BS EN 62056-4-7:2016

BSI Standards Publication

Electricity metering data exchange — The DLMS/COSEM suite

Part 4-7: DLMS/COSEM transport layer for IP networks

bsi.

National foreword

This British Standard is the UK implementation of EN 62056-4-7:2016. It is identical to IEC 62056-4-7:2015. It supersedes BS EN 62056-47:2007 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee PEL/13, Electricity Meters.

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

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

Published by BSI Standards Limited 2016

ISBN 978 0 580 86668 5
ICS 17.220.01; 35.100.40; 35.110; 91.140.50

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

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 December 2016.

Amendments/corrigenda issued since publication

Date	Text affected

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN 62056-4-7

December 2016

ICS 17.220; 35.110; 91.140.50

Supersedes EN 62056-47:2007

English Version

Electricity metering data exchange - The DLMS/COSEM suite - Part 4-7: DLMS/COSEM transport layer for IP networks (IEC 62056-4-7:2015)

Échange des données de comptage de l'électricité - La suite DLMS/COSEM - Partie 4-7: Couche transport DLMS/COSEM pour réseaux IP (IEC 62056-4-7:2015)

Datenkommunikation der elektrischen Energiemessung - DLMS/COSEM - Teil 4-7: DLMS/COSEM Transportschicht für IP-Netzwerke (IEC 62056-4-7:2015)

This European Standard was approved by CENELEC on 2015-06-24. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

Ref. No. EN 62056-4-7:2016 E

European foreword

The text of document 13/1570/CDV, future edition 1 of IEC 62056-4-7, prepared by IEC/TC 13 "Electrical energy measurement and control" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 62056-4-7:2016.

The following dates are fixed:

• latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement	(dop)	2017-06-09
• latest date by which the national standards conflicting with the document have to be withdrawn	(dow)	2019-12-09

This document supersedes EN 62056-47:2007.

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

This document has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association.

Endorsement notice

The text of the International Standard IEC 62056-4-7:2015 was approved by CENELEC as a European Standard without any modification.

Annex ZA
(normative)

Normative references to international publications with their corresponding European publications

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

NOTE 1 When an International Publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

NOTE 2 Up-to-date information on the latest versions of the European Standards listed in this annex is available here: www.cenelec.eu

Publication	Year	Title	EN/HD	Year
IEC 60050-300	2001	International Electrotechnical Vocabulary (IEV) - Electrical and electronic measurements and measuring instruments - Part 311: General terms relating to measurements - Part 312: General terms relating to electrical measurements - Part 313: Types of electrical measuring instruments - Part 314: Specific terms according to the type of instrument	-	-
IEC/TR 62051	1999	Electricity metering - Glossary of terms	-	-
IEC/TR 62051-1	2004	Electricity metering - Data exchange for meter reading, tariff and load control - Glossary of terms - Part 1: Terms related to data exchange with metering equipment using DLMS/COSEM	-	-
IEC 62056-5-3	2013	Electricity metering data exchange - The DLMS/COSEM suite - Part 5-3: DLMS/COSEM application layer	EN 62056-5-3	2014 [1)]
IEC 62056-6-2	2013	Electricity metering data exchange - The DLMS/COSEM suite - Part 6-2: COSEM interface classes	EN 62056-6-2	2013 [2)]
IEC 62056-9-7	2013	Electricity metering data exchange - The DLMS/COSEM suite - Part 9-7: Communication profile for TCP-UDP/IP networks	EN 62056-9-7	2013
STD 0006	-	User Datagram Protocol	-	-
STD 0007	-	Transmission Control Protocol	-	-

[1)] Superseded by EN 62056-5-3:2016 (IEC 62056-5-3:2016): DOW = 2019-12-09.

[2)] Superseded by EN 62056-6-2:2016 (IEC 62056-6-2:2016): DOW = 2019-12-09.

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

ELECTRICITY METERING DATA EXCHANGE – THE DLMS/COSEM SUITE –

Part 4-7: DLMS/COSEM transport layer for IP networks

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as “IEC Publication(s)”). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC 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 National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of 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.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees 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, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) 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.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this International Standard may involve the use of a maintenance service concerning the stack of protocols on which the present standard IEC 62056-4-7 is based.

The IEC takes no position concerning the evidence, validity and scope of this maintenance service.

The provider of the maintenance service has assured the IEC that he is willing to provide services under reasonable and non-discriminatory terms and conditions for applicants throughout the world. In this respect, the statement of the provider of the maintenance service is registered with the IEC. Information may be obtained from:

DLMS User Association
Zug/Switzerland
www.dlms.com

International Standard IEC 62056-4-7 has been prepared by IEC technical committee 13: Electrical energy measurement and control.

This first edition cancels and replaces IEC 62056-47 published in 2006. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

a) This standard is applicable now both for IP4 and IPv6 networks;

b) Latest editions of the IEC 62056 suite are referenced.

c) DLMS/COSEM IANA-registered port numbers added.

The text of this standard is based on the following documents:

CDV	Report on voting
13/1570/CDV	13/1595/RVC

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

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

A list of all parts in the IEC 62056 series, published under the general title *Electricity metering data exchange – The DLMS/COSEM suite*, can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

This standard specifies the DLMS/COSEM transport layer for IP (IPv4 or IPv6) networks. It shall be read together with IEC 62056-9-7:2013, *Electricity metering data exchange – The DLMS/COSEM suite – Part 9-7: Communication profile for TCP-UDP/IP networks.*

ELECTRICITY METERING DATA EXCHANGE – THE DLMS/COSEM SUITE –

Part 4-7: DLMS/COSEM transport layer for IP networks

1 Scope

This part of IEC 62056 specifies a connection-less and a connection oriented transport layer (TL) for DLMS/COSEM communication profiles used on IP networks.

These TLs provide OSI-style services to the service user DLMS/COSEM AL. The connection-less TL is based on the Internet Standard User Datagram Protocol (UDP). The connection-oriented TL is based on the Internet Standard Transmission Control Protocol (TCP).

The DLMS/COSEM TL consists of the UDP or TCP transport layer TCP and an additional sublayer, called wrapper.

Annex A shows how the OSI-style TL services can be converted to and from UDP and TCP function calls.

2 Normative references

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

IEC 60050-300:2001, *International Electrotechnical Vocabulary (IEV) – Electrical and electronic measurements and measuring instruments – Part 311: General terms relating to measurements – Part 312: General terms relating to electrical measurements – Part 313: Types of electrical measuring instruments – Part 314: Specific terms according to the type of instrument*

IEC TR 62051:1999, *Electricity metering – Glossary of terms*

IEC TR 62051-1:2004, *Electricity metering – Data exchange for meter reading, tariff and load control – Glossary of terms – Part 1: Terms related to data exchange with metering equipment using DLMS/COSEM*

IEC 62056-5-3:2013, *Electricity metering data exchange – The DLMS/COSEM suite – Part 5-3: DLMS/COSEM application layer*[1]

IEC 62056-6-2:2013, *Electricity metering data exchange – The DLMS/COSEM suite – Part 6-2: COSEM interface classes*[2]

---

[1] Edition 2 of IEC 62056-5-3 to be published.

[2] Edition 2 of IEC 62056-6-2 to be published.

IEC 62056-9-7:2013, *Electricity metering data exchange – the DLMS/COSEM suite – Part 9-7: Communication profile for TCP-UDP/IP networks*

STD 0006, *User Datagram Protocol. Edited by Jon Postel, August 1980.* Available from: http://www.faqs.org/rfcs/std/std6.html

STD 0007, *Transmission Control Protocol. Edited by Jon Postel, September 1981.* Available from: http://www.faqs.org/rfcs/std/std7.html

NOTE See also Bibliography for other related Internet RFCs.

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 60050-300, IEC TR 62051 and IEC TR 62051-1 apply as well as the following:

3.1.1
application process
an element within a real open system which performs the information processing for a particular application

[SOURCE: ISO/IEC 7498-1: 1994, 4.1.4]

3.1.2
application entity
system-independent application activities that are made available as application services to the application agent, e.g., a set of application service elements that together perform all or part of the communication aspects of an application process

3.2 Abbreviations

APDU	Application Layer Protocol Data Unit
AL	Application Layer
AP	Application Process
AE	Application Entity
COSEM	COmpanion Specification for Energy Metering
DHCP	Dynamic Host Configuration Protocol
DLMS	Device Language Message Sepcification
COSEM_on_IP	The TCP-UDP/IP based COSEM communication profile
FTP	File Transfer Protocol
HTTP	HyperText Transfer Protocol
IANA	Internet Assigned Numbers Authority
IP	Internet Protocol
PDU	Protocol Data Unit
PAR	Positive Acknowledgement with Retransmission
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TL	Transport Layer
UDP	User Datagram Protocol

WPDU Wrapper Protocol Data Unit

4 Overview

In the DLMS/COSEM_on_IP profiles, the DLMS/COSEM AL uses the services of one of these TLs, which use then the services of the Internet Protocol (IP) network layer to communicate with other nodes connected to the abstract IP network.

When used in these profiles, the DLMS/COSEM AL can be considered as an Internet standard application protocol (like the well-known HTTP, FTP or SNMP) and it may co-exist with other Internet application protocols, as it is shown in Figure 1.

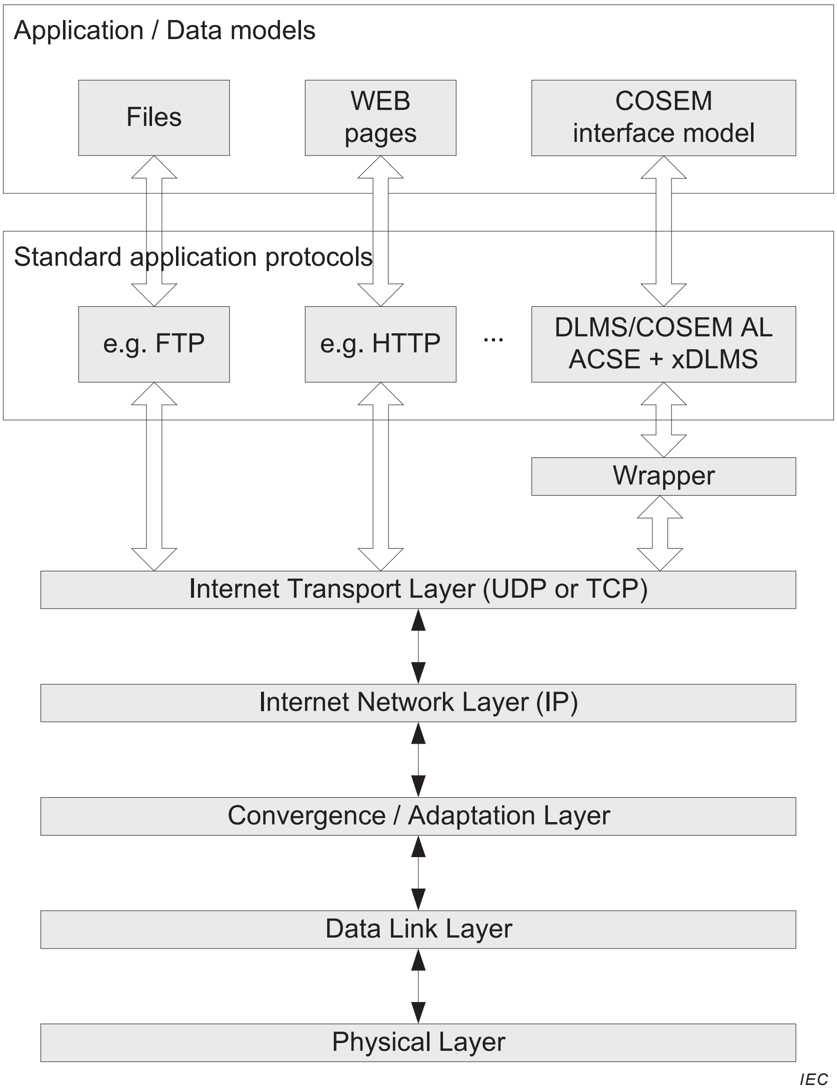


Figure 1 – DLMS/COSEM as a standard Internet application protocol

For DLMS/COSEM, the following port numbers have been registered by the IANA. See http://www.iana.org/assignments/port-numbers.

