS 35.240.99; 97.120

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW

BSI
Documents



Document Sharing Hub

National foreword

This Draft for Development is the official English language version of ENV 13321-2:1999.

This publication is not to be regarded as a British Standard.

It is being issued in the Draft for Development series of publications and is of a provisional nature because further experience of its application is required before publication as a European Standard. It should be applied on this provisional basis, so that information and experience of its practical application may be obtained.

Comments arising from the use of this Draft for Development are requested so that the UK experience can be reported to the European organization responsible for its conversion into a European Standard. A review of this publication will be initiated 2 years after its publication by the European organization so that a decision can be taken on its status at the end of its three-year life. The commencement of the review period will be notified by an announcement in *Update Standards*.

According to the replies received by the end of the review period, the responsible BSI Committee will decide whether to support the conversion into a European Standard, to extend the life of the prestandard or to withdraw it. Comments should be sent in writing to the Secretary of BSI Subcommittee RHE/16/1, Management systems for buildings, at 389 Chiswick High Road, London W4 4AL, giving the document reference and clause number and proposing, where possible, an appropriate revision of the text.

Cross-references

The British Standards which implement international or European publications referred to in this document may be found in the BSI Standards Catalogue under the section entitled “International Standards Correspondence Index”, or by using the “Find” facility of the BSI Standards Electronic Catalogue.

Summary of pages

This document comprises a front cover, an inside front cover, the ENV title page, pages 2 to 115 and a back cover.

The BSI copyright notice displayed in this document indicates when the document was last issued.

This British Standard, having been prepared under the direction of the Engineering Sector Committee, was published under the authority of the Standards Committee and comes into effect on 15 May 2000

© BSI 05-2000

ISBN 0 580 34201 8

Amendments issued since publication

Amd. No.	Date	Comments

English version

**Data communication for HVAC application - Automation net -
Part 2: ElBnet**

This European Prestandard (ENV) was approved by CEN on 22 December 1999 as a prospective standard for provisional application.

The period of validity of this ENV is limited initially to three years. After two years the members of CEN will be requested to submit their comments, particularly on the question whether the ENV can be converted into a European Standard.

CEN members are required to announce the existence of this ENV in the same way as for an EN and to make the ENV available promptly at national level in an appropriate form. It is permissible to keep conflicting national standards in force (in parallel to the ENV) until the final decision about the possible conversion of the ENV into an EN is reached.

CEN members are the national standards bodies of Austria, Belgium, Czech Republic, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

Central Secretariat: rue de Stassart, 36 B-1050 Brussels

Table of Contents

1	Foreword	5
2	Introduction	6
3	Scope	7
4	References	7
5	General Requirements	7
6	Physical Layer / MAC Layer / Logical Link Layer (A)	8
7	Logical Link Layer (B)	10
7.1	Address Types	10
7.1.1	MAC-Address	10
7.1.2	System-Identifier	10
7.1.3	Physical Address	10
7.1.4	Group Address	11
7.1.5	Poll-Group Address	11
7.2	Frame Formats	11
7.2.1	Frame Format L-Data	12
7.2.2	Frame Format L-Poll-Data	13
7.2.3	Destination SAP / Source SAP	14
7.2.4	LLC-Control	14
7.2.5	EIB-Control	14
7.2.6	System-Identifier	14
7.2.7	Control Field	14
7.2.8	Source Address	15
7.2.9	Destination Address	15
7.2.10	LL-Length	15
7.3	PollData Control	16
7.4	Logical Link Layer (B) Services	17
7.4.1	L_Data Service	17
7.4.2	L_Poll_Data Service and Protocol	18
7.5	L_Busmon Service	19
7.6	L_Service_Information Service	19
7.7	Parameters of Layer-2	19
8	Network Layer	20
8.1	NPDU	20
8.2	Network Layer Services	20
8.2.1	N_Data Service	20
8.2.2	N_Groupdata Service	21
8.2.3	N_Broadcast Service	22
8.3	Parameters of Layer-3	23
8.4	State Machine of Layer-3	23
8.5	The Layer-3 of a Bridge	23
8.6	The Layer-3 of a Router	23

9	Transport Layer	26
9.1	Communication Relationships	26
9.1.1	One-to-many connection-less (Multicast) Communication Relationship	26
9.1.2	One-to-all connection-less (Broadcast) Communication Relationship	26
9.1.3	One-to-one connection-less Communication Relationship	27
9.1.4	One-to-one connection-oriented Communication Relationship	27
9.2	TPDU	27
9.3	Transport Layer Services	28
9.3.1	T_Groupdata Service	28
9.3.2	T_Broadcast Service	29
9.3.3	T_Data_Unack	30
9.3.4	T_Connect Service	31
9.3.5	T_Disconnect Service	31
9.3.6	T_Data Service	32
9.4	Parameters of Layer-4	33
9.5	State Machine of Layer-4	33
10	Session Layer	40
11	Presentation Layer	40
12	Application Layer	42
12.1	APDU	43
13	Application Layer Services	45
13.1	Layer-7 Services on Multicast Communication Relationships	45
13.1.1	A_Read_Group Service	45
13.1.2	A_Write_Group Service	47
13.2	Layer-7 Services on Broadcast Communication Relationships	48
13.2.1	A_Set_Physaddr Service	48
13.2.2	A_Read_Physaddr Service	49
13.2.3	A_Read_Physaddr_Serno Service	51
13.2.4	A_Set_Physaddr_Serno Service	52
13.2.5	A_Service_Information Service	53
13.2.6	A_Set_SystemID Service	54
13.2.7	A_Read_SystemID Service	55
13.2.8	A_Read_SystemID_Selective_Request Service	56
13.3	Layer-7 Services on one-to-one connection-less Communication Relationships	56
13.3.1	A_Read_Property_Value Service	57
13.3.2	A_Write_Property_Value Service	59
13.3.3	A_FileRead_Property Service	61
13.3.4	A_FileWrite_Property Service	64
13.3.5	A_Read_Property_Description Service	67
13.3.6	A_Uread_Memory Service (optional)	69
13.3.7	A_Uwrite_Memory Service (optional)	71
13.3.8	A_Uwrite_Memory_Bit Service (optional)	73
13.3.9	A_Uread_MfactInfo Service (optional)	75

13.4 Layer-7 Services on one-to-one connection-oriented Communication Relationships	76
13.4.1 A_Read_Adc Service	76
13.4.2 A_Read_Memory Service	78
13.4.3 A_Write_Memory Service	80
13.4.4 A_Read_Mask_Version Service	81
13.4.5 A_Restart Service	83
13.4.6 A_Write_Memory_Bit Service (optional)	84
13.4.7 A_Authorize Service	86
13.4.8 A_Setkey Service	87
13.5 Parameters of Layer-7	90
14 Network Management	91
14.1 Configuration Management Sequence	91
14.1.1 Setting the Physical Address and the System-ID	92
14.1.2 Access Protection	92
14.2 Device Model	93
14.2.1 Major Parts of an EIB Device	94
14.2.2 Load Procedure for Device Configuration Data	94
14.3 EIB-Objects	98
14.3.1 Data types of Properties	100
14.3.2 Property-Identifier	101
14.3.3 Standard EIB Objects and Properties	104

1 Foreword

This European Prestandard has been prepared by Technical Committee CEN/TC 247 "Controls for mechanical building services", the secretariat of which is held by SNV.

According to the CEN/CENELEC Internal Regulations, the national standards organizations of the following countries are bound to announce this European Prestandard: Austria, Belgium, Czech Republic, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and the United Kingdom.

This European Prestandard (ENV) is part of a series of standards for system-neutral data communications in HVAC systems. Together with prENV 13221-1 BACnet, Profibus and WorldFIP® this Prestandard covers the data communication on the Automation level.

The position of this standard in the whole range of standards for mechanical building services is illustrated in figure 1.