- dlms/cosem 4059/TCP DLMS/COSEM
- dlms/cosem 4059/UDP DLMS/COSEM

As the DLMS/COSEM AL specified in IEC 62056-5-3 uses and provides OSI-style services, a wrapper has been introduced between the UDP/TCP layers and the DLMS/COSEM AL. Therefore, the DLMS/COSEM TLs consist of a wrapper sublayer and the UDP or TCP TL. The wrapper sublayer is a lightweight, nearly state-less entity: its main function is to adapt the OSI-style service set, provided by the DLMS/COSEM TL, to UDP or TCP function calls and vice versa.

In addition, the wrapper sublayer has the following functions:

- it provides an additional addressing capability (wPort) on top of the UDP/TCP port;
- it provides information about the length of the data transported. This feature helps the sender to send and the receiver to recognize the reception of a complete APDU, which may be sent and received in multiple TCP packets.

As specified in IEC 62056-9-7:2013, Clause 6, the DLMS/COSEM AL is listening only on one UDP or TCP port. On the other hand, as defined in IEC 62056-6-2:2013, 4.7, a DLMS/COSEM physical device may host several client APs or server logical devices. The additional addressing capability provided by the wrapper sublayer allows identifying these APs.

The structure of the DLMS/COSEM TL and their place in COSEM_on_IP is shown in Figure 2.

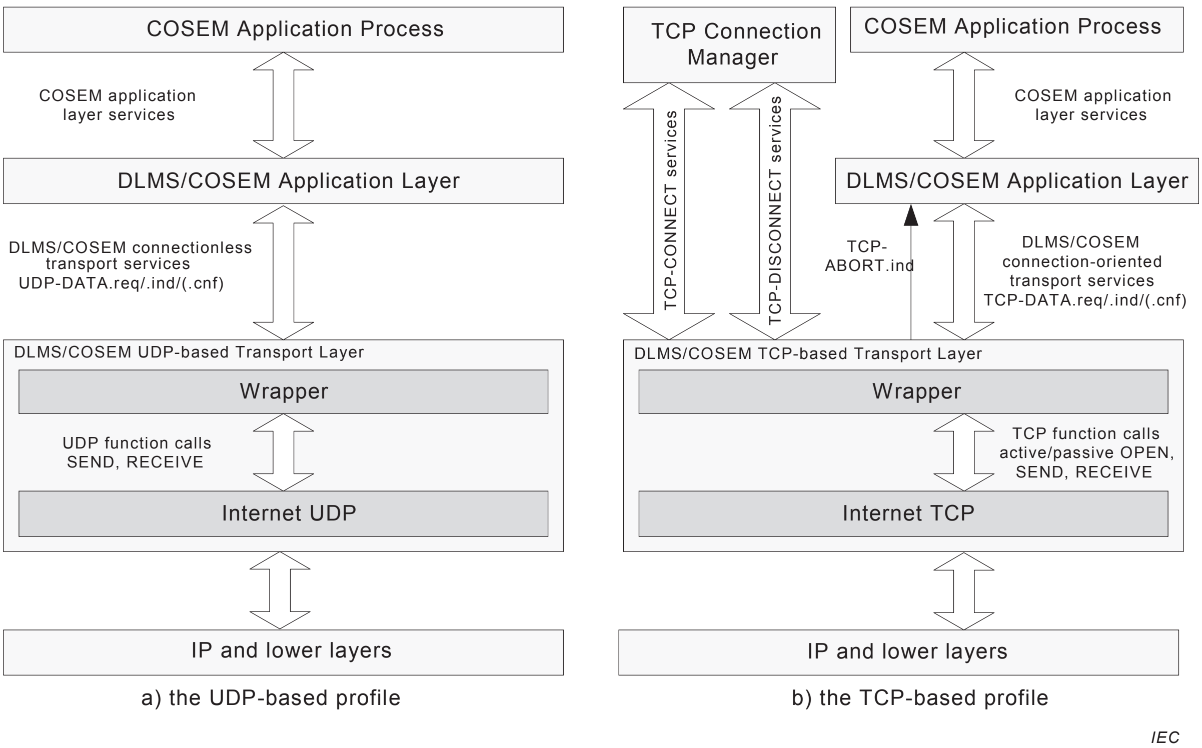


Figure 2 – Transport layers of the DLMS/COSEM_on_IP profile

The service user of both the UDP-DATA and the TCP-DATA services is the DLMS/COSEM AL. On the other hand, the service user of the TCP-CONNECT and TCP-DISCONNECT services is the TCP Connection Manager Process. The DLMS/COSEM TCP-based TL also provides a TCP-ABORT service to the service user DLMS/COSEM AL.

5 The DLMS/COSEM connection-less, UDP-based transport layer

5.1 General

The DLMS/COSEM connection-less TL is based on the User Datagram Protocol (UDP) as specified in STD 0006.

UDP provides a procedure for application programs to send messages to other programs with a minimum of protocol mechanism. On the one hand, the protocol is transaction oriented, and delivery and duplicate protection are not guaranteed. On the other hand, UDP is simple, it adds a minimum of overhead, it is efficient and easy to use. Several well-known Internet applications, like SNMP, DHCP, TFTP, etc., take advantage of these performance benefits, either because some datagram applications do not need to be reliable or because the required reliability mechanism is ensured by the application itself. Request/response type applications, like a confirmed COSEM application association established on the DLMS/COSEM UDP-based TL, then invoking confirmed xDLMS data transfer services is a good example for this second category. Another advantage of UDP is that being connection-less, it is easily capable of multi- and broadcasting.

UDP basically provides an upper interface to the IP layer, with an additional identification capability, the UDP port number. This allows distinguishing between APs, hosted in the same physical device and identified by its IP address.

NOTE The addressing/identification scheme for the COSEM_on_IP profiles is defined in IEC 62056-9-7:2013, Clause 6.

5.2 Service specification for the DLMS/COSEM UDP-based transport layer

5.2.1 General

The DLMS/COSEM UDP-based TL provides only a data transfer service: the connection-less UDP-DATA service. Consequently, the service specification for this service is the same for both the client and server TLs, as it is shown in Figure 3.

The .request and .indication service primitives are mandatory. The implementation of the local .confirm service primitive is optional.

The xDLMS APDU pre-fixed with the wrapper header shall fit in a single UDP datagram.

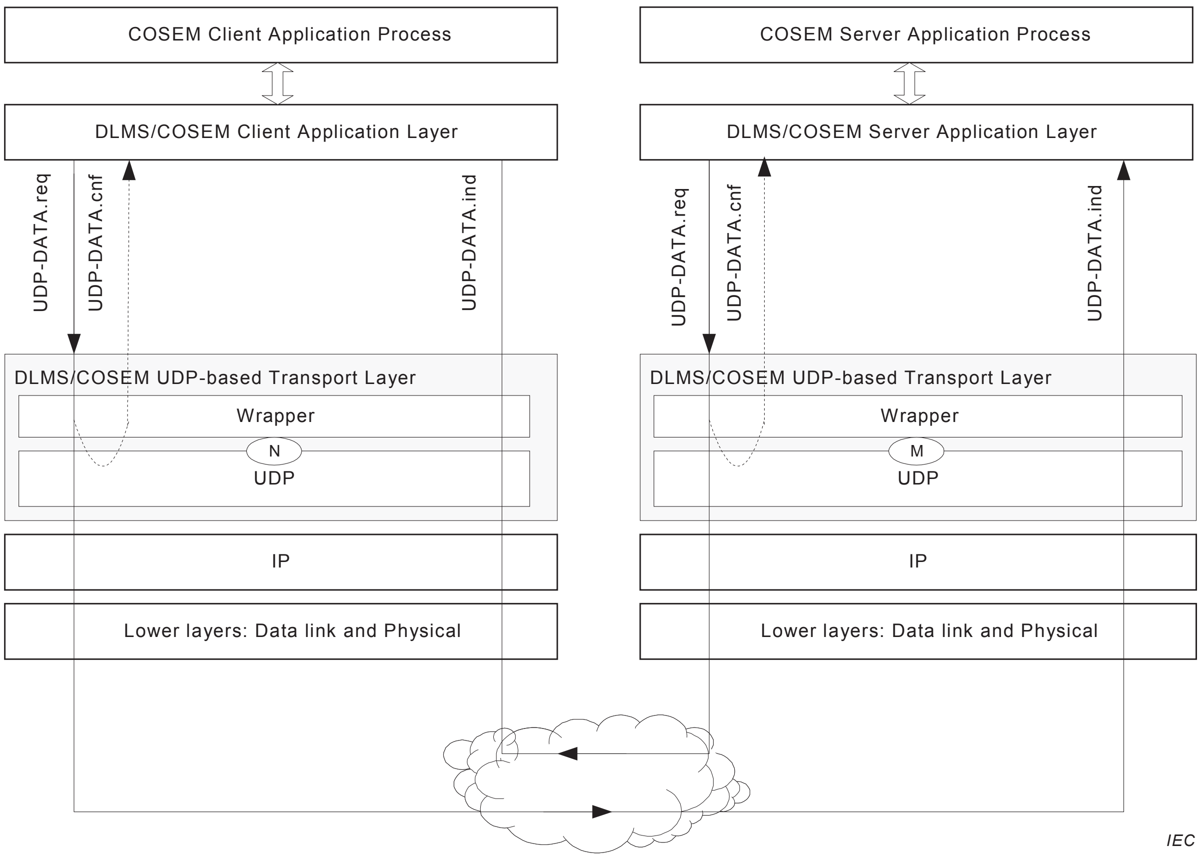


Figure 3 – Services of the DLMS/COSEM connection-less, UDP-based transport layer

5.2.2 The UDP-DATA service

5.2.2.1 UDP-DATA.request

Function

This primitive is the service request primitive for the connection-less mode data transfer service.

Semantics of the service primitive

The primitive shall provide parameters as follows:

UDP-DATA.request (
Local_wPort,
Remote_wPort,
Local_UDP_Port,
Remote_UDP_Port,
Local_IP_Address,
Remote_IP_Address,
Data_Length,
Data
)

The Local_wPort, Local_UDP_Port and Local_IP_Address parameters indicate wrapper Port number, UDP Port number and IP Address parameters belonging to the device / DLMS/COSEM AE requesting to send the Data. The Remote_wPort, Remote_UDP_Port and Remote_IP_Address parameters indicate the wrapper Port number, UDP Port number and IP Address parameters belonging to the device / DLMS/COSEM AE to which the Data is to be transmitted.

The Data_Length parameter indicates the length of the Data parameter in bytes.

The Data parameter contains the xDLMS APDU to be transferred to the peer AL.

Use

The UDP-DATA.request primitive is invoked by either the client or the server DLMS/COSEM AL to request sending an APDU to a single peer AL, or, in the case of multi- or broadcasting, to multiple peer ALs.

The reception of this service primitive shall cause the wrapper sublayer to pre-fix the wrapper header to the APDU received, and then to call the SEND() function of the UDP sublayer with the properly formed WPDU, see at 5.3.2, as DATA. The UDP sublayer shall transmit the WPDU to the peer wrapper sublayer as described in STD 0006.

5.2.2.2 UDP-DATA.indication

Function

This primitive is the service indication primitive for the connection-less mode data transfer service.

Semantics of the service primitive

The primitive shall provide parameters as follows:

```
UDP-DATA.indication (
                        Local_wPort,
                        Remote_wPort,
                        Local_UDP_Port,
                        Remote_UDP_Port,
                        Local_IP_Address,
                        Remote_IP_Address,
                        Data_Length,
                        Data
                      )
```

The Local_wPort, Local_UDP_Port and Local_IP_Address parameters indicate wrapper Port number, UDP Port number and IP Address parameters belonging to the device / DLMS/COSEM AE receiving the Data. The Remote_wPort, Remote_UDP_Port and Remote_IP_Address parameters indicate the wrapper Port number, UDP Port number and IP Address parameters belonging to the device / AE which has sent the data.

The Data_Length parameter indicates the length of the Data parameter in bytes.

The Data parameter contains the xDLMS APDU received from the peer AL.

Use

The UDP-DATA.indication primitive is generated by the DLMS/COSEM UDP based TL to indicate to the service user DLMS/COSEM AL that an APDU from the peer layer entity has been received.

The primitive is generated following the reception of an UDP Datagram by the UDP sublayer, if both the Local_UDP_Port and Local_wPort parameters of the message received contain valid port numbers, meaning that there is a DLMS/COSEM AE in the receiving device bound to the given port numbers. Otherwise, the message received shall simply be discarded.

5.2.2.3 UDP-DATA.confirm

Function

This primitive is the optional service confirm primitive for connection-less mode data transfer service.

Semantics of the service primitive

The primitive shall provide parameters as follows:

```
UDP-DATA.confirm   (
                        Local_wPort,
                        Remote_wPort,
                        Local_UDP_Port,
                        Remote_UDP_Port,
                        Local_IP_Address,
                        Remote_IP_Address,
                        Result
                   )
```

The Local_wPort, Remote_wPort, Local_UDP_Port, Remote_UDP_Port, Local_IP_Address and Remote_IP_Address parameters carry the same values as the corresponding UDP-DATA.request service being confirmed.

The value of the Result parameter indicates whether the DLMS/COSEM UDP-based TL was able to send the requested UDP Datagram (OK) or not (NOK).

Use

The UDP-DATA.confirm primitive is optional. If implemented, it is generated by the DLMS/COSEM TL to confirm to the service user DLMS/COSEM AL the result of the previous UDP-DATA.request. It is locally generated and indicates only whether the Data in the .request primitive could be sent or not. In other words, an UDP-DATA.confirm with Result == OK means only that the Data has been sent, and does not mean that the Data has been (or will be) successfully delivered to the destination.

5.3 Protocol specification for the DLMS/COSEM UDP-based transport layer

5.3.1 General

As it is shown in Figure 2, the DLMS/COSEM UDP-based TL includes the Internet Standard UDP layer, as specified in Internet Standard STD 0006, and the DLMS/COSEM-specific light-weight wrapper sublayer.

In this communication profile, the wrapper sublayer is a state-less entity: its only roles are to ensure source and destination DLMS/COSEM AE identification using the wPort numbers and to provide conversion between the OSI-style UDP-DATA.xxx service invocations and the SEND() and RECEIVE() interface functions provided by the standard UDP.

Although it is not necessary in the UDP-based profile, in order to have the same wrapper protocol control information – in other words the wrapper header – in both TLs, the wrapper sublayer shall also include the Data Length information in the wrapper protocol data unit.

5.3.2 The wrapper protocol data unit (WPDU)

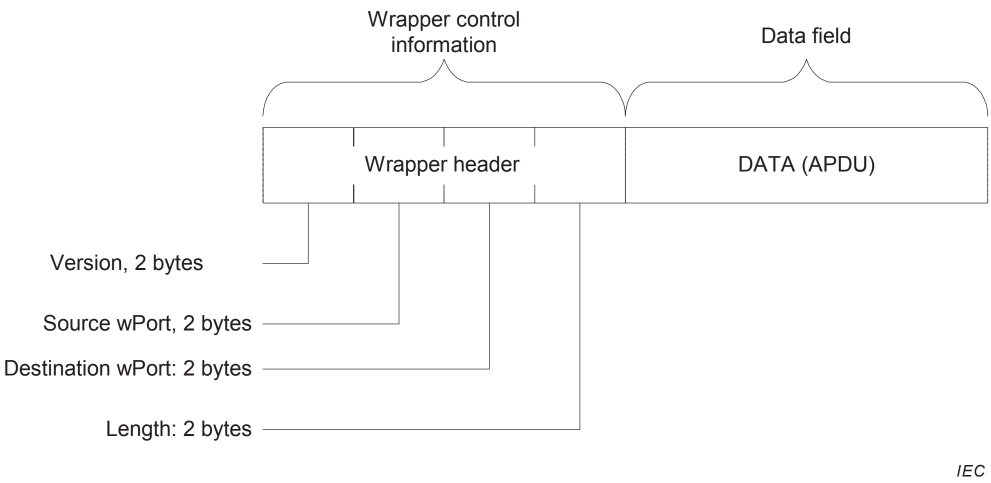
The WPDU consists of two parts:

- the wrapper header part, containing the wrapper control information; and

- the data part, containing the DATA parameter – an xDLMS APDU – of the corresponding UDP-DATA.xxx service invocation.

The wrapper header includes four fields, see Figure 4. Each field is a 16 bit long unsigned integer value.

- Version: carries the version of the wrapper. Its value is controlled by the DLMS UA. The current value is 0x0001. Note, that in later versions the wrapper header may have a different structure;
- Source wPort: carries the wPort number identifying the sending DLMS/COSEM AE;
- Destination wPort: carries the wPort number identifying the receiving DLMS/COSEM AE;
- Data length: indicates the length of the DATA field of the WPDU (the xDLMS APDU transported).



NOTE The maximum length of the APDU should be eight bytes less than the maximum length of the UDP datagram.

Figure 4 – The wrapper protocol data unit (WPDU)

5.3.3 The DLMS/COSEM UDP-based transport layer protocol data unit

In this profile, WPDUs shall be transmitted in UDP Datagrams, specified in Internet Standard STD 0006. They shall encapsulate the WPDU, as shown in Figure 5.

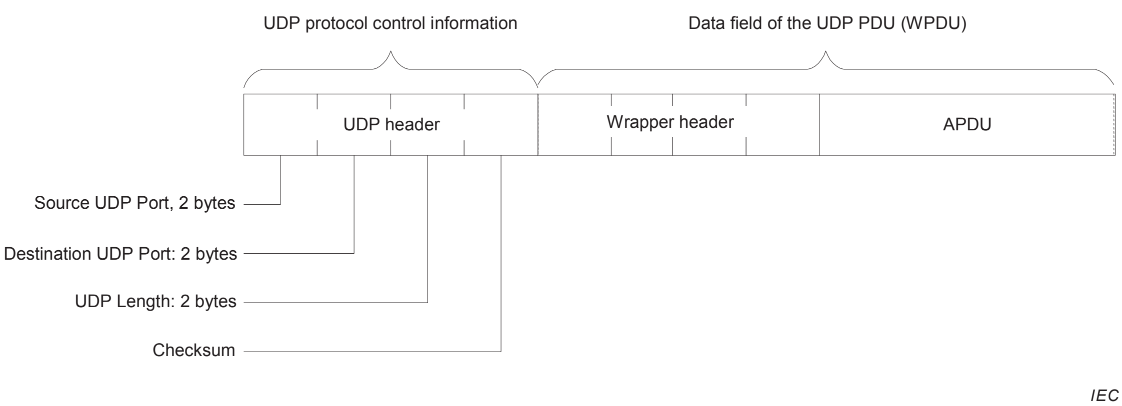


Figure 5 – The DLMS/COSEM connection-less, UDP-based transport layer PDU (UDP-PDU)

From the external point of view, the DLMS/COSEM connection-less TL PDU is an ordinary UDP Datagram: any DLMS/COSEM specific element, including the wrapper-specific header is inside the UDP Data field. Consequently, standard UDP implementations can be (re-)used to easily implement this TL.

The Source and Destination UDP ports may refer to either local or remote UDP ports depending on the direction of the data transfer: from the point of view of the sending device the Source UDP port in a Datagram corresponds to the Local_UDP_port, but from the point of view of the receiving device the Source UDP port in a Datagram corresponds to the Remote_UDP_Port service parameter.

According to the UDP specification, filling the source UDP Port and Checksum fields with real data is optional. A zero value – all bits are equal to zero – of these fields indicates that in the given UDP Datagram the field is not used. However, in the DLMS/COSEM_on_IP profile, the source UDP Port field shall always be filled with the source UDP port number.

5.3.4 Reserved wrapper port numbers (wPorts)

Reserved wPort Numbers are specified in Table 1:

Table 1 – Reserved wrapper port numbers in the UDP-based DLMS/COSEM TL

Client side reserved addresses	
	Wrapper Port Number
No-station	0x0000
Client Management Process	0x0001
Public Client	0x0010
Open for client SAP assignment	0x02…0x0F
	0x11…0xFF
Server side reserved addresses	
	Wrapper Port Number
No-station	0x0000
Management Logical Device	0x0001
Reserved	0x0002..0x000F
Open for server SAP assignment	0x0010,,,0x007E
All-station (Broadcast)	0x007F

5.3.5 Protocol state machine

As the wrapper sublayer in this profile is state-less, for all other protocol related issues – protocol state machine, etc. – the governing rules are as they are specified in the Internet Standard STD 0006. The only supplementary rule is concerning discarding inappropriate messages: messages with an invalid destination wPort number – meaning that there is no DLMS/COSEM AE in the receiving device bound to this wPort number – shall be discarded by the wrapper sublayer.

6 The DLMS/COSEM connection-oriented, TCP-based transport layer

6.1 General

The DLMS/COSEM connection-oriented TL is based on the connection-oriented Internet transport protocol, called Transmission Control Protocol. TCP is an end-to-end reliable protocol. This reliability is ensured by a conceptual “virtual circuit”, using a method called PAR, Positive Acknowledgement with Retransmission. It provides acknowledged data delivery, error detection, data re-transmission after an acknowledgement time-out, etc. Therefore it deals with lost, delayed, duplicated or erroneous data packets. In addition, TCP offers an efficient flow control mechanism and full-duplex operation, too.

TCP, as a connection-oriented transfer protocol involves three phases: connection establishment, data exchange and connection release. Consequently, the DLMS/COSEM TCP-based TL provides OSI-style services to the service user(s) for all three phases:

- for the connection establishment phase, the TCP-CONNECT service is provided to the service user TCP connection manager process;
- for the data transfer phase, the TCP-DATA service is provided to the service user DLMS/COSEM AL;
- for the connection closing phase, the TCP-DISCONNECT service is provided to the service user TCP connection manager process;
- in addition, a TCP-ABORT service is provided to the service user DLMS/COSEM AL.

The DLMS/COSEM connection-oriented, TCP-based TL contains the same wrapper sublayer as the DLMS/COSEM UDP-based TL. In addition to transforming OSI-style services to and from TCP function calls, this wrapper provides additional addressing and length information.

The DLMS/COSEM connection-oriented, TCP-based TL is specified in terms of services and protocols. The conversion between OSI-style services and TCP function calls is presented in Annex A.

6.2 Service specification for the DLMS/COSEM TCP-based transport layer

6.2.1 General

The DLMS/COSEM connection-oriented, TCP-based TL provides the same set of services both at the client and at the server sides, as it is shown in Figure 6.