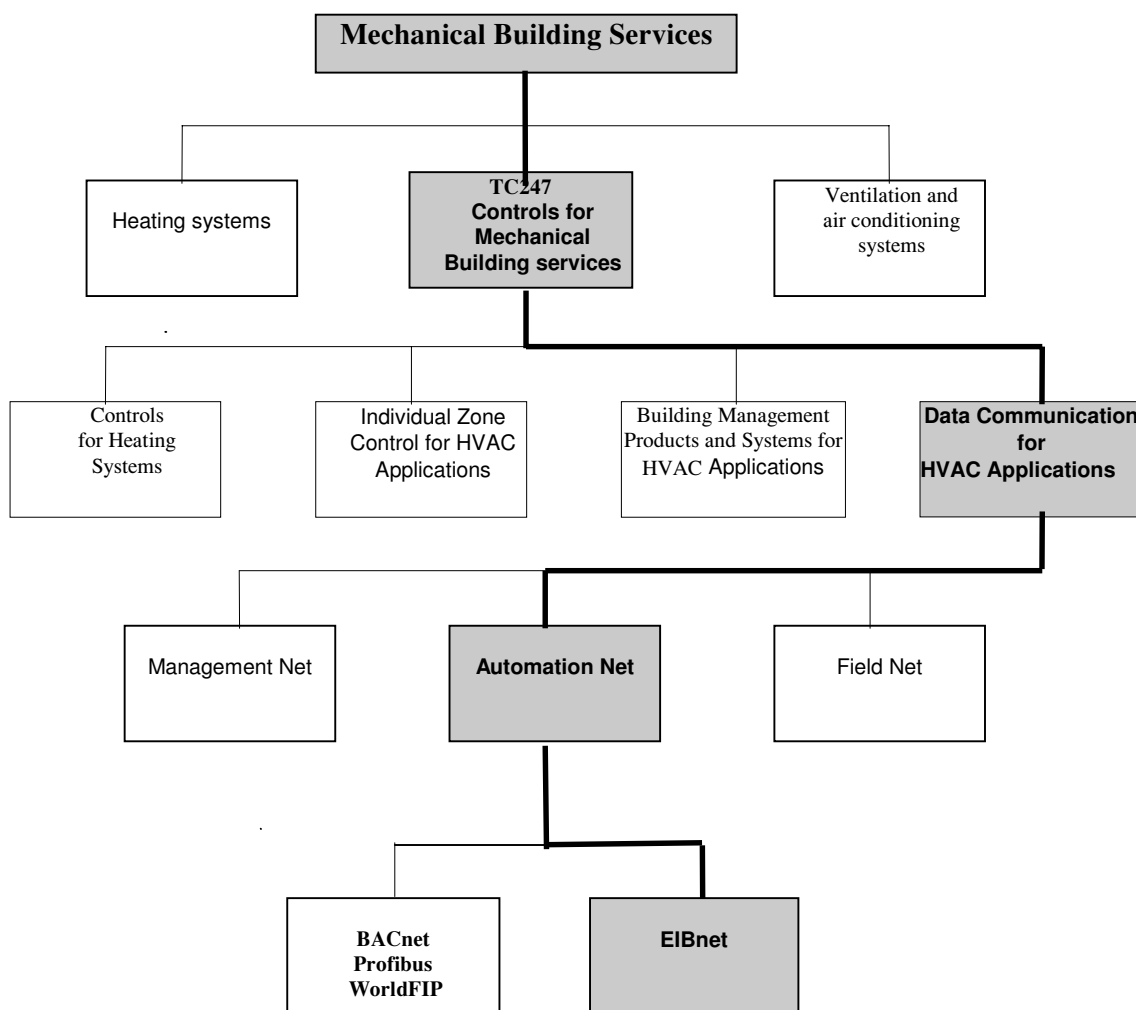


Figure 1 :Relationship to TC247

The shaded boxes in figure 1 indicate the contents and the hierarchy of this standard. The plain areas show the positioning of this standard in relationship to other relevant mechanical building services standards

2 Introduction

This specification of the EIB describe the usage of the EIB on the automation net level. The EIB on automation net level offers the possibility to use the EIB on faster Media. EIB devices for the automation net level use a homogeneous addressing method to the field net level. This allows to use the EIB with automation level devices like Application Specific Controller (ASC) and programmable controllers in home and light commercial building environment. EIB offers full compatibility for process communication and for monitoring, engineering and commanding of applications in ASC. Furthermore EIB on automation level is well suited for the interconnection of EIB-Networks. It supports the structured cabling approach.

The EIB system may be implemented on different media. Therefore this standard concerns only the media independent parts of the system called „protocol“.

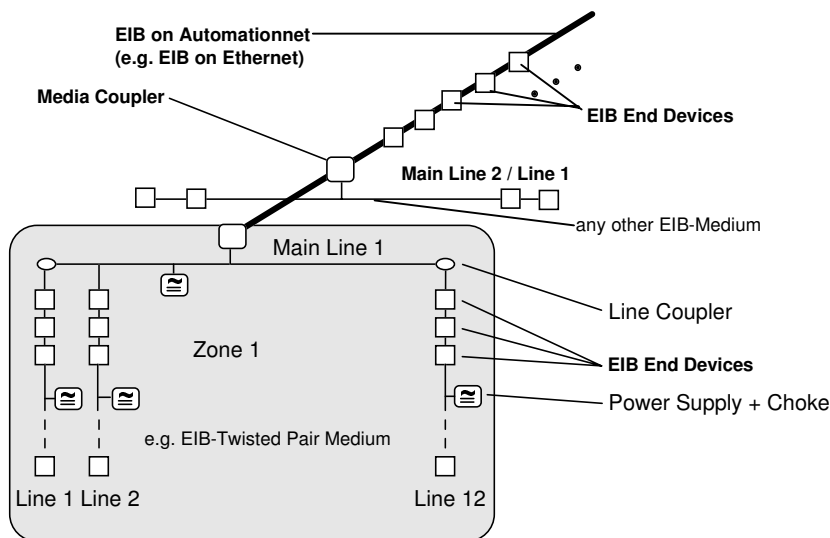


Figure 1: Relationship EIB to EIB on Automationnet

EIB	The EIB is a control system for all related applications in home and buildings.
Media Coupler	A Media Coupler connects a physical segment of EIB with a physical segment of an other EIB-Medium.
Device	A device is either a Coupler or an EIB end device. A device has a unique physical address.
EIB End Device	(EED) This is a device with a unique physical address, that performs an application in a heating, ventilating and air conditioning and related building management environment.
Line	A line may consist of up to 255 EIB end devices.
Line Coupler	A Line Coupler connects two lines.
EIB-System	An EIB-System is a number of EIB end devices using the same System-ID and being connected either to a line coupler or to a media coupler.
Network	A network or EIB network is a number of EIB-Systems.

3 Scope

This European Prestandard defines a system neutral data communication method for use at the automation level in heating, ventilating and air conditioning and related building management applications.

4 Normative references

This European Prestandard incorporates by reference, provisions from other publications. These normative references are cited at the appropriate places in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this European Prestandard only when incorporated in it by amendment or revision. For undated references the latest edition of the publication applies.

ISO/IEC 8802-2 Information technology - Telecommunications and information exchange between systems -Local and metropolitan area networks - Specific requirements

5 General Requirements

Each of the following clauses describes an OSI layer. Each clause contains sub-clauses giving a survey about the remainder of the clause. Interoperations of a layer with its adjacent layers are described there.