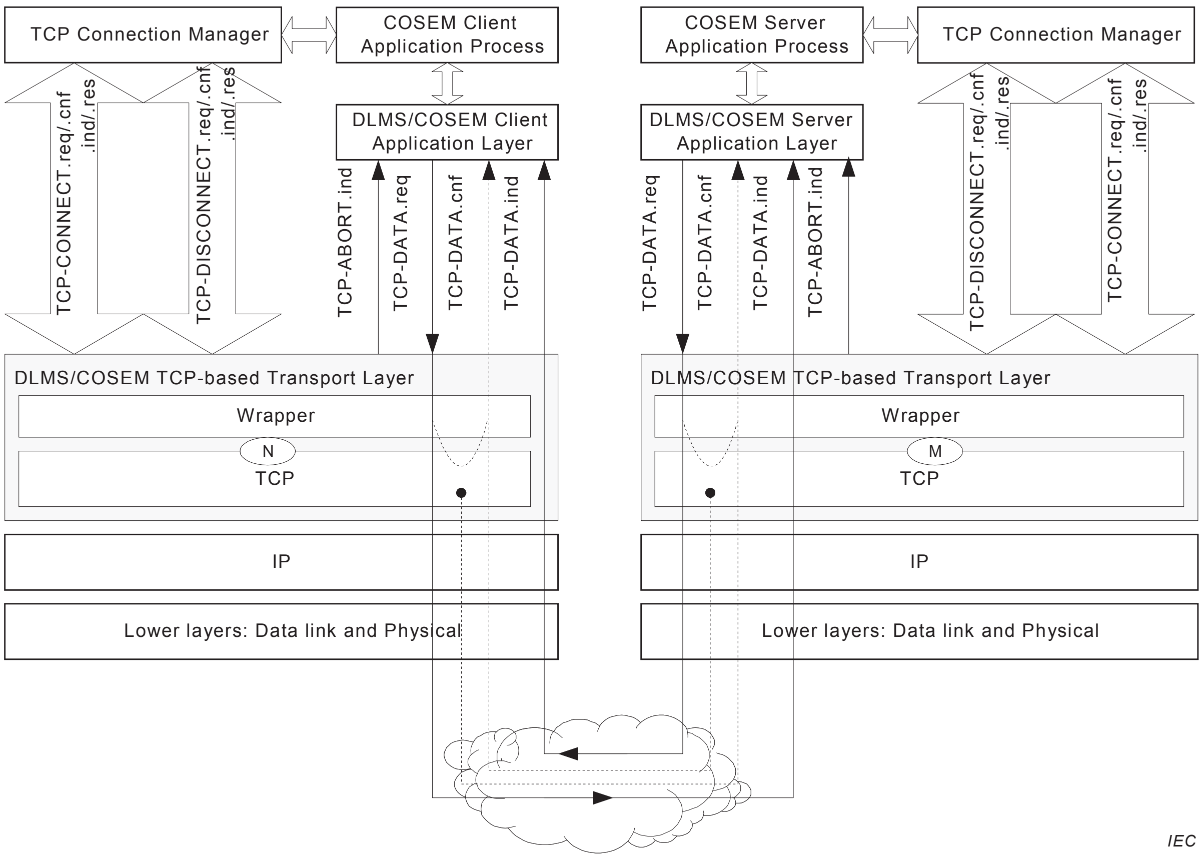


Figure 6 – Services of the DLMS/COSEM connection-oriented,TCP-based transport layer

In this communication profile, the full set of the service primitives of the TCP connection management services (TCP-CONNECT and TCP-DISCONNECT) is provided both at the client and at the server sides. This is to allow the server initiating and releasing a TCP connection, too.

NOTE Application association establishment is performed by the client AE.

The service user of the TCP connection management services is not the DLMS/COSEM AL, but the TCP connection manager process. The specification of this process is out of the scope of this companion specification; however, the DLMS/COSEM AL sets some requirements concerning this; see IEC 62056-9-7:2013, 9.1.

An additional COSEM-ABORT service is provided to indicate to the DLMS/COSEM AL the disruption or disconnection of the supporting TCP connection.

Like in the DLMS/COSEM UDP-based TL, the TCP-DATA.confirm service primitive is also optional. However, the TCP-DATA.request service can be confirmed either locally or remotely.

6.2.2 The TCP-CONNECT service

6.2.2.1 TCP-CONNECT.request

Function

This primitive is the service request primitive for the connection establishment service.

Semantics of the service primitive

The primitive shall provide parameters as follows:

```
TCP-CONNECT.request    (
                          Local_TCP_Port,
                          Remote_TCP_Port,
                          Local_IP_Address,
                          Remote_IP_Address
                       )
```

The Local_TCP_Port and Remote_TCP_Port parameters identify the local and remote TCP ports respectively. The Local_IP_Address and Remote_IP_Address parameters indicate the IP Address of the physical device requesting the TCP connection and of the target physical device, to which the TCP connection requested is to be established.

Use

The TCP-CONNECT.request primitive is invoked by the service user TCP connection manager process to establish a connection with the peer DLMS/COSEM TCP-based TL.

6.2.2.2 TCP-CONNECT.indication

Function

This primitive is the service indication primitive for the connection establishment service.

Semantics of the service primitive

The primitive shall provide parameters as follows:

```
TCP-CONNECT.indication (
                          Local_TCP_Port,
                          Remote_TCP_Port,
                          Local_IP_Address,
                          Remote_IP_Address
                       )
```

The Local_TCP_Port and Remote_TCP_Port parameters indicate the two TCP ports between which the requested TCP connection is to be established. The Local_IP_Address and Remote_IP_Address parameters indicate the IP addresses of the two devices participating in the TCP connection.

Use

The TCP-CONNECT.indication primitive is generated by the DLMS/COSEM TCP-based TL following the reception of a TCP packet, indicating to the TCP connection manager process that a remote device is requesting a new TCP connection.

6.2.2.3 TCP-CONNECT.response

Function

This primitive is the service response primitive for the connection establishment service.

Semantics of the service primitive

The primitive shall provide parameters as follows:

```
TCP-CONNECT.response   (
                          Local_TCP_Port,
                          Remote_TCP_Port,
```

Local_IP_Address,
Remote_IP_Address,
Result
)

The Local_TCP_Port and Remote_TCP_Port parameters indicate the two TCP ports between which the connection is being established. The Local_IP_Address and Remote_IP_Address parameters indicate the IP addresses of the two physical devices participating in the TCP connection.

The Result parameter indicates that the service user TCP connection manager has accepted the requested TCP connection. Its value is always SUCCESS.

Use

The TCP-CONNECT.response primitive is invoked by the TCP connection manager process to indicate to the DLMS/COSEM TCP-based TL whether the TCP connection requested previously has been accepted. The TCP connection manager cannot reject a requested connection.

6.2.2.4 TCP-CONNECT.confirm

Function

This primitive is the service confirm primitive for the connection establishment service.

Semantics of the service primitive

The primitive shall provide parameters as follows:

TCP-CONNECT.confirm (
Local_TCP_Port,
Remote_TCP_Port,
Local_IP_Address,
Remote_IP_Address,
Result,
Reason_of_Failure
)

The Local_TCP_Port and Remote_TCP_Port parameters indicate the two TCP ports between which the connection is being established. The Local_IP_Address and Remote_IP_Address parameters indicate the IP addresses of the two physical devices participating in this TCP connection.

The Result parameter indicates whether the requested TCP connection is established or not. Note, that this service primitive is normally the result of a remote confirmation – and as a TCP connection request cannot be rejected, the Result parameter shall always indicate SUCCESS.

However, the Result parameter may also indicate FAILURE, when it is locally confirmed. In this case the Reason_of_Failure parameter indicates the reason for the failure.

Use

The TCP-CONNECT.confirm primitive is generated by the DLMS/COSEM TCP-based TL to indicate to the service user TCP connection manager process the result of a TCP-CONNECT.request service invocation received previously.

6.2.3 The TCP-DISCONNECT service

6.2.3.1 TCP-DISCONNECT.request

Function

This primitive is the service request primitive for the connection termination service.

Semantics of the service primitive

The primitive shall provide parameters as follows:

```
TCP-DISCONNECT.request (
                        Local_TCP_Port,
                        Remote_TCP_Port,
                        Local_IP_Address,
                        Remote_IP_Address
                       )
```

The service parameters are the identifiers of the TCP connection to be released. The Local_TCP_Port and Local_IP_Address parameters designate the local TCP port and IP Address of the requesting device and application, the Remote_IP_Address and Remote_TCP_Port parameters refer to the remote device and application.

Use

The TCP-DISCONNECT.request primitive is invoked by the service user TCP connection manager process to request the disconnection of an existing TCP connection.

6.2.3.2 TCP-DISCONNECT.indication

Function

This primitive is the service indication primitive for the connection termination service.

Semantics of the service primitive

The primitive shall provide parameters as follows:

```
TCP-DISCONNECT.indication (
                           Local_TCP_Port,
                           Remote_TCP_Port,
                           Local_IP_Address,
                           Remote_IP_Address,
                           Reason
                          )
```

The Local_TCP_Port, Remote_TCP_Port, Local_IP_Address, Remote_IP_Address parameters identify the TCP connection, which is either requested to be released by the peer device, or has been aborted.

The Reason parameter indicates whether the service is invoked because of the peer device has requested a TCP disconnection (Reason == REMOTE_REQ), or it is locally originated by detecting a kind of event, which implies the disconnection of the TCP connection (Reason == ABORT).

NOTE The DLMS/COSEM Transport layer may give more detailed information about the reason for the ABORT via layer management services. However, those services are out of the scope of this International Standard.

Use

The TCP-DISCONNECT.indication primitive is generated by the DLMS/COSEM TCP-based TL to the service user TCP connection manager process to indicate that the peer entity has requested the disconnection of an existing TCP connection. The same primitive is used also to indicate if the TL detects a non-solicited disconnection of an existing TCP connection (for example, when the physical connection breaks down).

6.2.3.3 TCP-DISCONNECT.response

Function

This primitive is the service response primitive for the connection termination service.

Semantics of the service primitive

The primitive shall provide parameters as follows:

```
TCP-DISCONNECT.response    (
                                 Local_TCP_Port,
                                 Remote_TCP_Port,
                                 Local_IP_Address,
                                 Remote_IP_Address,
                                 Result
                           )
```

The Local_TCP_Port and Remote_TCP_Port parameters identify the two TCP ports between which the TCP connection has to be disconnected. The Local_IP_Address and Remote_IP_Address parameters indicate the IP addresses of the two physical devices participating in the TCP connection to be disconnected.

The Result parameter indicates that the service user TCP connection manager process has accepted to disconnect the TCP connection referenced. The value of this parameter is always SUCCESS.

Use

The TCP-DISCONNECT.response primitive is invoked by the TCP connection manager process to indicate to the DLMS/COSEM TCP-based TL whether the previously requested TCP disconnection is accepted. Note that the TCP connection manager process cannot reject the requested disconnection. This service primitive is invoked only if the corresponding TCP-DISCONNECT.indication service indicated a remotely initiated disconnection request (Reason == REMOTE_REQ).

6.2.3.4 TCP-DISCONNECT.confirm

Function

This primitive is the service confirm primitive for the connection termination service.

Semantics of the service primitive

The primitive shall provide parameters as follows:

```
TCP-DISCONNECT.confirm     (
                                 Local_TCP_Port,
                                 Remote_TCP_Port,
                                 Local_IP_Address,
                                 Remote_IP_Address,
                                 Result,
                                 Reason_of_Failure
```

)

The Local_TCP_Port and Remote_TCP_Port parameters identify the two TCP ports between which the TCP connection has to be disconnected. The Local_IP_Address and Remote_IP_Address parameters indicate the IP addresses of the two physical devices participating in the TCP connection to be disconnected.