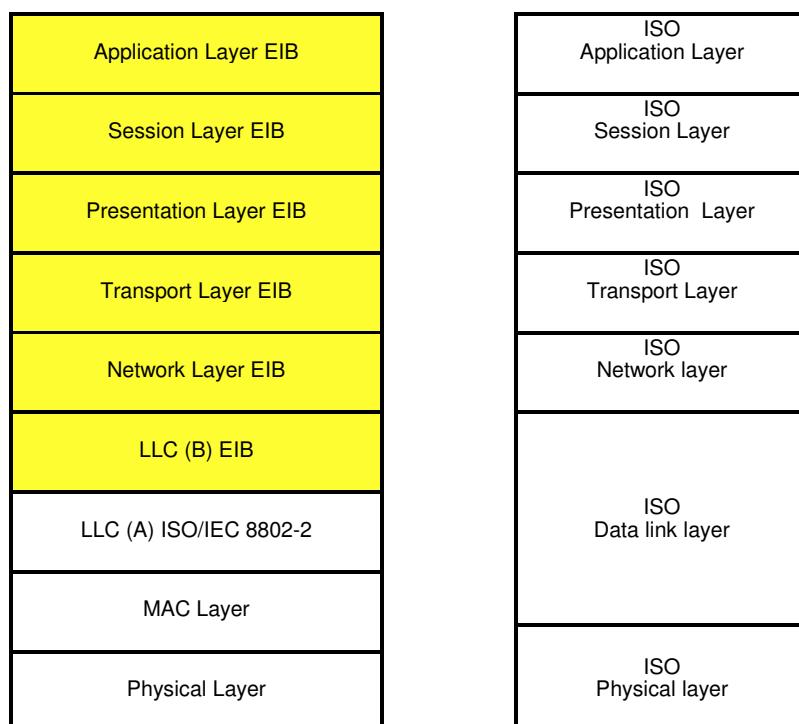


Figure 2: Relationship to LAN reference model

6 Physical Layer / MAC Layer / Logical Link Layer (A)

For the Physical Layer / MAC Layer all standards using the ISO/IEC 8802-2 Logical Link Layer can be used.

The International Standards for media access technologies using ISO/IEC 8802-2 as Logical Link Layer are as follows :

1. ISO/IEC 8802-3 / IEEE 802.3 a bus utilising CSMA/CD as the access method (Ethernet)
2. ISO/IEC 8802-4 / IEEE 802.4 a bus utilising token passing as the access method
3. ISO/IEC 8802-5 / IEEE 802.5 a ring utilising token passing as the access method
4. ISO/IEC 8802-6 / IEEE 802.6 a dual bus utilising distributed queuing as the access method
5. ISO/IEC 8802-7 / IEEE 802.7 a ring utilising slotted ring as the access method

Other types are under investigation.

THIS PAGE LEFT BLANK INTENTIONALLY

7 Logical Link Layer (B)

The EIB-Data Link Layer for automationnet level uses the Logical Link Layer from ISO/IEC 8802-2 with the DL-UNITDATA primitive only.

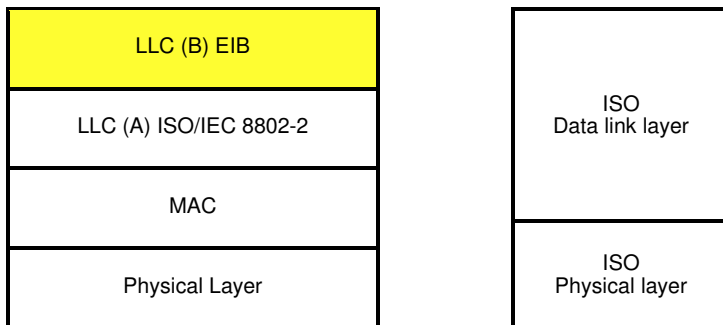


Figure 4: Relationship to LAN reference model

7.1 Address Types

7.1.1 MAC-Address

The MAC-Address is the physical device address of the used MAC-layer e.g. the Ethernet-Address. The MAC-address may be used as broadcast address.

7.1.2 System-Identifier

Only EIB-Devices with the same System-ID can communicate with each other. For communication between different systems there may exist EIB-Devices with more than one System-ID.

System-ID							
7	6	5	4	3	2	1	0

Figure 5: System-Identifier

7.1.3 Physical Address

Each device, i.e. a router or an EIB end device shall have a unique physical address in an EIB-System. The physical address is a two octet value that consists of an eight bit device address, a four bit line address and a four bit zone address.

Physical Address															
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
zone address				line address				device address							

Figure 6: Physical Address

The device address shall be unique within a line. Routers shall always have the device address zero: EIB end devices may have addresses 1-255.

The line address shall be unique within a zone (0-15). The devices in the main line of a zone shall always have the line address zero.

The zone address shall be unique within an EIB network (0-15). The devices at the backbone-line shall always have the zone address zero.

The physical addressing uses the DL-UNITDATA primitive with broadcast addressing and with a specified MAC-address over the EIB for automationnet level.

7.1.4 Group Address

Group Address															
Dest. Address (high)								Dest. Address (low)							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

Figure 7: Group Address

The group address is a two byte value that doesn't need to be unique. An EIB end device may have more than one group address.

Each EIB end device belongs to group zero, i.e. request frames with destination group address zero are broadcasts.

The group addressing uses the DL-UNITDATA primitive with broadcast addressing only over the EIB for automationnet level

7.1.5 Poll-Group Address

Group Address															
Dest. Address (high)								Dest. Address (low)							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

Figure 8: Poll-Group Address

The Poll-Group Address is a two byte value that doesn't need to be unique. An EIB end device may have more than one Poll-Group Address.

The poll group addressing uses the DL-UNITDATA primitive with broadcast addressing for a request frame and with a specified MAC-address for any response frame over EIB.

7.2 Frame Formats

There exists two types of frames, the L-Data frame and the L-PollData frame. The L-Data frame is used for normal communication and the L-PollData frame is only used for polling mechanism.

7.2.1 Frame Format L-Data

	Octet 0	Octet 1	Octet 2	Octet 3	Octet 4	
	Dest. SAP	Source SAP	LLC-Control	EIB-Control	System-ID	
	Dest. SAP (= "xx")	Source SAP (= "xx")	LLC-Control (= 03h)	Standard-EIB 0 0 0 0 0 0 0	System-ID	. . .
MAC-Header	L-2 a			L-2 b		

[illegible]

Figure 9: L_Data Frame Format

7.2.2 Frame Format L-Poll-Data

	Octet 0	Octet 1	Octet 2	Octet 3	Octet 4	
	Dest. SAP	Source SAP	LLC-Control	EIB-Control	System-ID	
	Dest. SAP (= "xx")	Source SAP (= "xx")	LLC-Control (= 03h)	Standard-EIB 0 0 0 0 0 0 0	System-ID	
MAC-Header	L-2 a			L-2 b		

..	Octet 5	Octet 6	Octet 7	Octet 8	Octet 9	Octet 10	Octet 11	Octet 12	Octet 13	...	Octet 26
:	Control Field	Source Device Addr.		Destination Addr.		Poll Control	Poll-Data 0	Poll-Data 1	Poll-Data 3	...	Poll-Data 15
..											
:											
.		zone address " " " "	line address " " " "	node address " " " " " "	dest. polling group address " " " " " " " "						
	L-2b										

Figure 10: L_PollData Frame Format

A frame is sent as a sequence of octets. Figure 10 shows the octets instead of the UART characters of the request frame to make it easier to read. The UART character that corresponds to octet 0 is sent first, the octet with the highest number is the last character being sent. The figure shows the octets with MSB first.

7.2.3 Destination SAP / Source SAP

The Destination and Source Service Access Points shall be a single octet value X'xx' and is used to indicate that the Link layer Service Data Unit (LSDU) containing a EIB Data Packet.

7.2.4 LLC-Control

The LLC-Control is always X'03' and indicates that only the DL-UNITDATA primitive from ISO/IEC 8802-2 is used.

7.2.5 EIB-Control

The EIB-Control is used to specify services that are only used on the EIB for automationnet. The EIB-Control Field is encoded as follows :

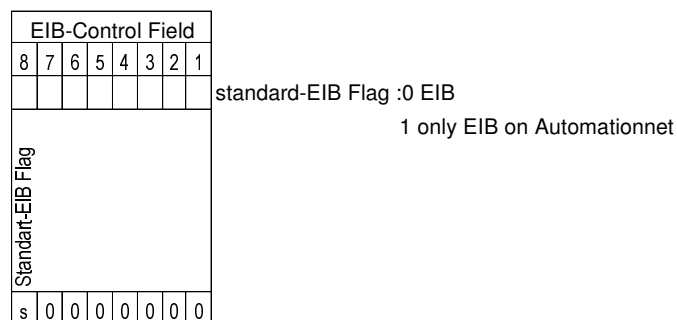


Figure 11: EIB-Control Field Format

7.2.6 System-Identifier

Only EIB-Devices with the same System-ID can communicate with each other. For communication between different systems there may exist EIB-Devices with more than one System-ID.

7.2.7 Control Field

The Control field is for compatibility to the EIB-Twisted Pair medium. The encoding of the control field is shown in the next figure:

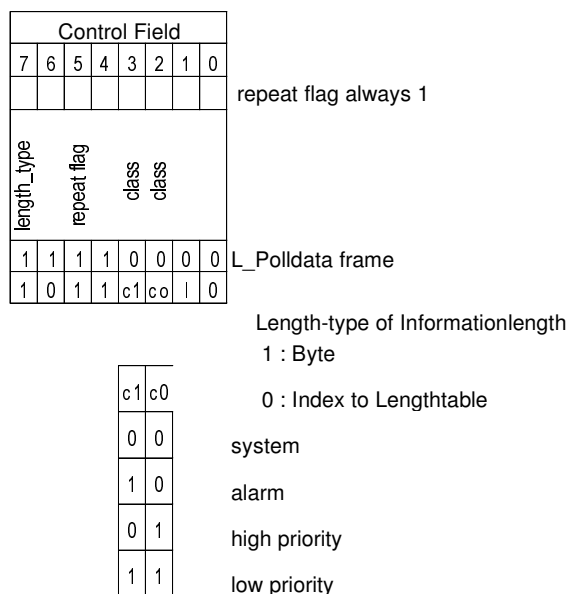


Figure 12: Control Field

The control field indicates the type of the L_Data frame. The two class bits of the control field control the priority of the frame.

7.2.8 Source Address

The Source Address is always the Physical Address of the sending device.

7.2.9 Destination Address

The Destination Address depends on the Destination Address Flag of the LL-Length field. It could be either a Physical Address or a Group Address.

7.2.10 LL-Length

The LL-Length field is encoding as follows:

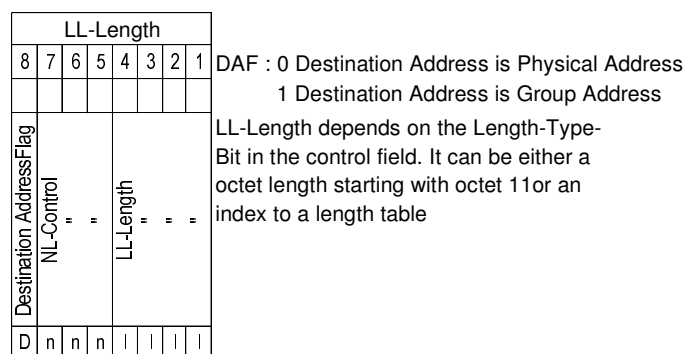


Figure 13: LL-Length Format

coding	length
0	16
1	17
2	18
3	19
4	20
5	22
6	25
7	29
8	34
9	40
10	80
11	120
12	160
13	200
14	244
15	ESC

Figure 14: LL-Length Table

The L_Data request frame format has a variable length, with length type 1 the maximum length is 28 characters. The LL-Length indicates the number of octets (1-15) of the LSDU starting with octet 11.

7.3 PollData Control

Poll Control							
8	7	6	5	4	3	2	1
Poll Control = = =				Slave = = =			
p	p	p	p				

Figure 15: Poll Control Format

Poll Control	Slave	
0000 b	Number of Slaves	Poll request for fieldlevel EIB (through a Media Coupler)
0100 b	Number of Slaves	Poll response from fieldlevel EIB
1000 b	Number of Slaves	Poll Data Master frame
1100 b	Slave ID	1100 b Poll Data Slave frame

7.4 Logical Link Layer (B) Services

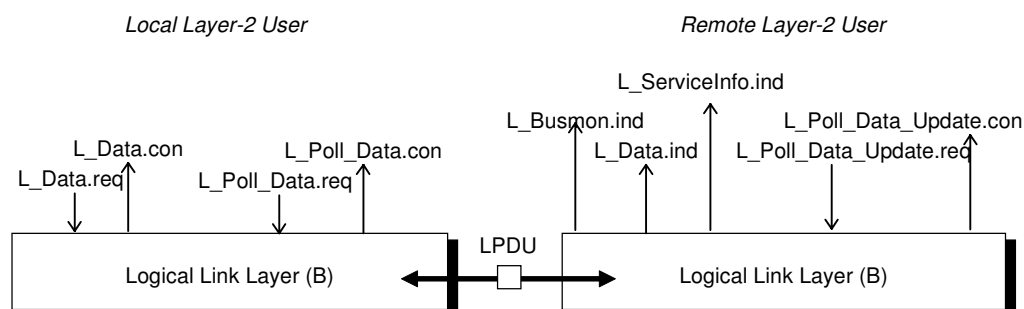


Figure 16: Interactivity of the Logical Link Layer (B)

7.4.1 L_Data Service

The local user of layer-2 prepares an LSDU for the remote user. The local user of layer-2 applies the L_Data.req primitive to pass the LSDU to the local layer-2. The local layer-2 accepts the service request and tries to send the LSDU to the remote layer-2. The destination address may be a physical address or a group address (multicast or broadcast). The local layer-2 passes an L_Data.con primitive to the local user that indicates the data transfer results.

If the request frame is received and the system-ID and destination address match, the remote layer-2 passes the LSDU with a L_Data.ind primitive to the remote user.

```
L_Data.req(MAC-Address, system-ID, source_address, destination_address, DAF,
class, l_sdu)
MAC-Address      destination Address of MAC-Layer
system-ID        Identifier of the EIB-System.
source_address:  Physical address of the EIB end device that
                  requested the L_Data service.
destination_address: Either a physical address or a group address
DAF:             Indicates whether destination_address is a
                  physical('0') or a group address ('1').
class:           System, alarm, high or low priority.
l_sdu:           This is the user data to be transferred by
                  layer-2.
```

```
L_Data.con(l_status)
l_status: OK;      Request frame sent successfully.
                 not_ok; Transmission of the request frame failed.
```

```
L_Data.ind(MAC-Address, source_address, destination_address, DAF, class,
l_sdu)
MAC-Address          source Address of MAC-Layer
source_address:      Physical address of the EIB end device that
                      requested the L_Data service.
destination_address: The physical address of this device or a
                      group address of this device.
DAF:                 Indicates whether destination_address is a
                      physical('0') or a group address ('1').
class:               System, alarm, high or low priority.
l_sdu:               This is the user data that has been
                      transferred by layer-2.
```

7.4.2 L_Poll_Data Service and Protocol

The L_Poll_Data service is a confirmed multicast service. The local user of layer-2 applies the L_Poll_Data.req primitive to request data from one or more remote users. The local layer-2 accepts the service request and tries to send the L_Poll_Data.req to the remote layer-2. The destination address is always a poll group address. The poll group address is a parameter of layer-2.

After receiving a correct L_Poll_Data request frame with a poll_group_address and System-ID matches, the remote layer-2 responds with a L_Poll_Data frame with a single Poll-Data octet to the address of the local layer-2. The remote layer-2 gets the Poll-Data octet from its user with the L_Poll_Update.req primitive. The Poll_Data character shall be transmitted with the correct slot-id. The response slot number is a parameter of layer-2.

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

The local layer-2 expects a number of L_PollData frames from the poll group. If an expected L_PollData frame has not been received after 3 seconds the local layer-2 fills the poll data with the fill octet FEh. The remote layer-2 can therefore still count L_PollData frames even if a member of the poll group doesn't respond.