The Result parameter indicates whether the disconnection of the TCP connection referenced has succeeded or not. Normally, this service primitive is invoked as the result of a remote confirmation, and as a TCP disconnection request cannot be rejected, the value of the Result parameter is always SUCCESS.

However, the Result parameter may also indicate FAILURE, when it is locally confirmed. In this case the Reason_of_Failure parameter indicates the reason of the failure.

Use

The TCP-DISCONNECT.confirm primitive is invoked by the DLMS/COSEM TCP-based TL to confirm to the service user TCP connection manager the result of a previous TCP-DISCONNECT.request service invocation.

6.2.4 The TCP-ABORT service

6.2.4.1 TCP-ABORT.indication

Function

This primitive is the service indication primitive for the connection termination service.

Semantics of the service primitive

The primitive shall provide parameters as follows:

```
TCP-ABORT.indication    (
                         Local_TCP_Port,
                         Remote_TCP_Port,
                         Local_IP_Address,
                         Remote_IP_Address,
                         Reason
                         )
```

The Local_TCP_Port and Remote_TCP_Port parameters identify the two TCP ports the connection between which has aborted. The Local_IP_Address and Remote_IP_Address parameters indicate the IP addresses of the two physical devices having participated in the TCP connection aborted.

The Reason parameter indicates the reason of the TCP abort. This parameter is optional.

Use

The TCP-ABORT.indication primitive is generated by the DLMS/COSEM TCP-based TL to indicate to the service user DLMS/COSEM AL a non-solicited disruption of the supporting TCP connection.

When this indication is received, the DLMS/COSEM AL shall release all AAs established using this TCP connection, and shall indicate this to COSEM AP using the COSEM-ABORT.indication service primitive. See also IEC 62056-5-3:2013, 6.4.

6.2.5 The TCP-DATA service

6.2.5.1 TCP-DATA.request

Function

This primitive is the service request primitive for the connection mode data transfer service.

Semantics of the service primitive

The primitive shall provide parameters as follows:

```
TCP-DATA.request  (
                      Local_wPort,
                      Remote_wPort,
                      Local_TCP_Port,
                      Remote_TCP_Port,
                      Local_IP_Address,
                      Remote_IP_Address,
                      Data_Length,
                      Data
                  )
```

The Local_wPort, Local_TCP_Port and Local_IP_Address parameters indicate wrapper Port number, TCP Port number and IP Address parameters of the device / DLMS/COSEM AE requesting to send the Data. The Remote_wPort, Remote_TCP_Port and Remote_IP_Address parameters indicate the wrapper Port number, TCP Port number and IP Address parameters belonging to the device / DLMS/COSEM AE to which the Data is to be transmitted.

The Data_Length parameter indicates the length of the Data parameter in bytes.

The Data parameter contains the xDLMS APDU to be transferred to the peer AL.

Use

The TCP-DATA.request primitive is invoked by either the client or the server DLMS/COSEM AL to request sending an APDU to a single peer application.

The reception of this primitive shall cause the wrapper sublayer to pre-fix the wrapper-specific fields (Local_wPort, Remote_wPort and the Data_Length) to the xDLMS APDU received, and then to call the SEND() function of the TCP sublayer with the properly formed WPDU, see 5.3.2, as DATA. The TCP sublayer shall transmit the WPDU to the peer TCP sublayer as described in STD 0007.

6.2.5.2 TCP-DATA.indication

Function

This primitive is the service indication primitive for the connection mode data transfer service.

Semantics of the service primitive

The primitive shall provide parameters as follows:

```
TCP-DATA.indication  (
                         Local_wPort,
                         Remote_wPort,
                         Local_TCP_Port,
                         Remote_TCP_Port,
                         Local_IP_Address,
                         Remote_IP_Address,
                         Data_Length,
                         Data
                     )
```

The Local_wPort, Local_TCP_Port and Local_IP_Address parameters indicate wrapper Port number, TCP Port number and IP Address parameters belonging to the device / DLMS/COSEM AE receiving the Data. The Remote_wPort, Remote_TCP_Port and Remote_IP_Address parameters indicate the wrapper Port number, TCP Port number and IP Address parameters belonging to the device / DLMS/COSEM AE which has sent the Data.

The Data_Length parameter indicates the length of the Data parameter in bytes.

The Data parameter contains the xDLMS APDU received from the peer AL.

Use

The TCP-DATA.indication primitive is generated by the DLMS/COSEM TL to indicate to the service user DLMS/COSEM AL that an xDLMS APDU has been received from a remote device. It is generated following the reception of a complete APDU (in one or more TCP packets) by the DLMS/COSEM TCP-based TL, if both the Local_TCP_Port and Local_wPort parameters in the TCP packet(s) carrying the APDU contain valid port numbers, meaning that there is a DLMS/COSEM AE in the receiving device bound to the given port numbers. Otherwise, the message received shall simply be discarded.

6.2.5.3 TCP-DATA.confirm

Function

This primitive is the optional service confirm primitive for the connection mode data transfer service.

Semantics of the service primitive

The primitive shall provide parameters as follows:

TCP-DATA.confirm (
Local_wPort,
Remote_wPort,
Local_TCP_Port,
Remote_TCP_Port,
Local_IP_Address,
Remote_IP_Address,
Confirmation_Type,
Result
)

The Local_wPort, Remote_wPort, Local_TCP_Port, Remote_TCP_Port, Local_IP_Address and Remote_IP_Address parameters carry the same values as the corresponding TCP-DATA.request service being confirmed.

The Confirmation_Type parameter indicates whether the confirmation service is a LOCAL or a REMOTE confirmation.

The value of the Result parameter indicates the result of the previous TCP-DATA.request service. Its value is either OK or NOK, but the meaning of this depends on the implementation of the .confirm primitive. See 6.3.5.4.

Use

The TCP-DATA.confirm primitive is optional. If implemented, it is generated by the DLMS/COSEM TL to confirm to the service user DLMS/COSEM AL the result of the execution of the previous .request primitive.

6.3 Protocol specification for the DLMS/COSEM TCP-based transport layer

6.3.1 General

As it is shown in Figure 2, the DLMS/COSEM CO, TCP-based TL includes the Internet standard TCP layer as specified in STD 0007, and the DLMS/COSEM-specific wrapper sublayer.

In the TCP-based TL the wrapper sublayer is more complex than in the UDP-based TL. On the one hand – similarly to the UDP-based TL – its main role is also to ensure source and destination DLMS/COSEM AE identification using the wPort numbers, and to convert OSI-style TCP-DATA service primitives to and from the SEND() and RECEIVE() interface functions provided by the standard TCP. On the other hand, the wrapper sublayer in the TCP-based TL has also the task to help the service user DLMS/COSEM ALs to exchange complete APDUs.

TCP is a “streaming” protocol meaning that it does not preserve data boundaries. Without entering into the details here (see more in Clause A.4) this means, that the SEND() and RECEIVE() function calls of the TCP sublayer return with success even if the number of the bytes sent / received actually is less than the number of bytes requested to be sent / received. It is the responsibility of the wrapper sublayer to know how much data had to be sent / received, to keep track how much has been actually sent / received, and repeat the operation until the complete APDU is transmitted.

Consequently, the wrapper sublayer in the TCP-based DLMS/COSEM TL is not a state-less entity: it is doing the above described track-keeping – re-trying procedure in order to make the “streaming” nature of the TCP transparent to the service user DLMS/COSEM AL.

6.3.2 The wrapper protocol data unit (WPDU)

The wrapper protocol data unit is as it is specified in 5.3.2.

6.3.3 The DLMS/COSEM TCP-based transport layer protocol data unit

WPDUs are transmitted in one or more TCP packets. The TCP Packet is specified in STD 0007 and encapsulates a part of the WPDU in its Data Field, as it is shown in Figure 7. The reason for having only a part of the WPDU in a TCP packet is the “streaming” nature of the TCP already mentioned.

0	4	8	10	16	24 31
Source TCP Port				Destination TCP Port	
Sequence Number					
Acknowledgement Number					
Data Offset	Reserved		Flags	Window	
Checksum				Urgent Pointer	
TCP Options					Padding
Data (part of the WPDU)					
…Data (part of the WPDU)…					

IEC

Figure 7 – The TCP packet format

From the external point of view the DLMS/COSEM TCP-based TL PDU is an ordinary TCP packet: any DLMS/COSEM specific element, including the wrapper-specific header in the first TCP packet, is inside the packet’s Data field.

The source and destination TCP ports may refer to either local or remote TCP ports, depending on the direction of the data transfer (i.e. from the point of view of the Sender device the source TCP port in a TCP Packet corresponds to the Local_TCP_port, but from the point of view of the Receiver device, the source TCP port of a Datagram corresponds to the Remote_TCP_Port service parameter).

6.3.4 Reserved wrapper port numbers

Reserved wPort Numbers are specified in Table 1.

6.3.5 Definition of the procedures

6.3.5.1 TCP connection

Establishment of a TCP connection is initiated by the TCP-CONNECT.request service invocation. Although this service – as all DLMS/COSEM TL services – is provided to the service user entity by the wrapper sublayer, the TCP connection is established between the two (local and remote) TCP sublayers. The role of the wrapper in this procedure is just to convert the TCP-CONNECT service primitives (.request, .indication, .response and .confirm) to and from TCP function calls.

From the service user point of view, only the TCP-CONNECT service primitives are visible: according to this, the TCP connection establishment takes place as it is shown in Figure 8.

The TCP connection is established using a three-way handshake mechanism, as described in STD 0007. This requires three message exchanges as shown above and guarantees that both sides know that the other side is ready to transmit and also that the two sides are synchronized: the initial sequence numbers are agreed upon.

Both the client and server side TCP connection manager processes are allowed to initiate the TCP connection. To establish the connection, one of them plays the role of the initiator, and the other that of the responder.

In order to be able to respond, the responder has to perform a 'passive' opening before receiving the first, SYN packet. To do this, it has to contact the local operating system (OS) to indicate, that it is ready to accept incoming connection requests. As the result of this contact, the OS assigns a TCP port number to that end-point of the connection and reserves the resources required for a future connection – but no message is sent out.

NOTE In the case of the DLMS/COSEM Transport layer, the implementation forces the OS to assign the requested TCP / UDP port number to the local end point of the connection.

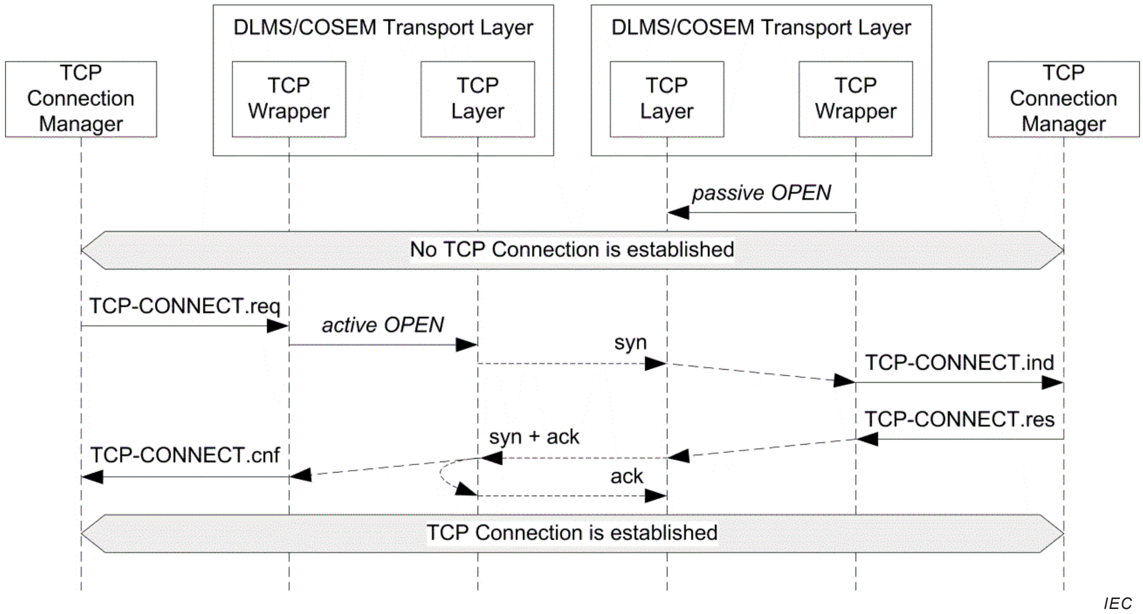


Figure 8 – TCP connection establishment

In the case of the DLMS/COSEM TCP-based TL, the wrapper sublayer initiates this passive opening autonomously during system initialisation. In other words, as this passive opening is the responsibility of the wrapper sublayer, no service is provided to an external entity to initiate the passive opening.