The local layer-2 passes 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 EIB end devices on a single physical segment. The number of expected Poll_Data characters is limited to 16.

```
L_Poll_Data.req(MAC-Address, destination, no_of_expected_poll_data)
MAC-Address          destination Address of MAC-Layer
destination:         a poll group address
no_of_expected_poll_data: number of expected poll data

L_Poll_Data.con(l_status, poll_data_sequence)
l_status: OK;         valid poll_data_sequence
                  not_ok; invalid poll_data_sequence, i.e. collision
                      occurred during transmission of a FILL, or
                      at least one Poll_Data not correct
poll_data_sequence:  sequence of Poll_Data octets and FILL octets
```

`L_Poll_Update.req(Poll_Data)`
`Poll_Data:`

The value of the `Poll_Data` octet to 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.

7.5 L_Busmon Service

The `L_Busmon` service is a local data link service available only in data link bus monitor mode. It consists of the `L_Busmon.ind` primitive which transfers each received frame from the local layer-2 to the local layer-2 user.

`L_Busmon.ind(MAC-Address, l_status, time_stamp, lpdu)`
`MAC-Address` source Address of MAC-Layer
`l_status:` information whether a frame error, bit error or a parity error was detected in the received frame. Additional information about the number of already received frames may also be contained.
`time_stamp:` timing information, when the start bit of the frame was received
`lpdu:` all octets of the received frame

7.6 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()`. a frame was received which contained the physical address of the local layer-2 as source address.

7.7 Parameters of Layer-2

The following parameters influence the behaviour of layer-2 and are required inside layer-2 in order to operate correctly:

physical address	unique physical address of this device
address table	address table with the group address(es) of this EIB end device.
system-ID	defines the system in which a device can communicate
poll group address	the poll group address of this EIB end device
slot number	the response slot number of this EIB end device
data link layer mode	the normal or the bus monitor mode of the data link layer.
response slot number	the response slot number of this EIB end device

8 Network Layer

The network layer is controlling the number of routers and bridges that a frame with a group address as a destination is passing on its way from the source to the destination. The network layer is the lowest layer that deals with end-to-end transmission. In addition layer-3 offers more detailed services to the transport layer by encoding and decoding L_Data services of layer-2 which are mapped to N_Data, N_Broadcast and N_Group_Data services.

The Network link layer is using the services of the data link layer and provides services to the transport layer (Figure 17).

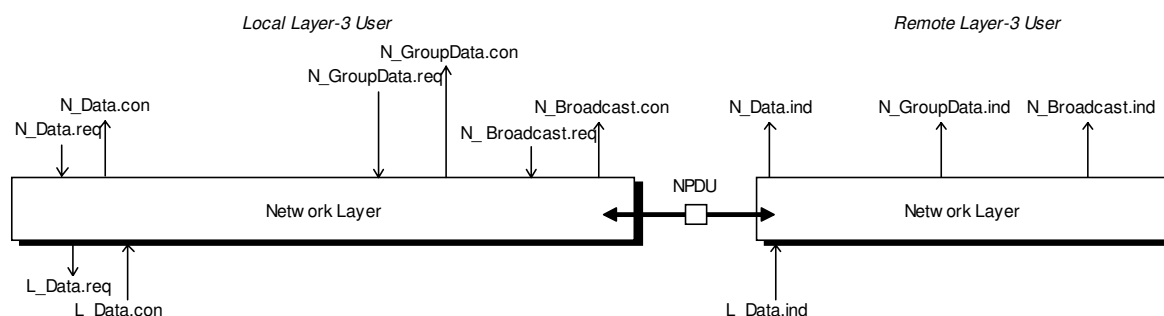


Figure 17: Interactivity of the Network Layer

8.1 NPDU

The NPDU corresponds to the LPDU without the routing counter (Figure 18).

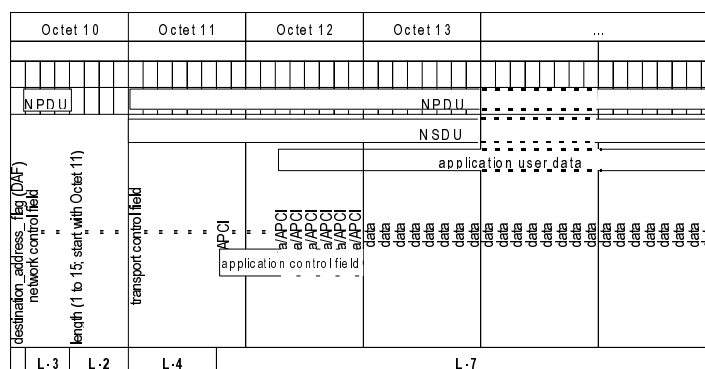


Figure 18: Format of the NPDU

8.2 Network Layer Services

8.2.1 N_Data Service

The N_Data service is confirmed locally. The local user of layer-3 prepares an NSDU for the remote user of layer-3, the destination is addressed with a physical address. The local user of layer-3 applies the N_Data.req primitive to pass the NSDU to the local layer-3. The local layer-3 accepts the service request and passes it with an L_Data.req with DAF = '0' to the local layer-2.

The local layer-3 encodes the NSDU to the LSDU by adding the routing_counter and mapping the arguments MAC-Address,

destination_address and class to the corresponding arguments destination_address and class of the L_Data.req primitive.

The remote layer-3 is mapping an L_Data.ind primitive with DAF = '0' to a N_Data.ind primitive. The argument l_sdu is mapped to the argument n_sdu by removing the routing counter, the arguments MAC-Address, destination_address and class are mapped to the corresponding arguments destination_address and class of the N_Data.ind primitive.

The local layer-3 is mapping the L_Data.con primitive to the N_Data.con primitive. The argument l_status is mapped to the corresponding argument n_status of the N_Data.con primitive.

```
N_Data.req(MAC-Address, destination_address, class, n_sdu)
MAC-Address          destination Address of MAC-Layer
destination_address:  physical address of the destination
class:               system, alarm, high or low priority
n_sdu:              this is the user data to be transferred by
                    layer-3

N_Data.con(MAC-Address, destination_address, n_status)
MAC-Address          destination Address of MAC-Layer
destination_address:  physical address of the destination
n_status: OK;        N_Data sent successfully with L_Data
                    not_ok; transmission of the associated L_Data
                    request frame didn't succeed

N_Data.ind(MAC-Address, source_address, destination_address, class, n_sdu)
MAC-Address          source Address of MAC-Layer
source_address:      physical address of the EIB end device that
                    requested the N_Data service
destination_address:  the physical address of this device
class:              system, alarm, high or low priority
n_sdu:              this is the user data that has been
                    transferred by layer-3
```

8.2.2 N_Groupdata Service

The N_Groupdata service is confirmed locally. The local user of layer-3 prepares an NSDU for the remote user of layer-3, the destination is addressed with a group address. The local user of layer-3 is applying the N_GROUPDATA.req primitive to pass the NSDU to the local layer-3. The local layer-3 accepts the service request, adds the MAC-Address for broadcast and passes it with an L_Data.req with DAF = '1' to the local layer-2.

The local layer-3 is encoding the NSDU to the LSDU by adding the routing_counter and mapping the arguments destination_address and class to the corresponding arguments destination_address and class of the L_Data.req primitive.

The remote layer-3 is mapping an L_Data.ind primitive with DAF = '1' and destination_address<>'0' to a N_Groupdata.ind primitive. The argument l_sdu is mapped to the argument n_sdu by removing the routing counter, the arguments destination_address and class are mapped to the corresponding arguments destination_address and class of the N_Groupdata.ind primitive. The argument MAC-Address from the remote layer-2 is dropped.

The local layer-3 is mapping the L_Data.con primitive to the N_Groupdata.con primitive. The argument l_status is mapped to the corresponding argument

n_status of the N_Groupdata.con primitive. The argument MAC-Address from the local layer-2 is dropped.

```
N_Groupdata.req(destination_address, class, n_sdu)
  destination_address:  group address of the destination
  class:                system, alarm, high or low priority
  n_sdu:                this is the user data to be transferred by
                       layer-3

N_Groupdata.con(destination_address, n_status)
  destination_address:  group address of the destination
  n_status: OK;         N_Groupdata sent successfully with L_Data
                       service
                       not_ok; transmission of the associated L_Data
                       request frame didn't succeed

N_Groupdata.ind(source_address, destination_address, class, n_sdu)
  source_address:       physical address of the EIB end device that
                       requested the N_Groupdata service
  destination_address:  the addressed group address of this device
  class:                system, alarm, high or low priority
  n_sdu:                this is the user data that has been
                       transferred by layer-3
```

8.2.3 N_Broadcast Service

The N_Broadcast service is confirmed locally. The local user of layer-3 prepares an NSDU for all the remote user of layer-3, the destination is addressed with the broadcast address (destination address = '0' and DAF = '1'). The local user of layer-3 applies the N_Broadcast.req primitive to pass the NSDU to the local layer-3. The local layer-3 accepts the service request and passes it with an L_Data.req with DAF = '1' to the local layer-2.

The local layer-3 is encodes the NSDU to the LSDU by adding the routing_counter, the argument MAC-Address for broadcast and mapping the arguments destination_address and class to the corresponding arguments destination_address and class of the L_Data.req primitive.

The remote layer-3 is mapping an L_Data.ind primitive with DAF = '1' and destination_address = '0' to a N_Broadcast.ind primitive. The argument l_sdu is mapped to the argument n_sdu by removing the routing counter, the argument class is mapped to the corresponding argument class of the N_Broadcast.ind primitive. The argument MAC-Address from the remote layer-2 is dropped.

The local layer-3 is mapping the L_Data.con primitive to the N_Broadcast.con primitive. The argument l_status is mapped to the corresponding argument n_status of the N_Broadcast.con primitive. The argument MAC-Address from the remote layer-2 is dropped.

```
N_Broadcast.req(class, n_sdu)
  class:                system, alarm, high or low priority
  n_sdu:                this is the user data to be transferred by
                       layer-3
```

<code>N_Broadcast.con(n_status)</code>	
<code>n_status: OK;</code>	N_Broadcast sent successfully with L_Data service
<code>not_ok;</code>	transmission of the associated L_Data request frame didn't succeed
<code>N_Broadcast.ind(source_address, class, n_sdu)</code>	
<code>source_address:</code>	physical address of the EIB end device that requested the N_Broadcast service
<code>class:</code>	system, alarm, high or low priority
<code>n_sdu:</code>	this is the user data that has been transferred by layer-3

8.3 Parameters of Layer-3

The layer-3 only needs information about the device: either EIB end device or bridge or router.

8.4 State Machine of Layer-3

The state machine of layer-3 in an EIB end device is mapping services as described above. The value of the routing counter is set to six when layer-4 is applying a layer-3 request primitive.

8.5 The Layer-3 of a Bridge

Bridges and routers do also have a layer-3 but their layer-3 state machine differs from EIB end devices.

If an L_Data.ind with DAF = '1' and routing_counter in [1...6] is received, the bridge decrements the routing_counter and transmits the service parameters of the L_Data.ind with the corresponding service parameters (source address, destination_address, DAF, class, l_sdu) of a L_Data.req to the other side.

If an L_Data.ind with DAF = '0' or routing_counter equals seven is received, the bridge transmits the service parameters of the L_Data.ind with the corresponding service parameters of a L_Data.req to the other side.

Otherwise the layer-3 of the bridge is discarding the L_Data.ind.

8.6 The Layer-3 of a Router

Routers behave like bridges but there are two additional actions:

- checks if the destination address equals to the physical address of the router, then identical to EIB end device
- checks if the destination address is a group address and is listed in the routing table

If an L_Data.ind with DAF = '1' and routing_counter in [1...6] is received and the destination address is listed in the routing table, the router decrements the routing_counter and transmits 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 DAF = '1' and the routing_counter equals seven is received and the destination address is listed in the routing table, the router transmits 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 DAF = '0' and destination address equal to the physical address of the router is received, the router is processing the L_Data.ind identical to an EIB end device.

If an L_Data.ind with DAF = '0' is received and the destination address is listed in the routing table, the router transmits the service parameters of the L_Data.ind with the corresponding service parameters of a L_Data.req to the other side.

Otherwise the layer-3 of the router is discarding the L_Data.ind.

THIS PAGE LEFT BLANK INTENTIONALLY

9 Transport Layer

9.1 Communication Relationships

The transport layer (layer-4) provides a reliable data transmission over communication relationships. Communication relationships are logical channels that connect users of layer-4 with each other. Layer-4 provides four different types of communication relationships:

- one-to-many connection-less (multicast)
- one-to-all connection-less (broadcast)
- one-to-one connection-less
- one-to-one connection-oriented

Communication relationships are identified by a local communication relationship identifier (cr_id) which shall be unique for all communication relationships of this EIB end device. The layer-4 converts the cr_id into the destination_address and vice versa using a communication relationship list. Every communication relationship type provides specific layer-4 services.

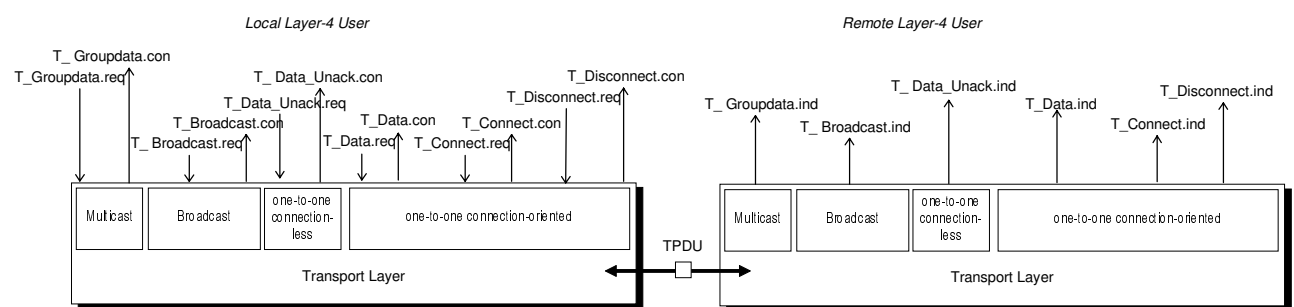


Figure 19: Interactivity of the Transport Layer

9.1.1 One-to-many connection-less (Multicast) Communication Relationship

A multicast communication relationship connects group-objects that belong to the same group. Group-objects may be distributed to a number of EIB end devices. Each EIB end device may be transmitter. More than one group-object may exist in an EIB end device. The group-objects of an EIB end device may belong to the same or to different groups.

The following layer-4 service can only be used on a multicast communication relationship:

- T_Groupdata

9.1.2 One-to-all connection-less (Broadcast) Communication Relationship

The broadcast communication relationship connects a single EIB end device with all communication partners. The single EIB end device is always a transmitter, the communication partners are always receiver.

The following layer-4 service can only be used on a broadcast communication relationship:

- T_Broadcast

9.1.3 One-to-one connection-less Communication Relationship

Every EIB end device has a one-to-one connection-less communication relationship with every other EIB end device. A one-to-one connection-less communication relationships shall not be used if the connection-oriented communication relationship is established to the same partner at the same time. The following layer-4 service can only be used on a one-to-one connection-less communication relationship:

- T_Data_Unack

9.1.4 One-to-one connection-oriented Communication Relationship

An EIB end device only has a single one-to-one connection-oriented communication relationship. The following layer-4 services can only be used on a connection-oriented communication relationship:

- T_Connect
- T_Data
- T_Disconnect

The user of this type shall establish the connection before it can be used. The user may release the connection if it isn't needed any more. The layer-4 provides a supervision of the connection with a connection-time-out-timer. If the timer expires or if an unrecoverable error occurs, the layer-4 will release the connection immediately. Layer-4 also provides a reliable end-to-end transmission over bridges and routers on the connection-oriented communication relationship. T_Data services are repeated up to three times if the T_Data.req is not acknowledged from the remote layer-4 entity within an acknowledgment-time-out-time. Repetitions of T_Data services are detected using a sequence number. Parallel services are not allowed on a connection-oriented communication relationship. The connection-oriented communication relationship is processed according to the layer-4 state machine described in 9.3.4.