As both the client and the server side TCP connection manager processes are allowed to play the role of the "Responder" application, the TLs on both sides shall perform a passive opening during the system initialisation.

More details about TCP connection establishment are provided in Clause A.1.

6.3.5.2 TCP disconnection

The TCP is disconnected using the TCP-DISCONNECT service, as shown in Figure 9.

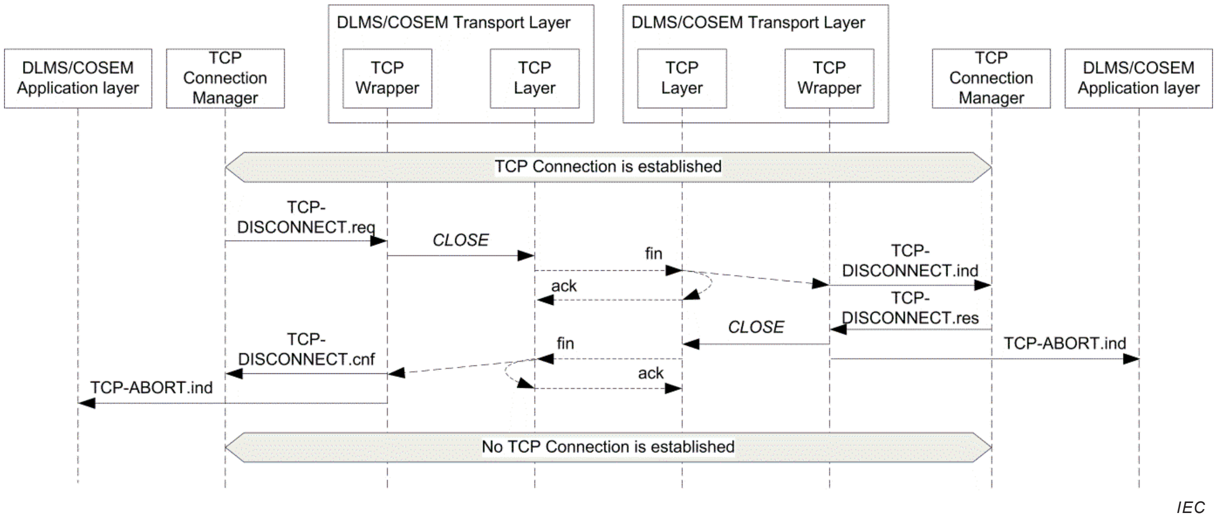


Figure 9 – TCP disconnection

The procedure can be initiated either by the client or the server side TCP connection manager process, by invoking the TCP-DISCONNECT.request primitive. This request is transformed by the “wrapper” to a CLOSE () function call to the TCP interface.

The TCP sends a fin segment, which is acknowledged by the peer TCP.

NOTE TCP uses an improved 3-way handshake to release a connection, to ensure that possible duplication and delay – introduced by the non-reliable IP layer – do not pose problems. More about this procedure can be found in STD 0007.

At the same time, through the wrapper, the TCP-DISCONNECT.indication primitive is generated, informing the user TCP connection manager that the connection is closing. The connection manager – in order to gracefully release the connection – responds with a TCP-DISCONNECT.response primitive. The TCP wrapper calls the CLOSE function and the TCP sends out its fin segment. At the same time, the TCP wrapper indicates the closing of the TCP connection to the DLMS/COSEM AL using the TCP-ABORT.indication primitive.

On the requesting side, the TCP sends an acknowledgement and upon the reception of this by the peer the TCP connection is deleted. At the same time, the wrapper generates the TCP-DISCONNECT.confirm primitive informing the connection manager process that the disconnection request has been accepted. Similarly to the peer, the TCP disconnection is also indicated to the DLMS/COSEM AL with the help of the COSEM-ABORT.indication primitive.

More details about TCP disconnection are provided in Clause A.2.

6.3.5.3 TCP connection abort

The DLMS/COSEM TCP-based TL indicates the disruption or disconnection of the supporting TCP connection to the DLMS/COSEM AL with the help of the TCP-ABORT.indication primitive. Note that this is the only TCP connection management service provided to the DLMS/COSEM AL.

The service is invoked either when the TCP connection is disconnected by the TCP connection manager process – the case of graceful disconnection – or when the TCP disconnection occurs in a non-solicited manner, for example the TCP sublayer is detecting a non-resolvable error or the physical connection is shut down.

The purpose of this service is to inform the DLMS/COSEM AL about the disruption of the TCP connection, so that it could release all existing AAs.

6.3.5.4 Data transfer using the TCP-DATA service

Once the TCP connection is established, reliable data transfer can be performed via this connection. Although providing this reliable data transfer is a quite complex operation involving reliability mechanisms such as Positive Acknowledgement with Retransmission (PAR) or flow control with sliding windows – provided by TCP and specified in STD 0007 – the DLMS/COSEM TCP-based TL layer provides only data transfer service, the TCP-DATA service, as shown in Figure 10.

The use of the TCP-DATA service is the same both on the client and at the server side.

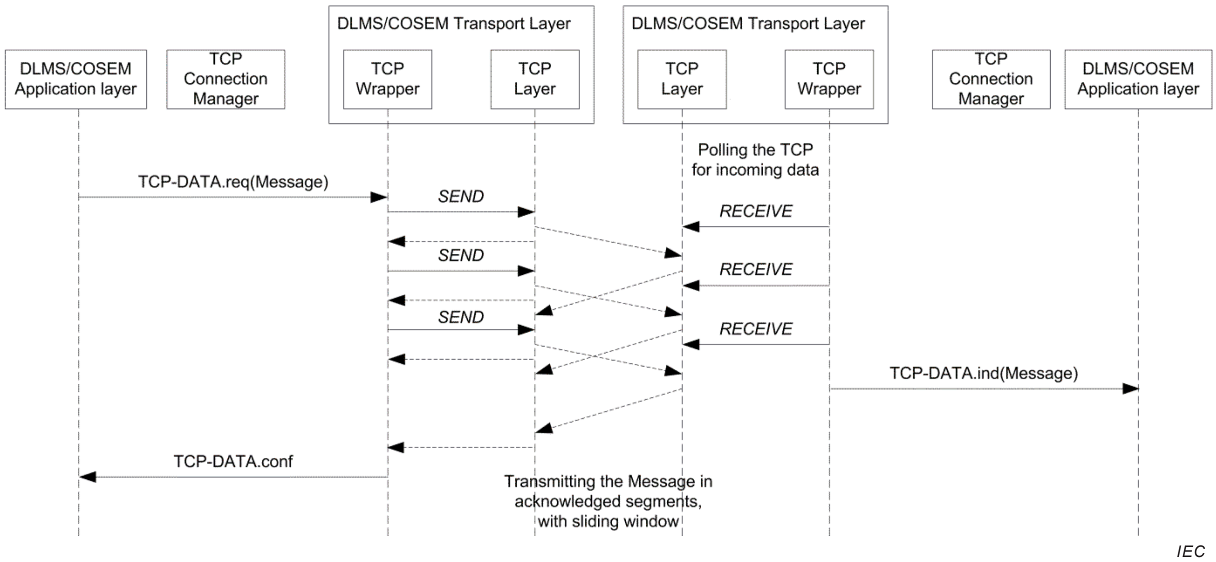


Figure 10 – Data transfer using the DLMS/COSEM TCP-based transport layer

The optional TCP-DATA.confirm primitive indicates the result of the TCP-DATA.request primitive invoked previously, which is either OK or NOK. However, the meaning of this result is implementation dependent. When the .confirm primitive is implemented as a local confirmation, the result indicates whether the DLMS/COSEM TL was able to buffer for sending or to send out the APDU or not. When it is implemented as a remote confirmation, the result indicates whether the APDU has been successfully delivered to the destination or not.

As shown in Figure 10, the message (a WPDU) may be transported (sent / received) in more than one TCP packet. It is because TCP sends data as a stream of octets, without preserving data boundaries. It is the responsibility of the wrapper sublayer to hide this property of the TCP sublayer from the service user DLMS/COSEM AL. The sender side wrapper keeps track about the amount of data sent with one SEND() function call and repeats the operation until the whole WPDU is sent. The receiver side wrapper continues to receive incoming TCP packets until a complete WPDU is received. For more details, see Clause A.4.

6.3.5.5 High-level state transition diagram of the wrapper sublayer

The high level state-diagram of the wrapper sublayer is shown in Figure 11.

In both macro-states – No TCP Connection and TCP Connected – the wrapper keeps polling the TCP layer for its connection status, and transits into the other macro-state if the status has changed.

The wrapper enters always into the IDLE sub-state of the TCP Connected state, and transits to the composite SEND/RECEIVE state either on a TCP-DATA.request or on the reception of a TCP packet. In this state, the wrapper sends and/or receives WPDUs, as described in Annex A.

NOTE TCP on the top of a full-duplex lower layer protocol stack can simultaneously send and receive.

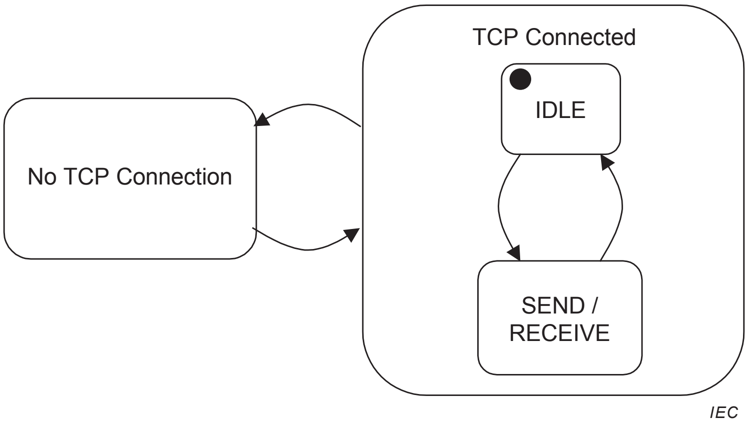


Figure 11 – High-level state transition diagram for the wrapper sublayer

Annex A
(informative)

Converting OSI-style TL services to and from RFC-style TCP function calls

A.1 Transport layer and TCP connection establishment

As specified in STD 0007, a TCP connection is established by calling the OPEN function. This function can be called in active or passive manner.

According to the TCP connection state diagram (see Figure A.1) a passive OPEN takes the caller device to the LISTEN state, waiting for a connection request from any remote TCP and port.