9.2 TPDU

The TPDU is shown in the following figure (Figure 20).

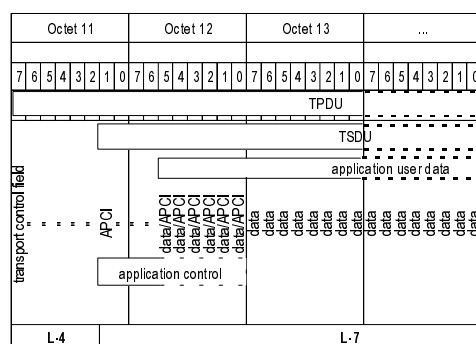


Figure 20: Format of the TPDU

The TPDU corresponds to the NPDU, but reduced by the network control field. The transport control field is encoded and decoded by layer-4 and contains the layer-4 service codes and the sequence_number:

Octet 10								Octet 11							
								transport control field							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
destination_address_flag (DAF)															
								1		0	0	0	0	0	
								1		0	0	0	0	0	
								0		0	0	0	0	0	
								0		0	1	SeqNo	SeqNo	SeqNo	
								0		1	0	0	0	0	0
								0		1	0	0	0	0	1
								0		1	0	0	0	0	1
								0		1	1	SeqNo	SeqNo	SeqNo	1
								0		1	1	SeqNo	SeqNo	SeqNo	1
								0		1	1	SeqNo	SeqNo	SeqNo	1

Figure 21: Transport Control Field

9.3 Transport Layer Services

All the layer-4 services provide a confirmation to the requester (user of layer-4). The confirmation of the T_Data service indicates that the remote layer-4 entity did acknowledge the T_Data.req. The confirmation of the other layer-4 services is caused by the local confirmation of layer-2.

The user of layer-4 shall not request a service primitive before the preceding request is confirmed by layer-4, i.e. no parallel services allowed.

9.3.1 T_Groupdata Service

The T_Groupdata service is applied by the user of layer-4, to transmit a TSDU over a multicast communication relationship to one or more remote partners. The T_Groupdata service is neither acknowledged nor confirmed by the remote layer-4 entity. The confirmation is a local confirmation caused by the L_Data.con of layer-2.

The local user of layer-4 is preparing a TSDU for the remote user. The local user of layer-4 is applying the T_Groupdata.req primitive to pass the TSDU to the local layer-4. The destination is defined by the cr_id. The local layer-4 is mapping the cr_id to the group address using a communication reference list. The local layer-4 accepts the service request and passes it with a N_Groupdata.req to the local layer-3.

The local layer-4 is encoding the TSDU to the NSDU and is mapping the arguments group_address and class to the corresponding arguments destination_address and class of the N_Groupdata.req primitive.

The remote layer-4 is mapping a N_Groupdata.ind primitive to a T_Groupdata.ind primitive. The remote layer-4 is mapping the destination_address to the cr_id using a communication reference list. The argument n_sdu is mapped to the argument t_sdu, the argument class is mapped to the corresponding argument class of the T_Groupdata.ind primitive.

Prior to passing a T_Groupdata.con primitive to the local user, the local layer-4 needs a N_Groupdata.con from the local layer-3. If the confirmation is positive (n_status = OK), the local layer-4 passes a positive T_Groupdata.con (t_status = OK) to the local user. If the confirmation is negative (n_status = not_ok), the local layer-4 passes a T_Groupdata.con (t_status = not_ok) to the local user indicating that the transmission of the associated N_Groupdata.req didn't succeed.

```
T_Groupdata.req(cr_id, class, tsdu)
  cr_id:          identifier of the communication relationship
  class:          system, alarm, high or low priority
  tsdu:           this is the user data to be transferred by
                  layer-4
```

```
T_Groupdata.con(cr_id, t_status)
  cr_id:          identifier of the communication relationship
  t_status: OK;   T_Groupdata sent successfully with
                  N_Groupdata service
                  not_ok; transmission of the associated N_Groupdata
                  request frame didn't succeed
```

```
T_Groupdata.ind(cr_id, class, t_sdu)
  cr_id:          identifier of the communication relationship
  class:          system, alarm, high or low priority
  t_sdu:          this is the user data that has been
                  transferred by layer-4
```

9.3.2 T_Broadcast Service

The T_Broadcast service is applied by the user of layer-4, to transmit a TSDU over a connection-less communication relationship to all remote partners. The T_Broadcast service is neither acknowledged nor confirmed by the remote layer-4 entity. The confirmation is a local confirmation caused by the L_Data.con of layer-2.

The local user of layer-4 is preparing a TSDU for all the remote users of layer-4. The local user of layer-4 is applying the T_Broadcast.req primitive to pass the TSDU to the local layer-4. The local layer-4 accepts the service request and passes it with a N_Broadcast.req to the local layer-3.

The local layer-4 is encoding the TSDU to the NSDU and is mapping the argument class to the corresponding argument class of the N_Broadcast.req primitive.

The remote layer-4 is mapping an N_Broadcast.ind primitive to a T_Broadcast.ind primitive. The argument n_sdu is mapped to the argument t_sdu, the argument class is mapped to the corresponding argument class of the T_Broadcast.ind primitive.

Prior to passing a T_Broadcast.con primitive to the local user, the local layer-4 needs a N_Broadcast.con from the local layer-3. If the confirmation is positive (n_status = OK), the local layer-4 passes a positive T_Data.con (t_status = OK) to the local user. If the confirmation is negative (n_status = not_ok), the local layer-4 passes a T_Data.con (t_status = not_ok) to the local user indicating that the transmission of the associated N_Broadcast.req didn't succeed.

```
T_Broadcast.req(class, tsdu)
  class:          system, alarm, high or low priority
  tsdu:           this is the user data to be transferred by
                  layer-4

T_Broadcast.con(t_status)
  t_status: OK;    T_Broadcast sent successfully with
                  N_Broadcast service
                  not_ok; transmission of the associated N_Broadcast
                  request frame didn't succeed

T_Broadcast.ind(source_address, class, t_sdu)
  source_address: physical address of the EIB end device that
                  requested the T_Broadcast service
  class:          system, alarm, high or low priority
  t_sdu:          this is the user data that has been
                  transferred by layer-4
```

9.3.3 T_Data_Unack

The T_Data_Unack service is applied by the user of layer-4, to transmit a TSDU over a connection-less one-to-one communication relationship to exactly one remote partner. The T_Data_Unack service is neither acknowledged nor confirmed by the remote layer-4 entity. The confirmation is a local confirmation caused by the L_Data.con of layer-2.

The local user of layer-4 is preparing a TSDU for the remote user. The local user of layer-4 is applying the T_Data_Unack.req primitive to pass the TSDU to the local layer-4. The destination is defined by the destination_address.

The local layer-4 is encoding the TSDU to the NSDU and is mapping the arguments destination_address and class to the corresponding arguments of the N_Data.req primitive.

The remote layer-4 is mapping an N_Data.ind primitive containing a T_DATA_UNACK_PDU to a T_Data_Unack.ind primitive. The argument n_sdu is mapped to the argument t_sdu, the arguments class and physical address are mapped to the corresponding argument class of the T_Data_Unack.ind primitive.