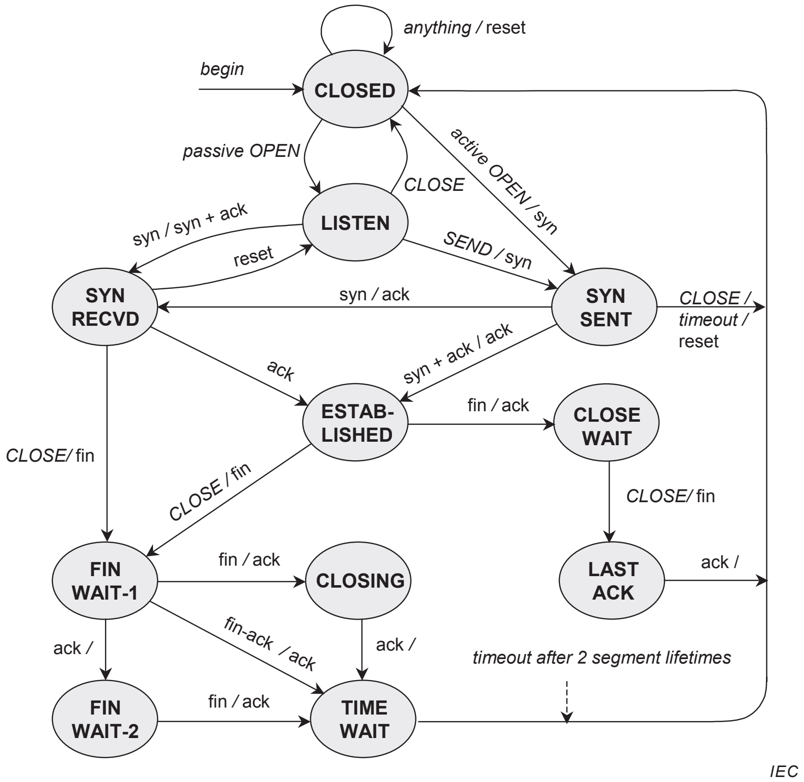
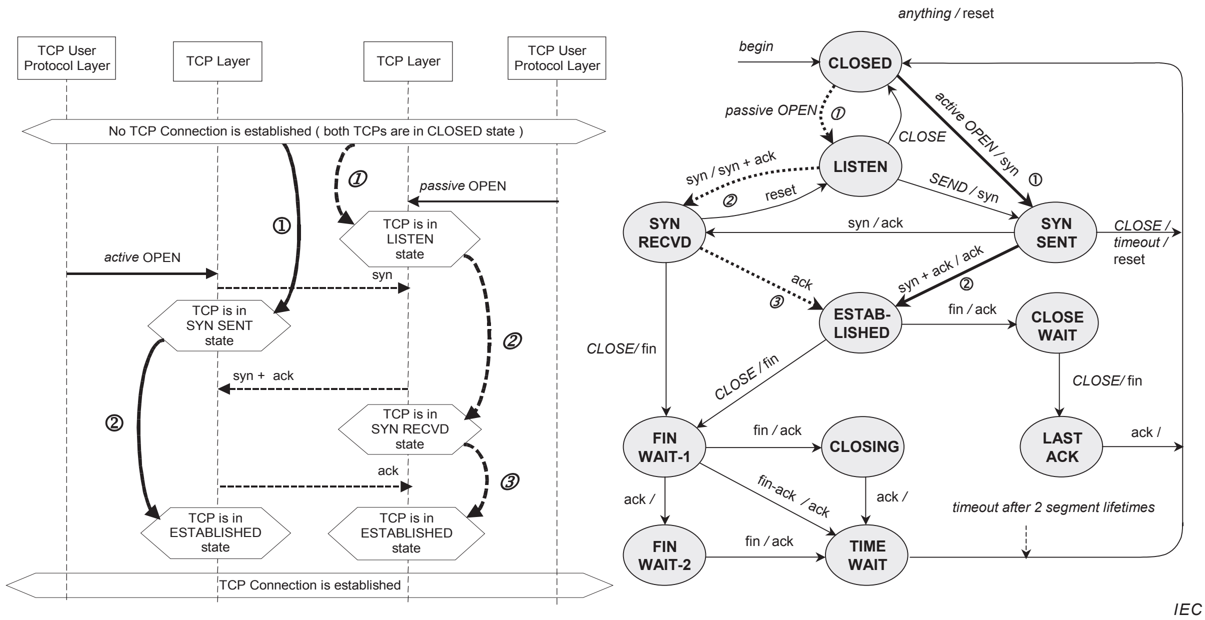


Figure A.1 – TCP connection state diagram

An active OPEN call makes the TCP to establish the connection to a remote TCP.

The establishment of a TCP Connection is performed by using the so-called “Three-way handshake” procedure. This is initiated by one TCP calling an active OPEN and responded by another TCP, the one which has already been called a passive OPEN and consequently is in the LISTEN state.

The message sequence and the state transitions corresponding to that message exchange for this “three-way handshake” procedure are shown in Figure A.2.



NOTE In the case of the DLMS/COSEM transport layer, the TCP user protocol layer is the wrapper sublayer.

Figure A.2 – MSC and state transitions for establishing a transport layer and TCP connection

This process, consisting of three messages, establishes the TCP connection and "synchronizes" the initial sequence numbers at both sides. This mechanism has been carefully designed to guarantee, that both sides are ready to transmit data and know that the other side is ready to transmit as well. Note that the procedure also works if two TCPs simultaneously initiate the procedure.

NOTE Sequence numbers are part of the TCP packet, and are fundamental to reliable data transfer. For more details about sequence numbers (or other TCP related issues), refer to STD 0007.

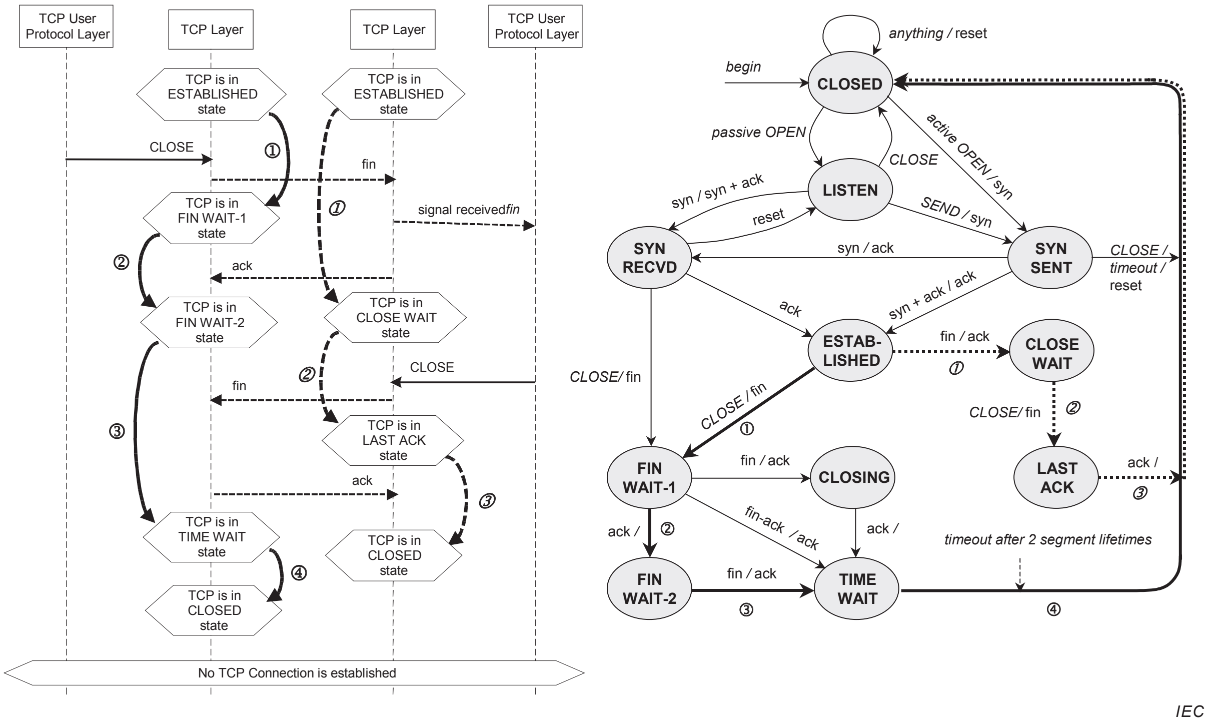
A.2 Closing a transport layer and a TCP connection

Closing a TCP connection is done by calling the CLOSE function, generally when there is no more data to be sent.

Upon the invocation of the TCP-DISCONNECT.request service primitive by the TCP connection manager process, the wrapper sublayer invokes the CLOSE function of the TCP sublayer.

However, as the TCP connection is full duplex, the other side may still have data to send. Therefore, after calling the CLOSE function, the TCP-based transport later may continue to receive data and send it to the DLMS/COSEM AL, until it is told that the other side has CLOSED, too. At this point it generates the COSEM-ABORT.indication primitive, and all AAs are released.

The message sequence chart and the state transitions corresponding to a successful TCP connection release are shown in Figure A.3.



NOTE In the case of the DLMS/COSEM TL, the TCP user protocol layer is the wrapper sublayer.

Figure A.3 – MSC and state transitions for closing a transport layer and TCP connection

A.3 TCP connection abort

STD 0007 does not specify a standard function to indicate an unexpected abort at TCP level. However, it can be detected by the TCP user entity by polling the status of the TCP with the STATUS() function, as shown in Figure A.4

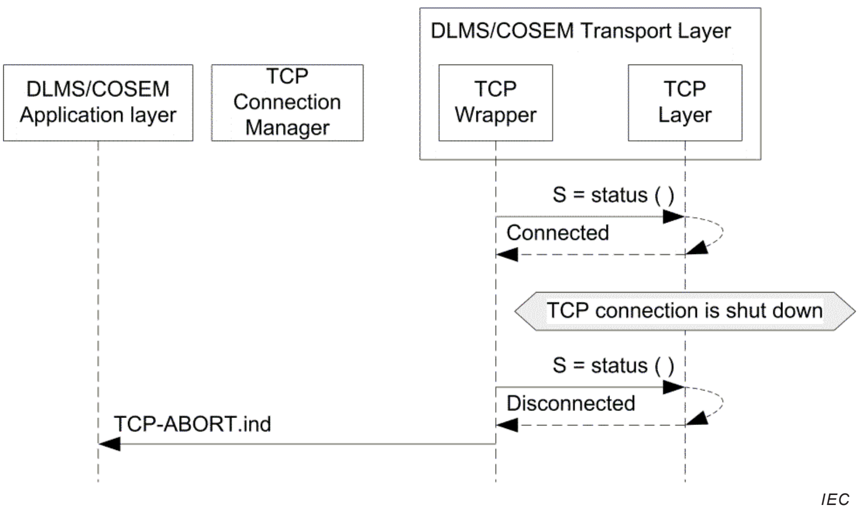


Figure A.4 – Polling the TCP sublayer for TCP abort indication

A.4 Data transfer using the TCP-DATA service

To send an APDU to the peer, the DLMS/COSEM AL simply invokes the TCP-DATA.request primitive of the DLMS/COSEM TCP-based TL. Also, when a complete APDU is received, this is indicated to the DLMS/COSEM AL with the help of the TCP-DATA.indication primitive. Thus, for the AL the TL behaves as if it would transport the whole APDU in one piece.

However, as TCP is a streaming protocol, not preserving data boundaries, as described in 6.3.1, nothing ensures that an APDU is actually transmitted in one TCP packet. As already mentioned in 6.3.5.4, in the DLMS/COSEM TCP-based TL it is the responsibility of the wrapper sublayer to “hide” the streaming nature of the TCP sublayer.

The following example illustrates how the wrapper sublayer accomplishes this task. Let us suppose, that an AL entity wants to send an APDU containing 992 bytes via the DLMS/COSEM TCP-based TL.

NOTE Both the client and server side ALs can be either sender or receiver.