Prior to passing a T_Data_Unack.con primitive to the local user, the local layer-4 needs a N_Data.con from the local layer-3. If the confirmation is positive (n_status = OK), the local layer-4 passes a positive T_Data_Unack.con (t_status = OK) to the local user. If the confirmation is negative (n_status = not_ok), the local layer-4 passes a T_Data_Unack.con (t_status = not_ok) to the local user indicating that the transmission of the associated N_Data.req didn't succeed.

```
T_Data_Unack.req(address, class, tsdu)
  destination_address: physical address of the device is addressed.
  class:              system, alarm, high or low priority
  tsdu:              this is the user data to be transferred by
                    layer-4
```

```
T_Data_Unack.con(address, n_status)
  destination_address:    physical address of the device is addressed
  n_status: OK;           T_Data_Unack sent successfully with N_Data
                           service
                           not_ok;           transmission of the associated N_Data
                                           request frame didn't succeed

T_Data_Unack.ind(address, class, t_sdu)
  destination_address:    physical address of the device which
                           initiates the service
  class:                  system, alarm, high or low priority
  t_sdu:                  this is the user data that has been
                           transferred by layer-4
```

9.3.4 T_Connect Service

The T_Connect service is applied by the user of layer-4, to establish a transport connection on a connection-oriented communication relationship. The T_Connect primitives are mapped to N_Data primitives and vice versa according to the layer-4 state machine described in 9.3.4. The local layer-4 accepts the service request only if the connection is not established (state CLOSED) and tries to send the T_CONNECT_REQ_PDU to the remote layer-4 with a N_Data.req. The destination address shall be a physical address. The T_Connect service is confirmed by the remote layer-4 with a N_Data primitives with a T_Connect.confirm PDU. The local confirmation from the local layer-3 is dropped.

```
T_Connect.req(address)
  address:                physical address of the device where the
                           transport connection shall be established

T_Connect.con(cr_id)
  cr_id:                  connection is established,
                           identifier of the communication relationship

T_Connect.ind(cr_id)
  cr_id:                  connection established, remote initiator
                           identifier of the communication relationship
```

9.3.5 T_Disconnect Service

The T_Disconnect service is applied by the user of layer-4, to release a transport connection on a connection-oriented communication relationship. The T_Disconnect primitives are mapped to N_Data primitives and vice versa according to the layer-4 state machine described in 9.3.4. The T_Disconnect service is neither acknowledged nor confirmed by the remote layer-4 entity. The confirmation is a local confirmation caused by the L_Data.con of layer-2.

The T_Disconnect.ind primitive may also be caused by the layer-4 entity in order to indicate a protocol error.

```
T_Disconnect.req(cr_id)
  cr_id:                  identifier of the communication relationship
                           to be released

T_Disconnect.con(cr_id)
                           Connection released, state = CLOSED

T_Disconnect.ind(cr_id)
                           Connection released, state = CLOSED, caused
                           by an error detected by the local layer-4 or
                           a T_DISCONNECT_REQ_PDU was received
```

9.3.6 T_Data Service

The T_Data service is applied by the user of layer-4, to transmit a TSDU over a transport connection on a connection-oriented communication relationship to a remote partner. The T_Data service is acknowledged with a T_DATA_ACK_PDU by the remote layer-4 entity. The T_Data primitives are mapped to N_Data primitives and vice versa according to the layer-4 state machine described in 9.3.4.

The local user of layer-4 is preparing a TSDU for the remote user. The local user of layer-4 is applying the T_Data.req primitive to pass the TSDU to the local layer-4. The local layer-4 accepts the service request only if the connection is established (state OPEN_IDLE) and tries to send the TSDU to the remote layer-4 with a N_Data.req. The destination address shall be a physical address. The local layer-4 passes a T_Data.con primitive to the local user that indicates either a correct data transfer or it passes a T_Disconnect.ind primitive to the local user that indicates an erroneous data transfer.

Prior to passing the confirmation to the local user, the local layer-4 needs an acknowledgment from the remote layer-4. If the acknowledgment is a positive acknowledgment (T_DATA_ACK_PDU), the local layer-4 passes a T_Data.con to the local user. If the acknowledgment is a negative acknowledgment (T_DATA_NAK_PDU), the local layer-4 passes a T_Disconnect.ind primitive to the local user indicating that the connection is released (state = CLOSED) caused by an error.

The remote layer-4 will only accept the N_Data.ind with the T_DATA_REQ_PDU received, if the connection is established, i.e. in the states OPEN_IDLE, OPEN_WAIT_FOR_DATA_ACK, OPEN_REPEAT_T_DATA_REQ. Therefore mutual T_Data services are allowed on a connection-oriented communication relationship.

The local layer-4 repeats the transmission of the T_DATA_REQ_PDU up to 3 times with an acknowledgment time-out time of 3 s. If it fails, the local layer-4 passes a T_Disconnect.ind primitive to the local user indicating that the connection is released (state = CLOSED).

T_Data.req(cr_id, class, tsdu)	
cr_id:	identifier of the communication relationship
class:	system, alarm, high or low priority
tsdu:	this is the user data to be transferred by layer-4
T_Data.con(cr_id)	transmission successful
cr_id:	identifier of the communication relationship
T_Data.ind(cr_id, class, tsdu)	
cr_id:	identifier of the communication relationship
class:	system, alarm, high or low priority
tsdu:	this is the user data that has been transferred by layer-4

9.4 Parameters of Layer-4

The communication relationship list is the only parameter of layer-4. The implementation of this parameter, e.g. table or algorithmic is up to the manufacturer. The latter could be achieved by mapping the address of the communication partner (group address or physical address with DAF flag) to a 13 bit `cr_id`.

Communication relationship list: maps `cr_ids` to a destination addresses and vice versa.

9.5 State Machine of Layer-4

The layer-4 state machine is processing the services `T_Connect`, `T_Disconnect` and `T_Data`. Other layer-4 services are directly mapped as described for each individual service independent from the actual state of the layer-4 state machine. Invalid PDUs are ignored.

Events caused by the user of layer-4:

```
T_Connect.req from user
\ -
T_Disconnect.req from user
\ -
T_Data.req from user
\ -
```

Events caused inside layer-4:

```
Connection_Timeout
\ -
Acknowledgement_Timeout
\ -
```

Events caused by layer-3:

```
N_Data.ind from layer-3
\source_address<>connection_address
N_Data.ind from layer-3
\source_address=connection_address & nsdu=T_CONNECT_REQ_PDU
N_Data.ind from layer-3
\source_address=connection_address & nsdu=T_CONNECT_CON_PDU
N_Data.ind from layer-3
\source_address=connection_address & nsdu=T_DISCONNECT_REQ_PDU
N_Data.ind from layer-3
\source_address=connection_address & nsdu=T_DATA_REQ_PDU &
N_Data.ind from layer-3
\source_address=connection_address & nsdu=T_DATA_ACK_PDU
N_Data.ind from layer-3
\source_address=connection_address & nsdu=T_DATA_NAK_PDU
N_Data.con from layer-3
\ -
```

Local variables of layer-4:

<code>connection_address</code>	used to store the actual physical address of the partner
<code>SeqNoSend</code>	binary 4 bit value, used to handle the sequence number of the <code>T_DATA_XXX_PDU</code>
<code>SeqNoRcv</code>	binary 4 bit value, used to handle the sequence number of the <code>T_DATA_XXX_PDU</code>

connection_timeout_timer	time interval of 6 s; starts with transition CLOSED→CONNECTING; stops with transition DISCONNECTING→CLOSED; restarts if N_Data.req is applied in the state machine or N_Data.ind received with source_address=connection_address
acknowledgment_timeout_timer	time interval of 3 s; starts with transition OPEN_IDLE→OPEN_WAIT_FOR_T_DATA_A CK; stops if N_DATA.ind & source_address=connection_address & nsdu=T_DATA_ACK_PDU & SeqNo_of_PDU=SeqNoRcv received or with transition into →CLOSED
rep_count	used to count the number of T_DATA_REQ repetitions

The user of layer-4 always gets a confirmation for a request. When reading the state machine, keep in mind that the user cannot request a second service primitive before getting the confirmation to the preceding primitive, i.e. no parallel services allowed.

The event N_Data.con after the transmission of a N_Data.req is not described in all of the states. In states where this event is not mentioned, a subordinate state is active where the state machine is waiting for either a time-out or the a N_Data.con. As soon as the N_Data.con or the time-out appears, the state machine is switching back to the associated superordinate state.