It invokes the TCP-DATA.request service with this APDU as the DATA service parameter as shown in Figure A.5.

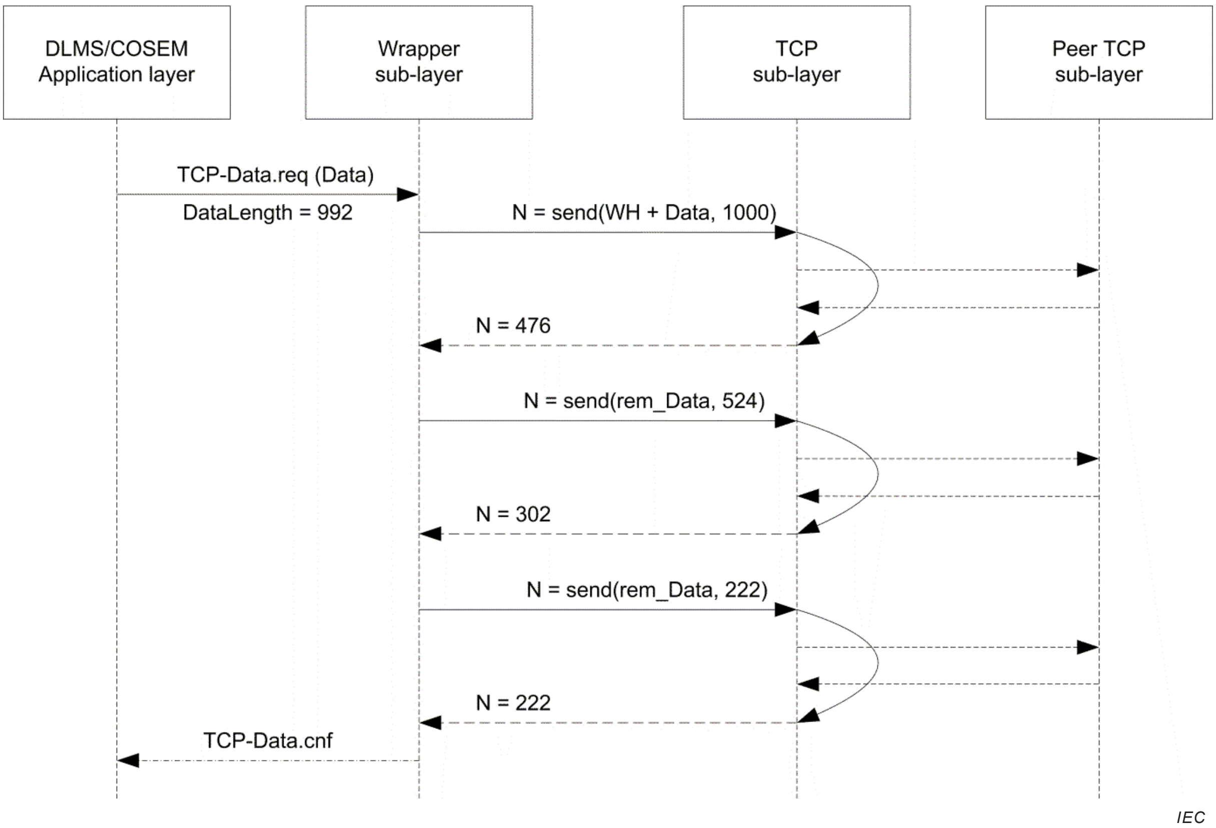


Figure A.5 – Sending an APDU in three TCP packets

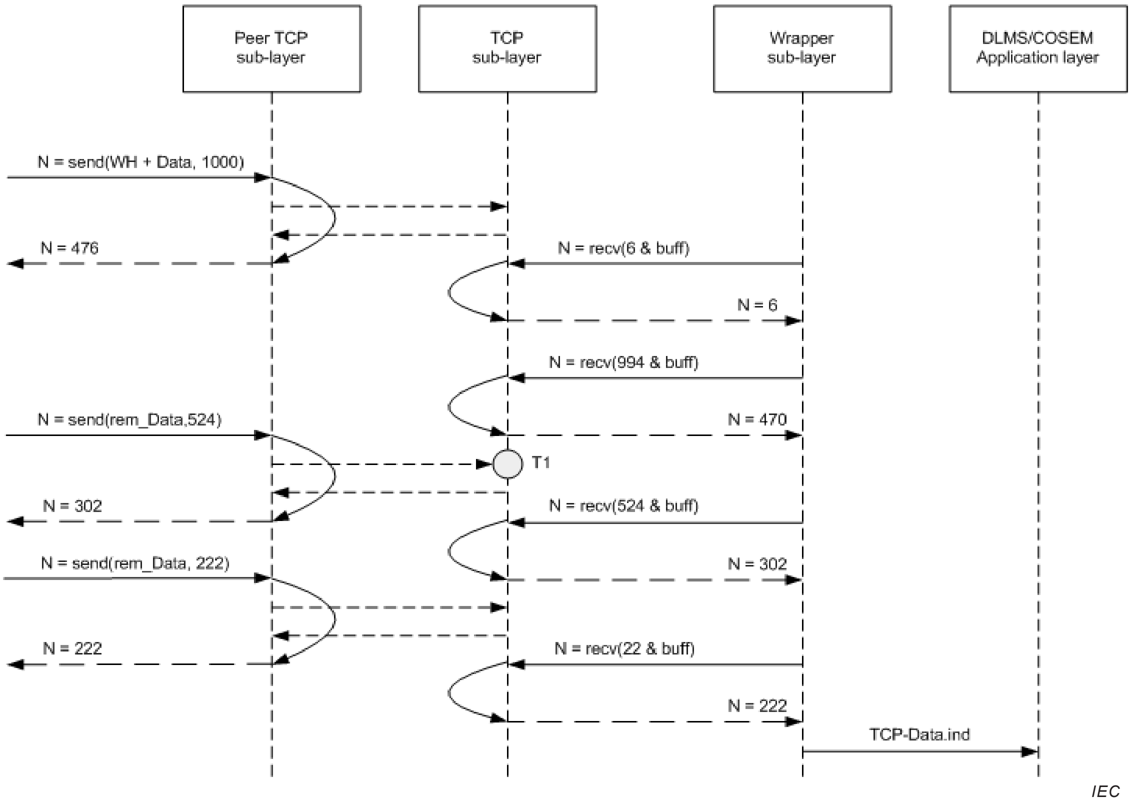
Upon the reception of this service invocation, the wrapper sublayer constructs the WPDU: it pre-fixes the APDU with the wrapper header (WH), including the local and remote wPort numbers and the APDU length. It calls then the SEND() function of the TCP sublayer, requesting to send the WPDU, which is now 1 000 bytes long: 8 bytes of wrapper header plus 992 bytes of APDU.

The SEND() function returns with the number of bytes sent or an error (a negative value). Let us suppose, that no error occurs, and the SEND() function successfully returns with the value

476. This number is the number of bytes sent. This also illustrates the meaning of the “streaming” nature of the TCP: in fact, the SEND() function returns with success even if the number of bytes sent is less than the number of bytes requested to be sent. From the value returned, the wrapper knows that not the whole WPDU has been sent. It calls the SEND() function again, with the remaining part of the WPDU, and so on, until the complete WPDU is sent.

As already mentioned in 6.3.5.4, depending on the implementation, the successful return of the SEND() function may even not mean that something has been really sent to the network. It may mean only that the protocol implementation took and buffered the data. It may happen that the protocol implementation delays the transmission to comply with protocol conventions or network traffic related algorithms.

On the receiving side, it is also the responsibility of the wrapper sublayer to assemble the complete APDU before invoking the TCP-DATA.indication primitive. This is possible by using the length bytes of the WPDU header. The wrapper repeats RECEIVE() calls until the number of bytes, indicated in the WPDU header is received. This is shown in Figure A.6.



NOTE 1 As calling the RECEIVE() function is asynchronous with regard to the TCP communications, it is perfectly possible, that the receiver calls the RECEIVE() function at a moment, when the reception of a TCP packet is in progress (T1 on the Figure above) or even if when no characters have been received since the last RECEIVE() call. It does not lead to erroneous reception: it increases only the number of necessary RECEIVE() function calls to get the complete message.

NOTE 2 It is also possible that one or more SEND() calls result in sending more than one TCP packet. It does not lead to erroneous reception either: sooner or later, the receiver gets the whole message.

Figure A.6 – Receiving the message in several packets

All these SEND() and RECEIVE() calls are internal to the DLMS/COSEM TL. The service user DLMS/COSEM AL simply uses the TCP-DATA services, and observes a reliable data transfer service preserving the data boundaries of the APDUs.

INDEX

Bibliography

RFC 0768, *User Datagram Protocol (Also: IETF STD0006).* August 1980. Available from: http://www.ietf.org/rfc/rfc768.txt

RFC 0791, *Internet Protocol (Also: IETF STD 0005).* Edited by J. Postel, September 1981. Available from http://www.ietf.org/rfc/rfc791.txt

RFC 0792, *Internet Control Message Protocol (Also: IETF STD 0005),* Edited by J. Postel, September 1981. Available from http://www.ietf.org/rfc/rfc792.txt

RFC 0793, *Transmission Control Protocol (Also: IETF STD 0005).* Edited by J. Postel, September 1981. Available from http://www.ietf.org/rfc/rfc793.txt

RFC 0919, *Broadcasting Internet Datagrams (Also: IETF STD 0005).* Edited by J. Mogul, October 1984. Available from http://www.ietf.org/rfc/rfc919.txt

RFC 0922, *Broadcasting Internet datagrams in the presence of subnets (Also: IETF STD 0005).* Edited by J. Mogul, October 1984. Available from http://www.ietf.org/rfc/rfc922.txt

RFC 0950, *Internet Standard Subnetting Procedure (Also: IETF STD 0005).* Edited by J. Mogul, J. Postel August 1985. Available from http://www.ietf.org/rfc/rfc950.txt

RFC 1112, *Host extensions for IP multicasting (Also: IETF STD 0005).* Edited by S. Deering August 1989. Available from http://www.ietf.org/rfc/rfc1112.txt

RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification.* Edited by S. Deering and R. Hinden. December 1998. Available from: http://www.ietf.org/rfc/rfc2460.txt

RFC 3513, *Internet Protocol Version 6 (IPv6) Addressing Architecture. Edited by R. Hinden S. Deering, April 2003.* Available from: http://www.ietf.org/rfc/rfc3513.txt

This page deliberately left blank

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Copyright in BSI publications

All the content in BSI publications, including British Standards, is the property of and copyrighted by BSI or some person or entity that owns copyright in the information used (such as the international standardization bodies) and has formally licensed such information to BSI for commercial publication and use.

Save for the provisions below, you may not transfer, share or disseminate any portion of the standard to any other person. You may not adapt, distribute, commercially exploit, or publicly display the standard or any portion thereof in any manner whatsoever without BSI's prior written consent.

Storing and using standards

Standards purchased in soft copy format:

- A British Standard purchased in soft copy format is licensed to a sole named user for personal or internal company use only.
- The standard may be stored on more than 1 device provided that it is accessible by the sole named user only and that only 1 copy is accessed at any one time.
- A single paper copy may be printed for personal or internal company use only.

Standards purchased in hard copy format:

- A British Standard purchased in hard copy format is for personal or internal company use only.
- It may not be further reproduced – in any format – to create an additional copy. This includes scanning of the document.

If you need more than 1 copy of the document, or if you wish to share the document on an internal network, you can save money by choosing a subscription product (see 'Subscriptions').

Reproducing extracts

For permission to reproduce content from BSI publications contact the BSI Copyright & Licensing team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email subscriptions@bsigroup.com.

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Useful Contacts

Customer Services
Tel: +44 345 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 345 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

bsi.