

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Electricity metering data exchange – The DLMS/COSEM suite –
Part 4-7: DLMS/COSEM transport layer for IP networks**

**Échange des données de comptage de l'électricité – La suite DLMS/COSEM –
Partie 4-7: Couche transport DLMS/COSEM pour réseaux IP**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2015 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

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

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 15 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

More than 60 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Catalogue IEC - webstore.iec.ch/catalogue

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

Recherche de publications IEC - www.iec.ch/searchpub

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 15 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

Plus de 60 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Electricity metering data exchange – The DLMS/COSEM suite –
Part 4-7: DLMS/COSEM transport layer for IP networks**

**Échange des données de comptage de l'électricité – La suite DLMS/COSEM –
Partie 4-7: Couche transport DLMS/COSEM pour réseaux IP**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 17.220; 35.110; 91.140.50

ISBN 978-2-8322-2609-4

Warning! Make sure that you obtained this publication from an authorized distributor. Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.
--

CONTENTS

FOREWORD.....	4
INTRODUCTION.....	6
1 Scope.....	7
2 Normative references	7
3 Terms, definitions and abbreviations	8
3.1 Terms and definitions.....	8
3.2 Abbreviations.....	8
4 Overview	9
5 The DLMS/COSEM connection-less, UDP-based transport layer	10
5.1 General.....	10
5.2 Service specification for the DLMS/COSEM UDP-based transport layer	11
5.2.1 General	11
5.2.2 The UDP-DATA service	12
5.3 Protocol specification for the DLMS/COSEM UDP-based transport layer.....	14
5.3.1 General	14
5.3.2 The wrapper protocol data unit (WPDU).....	14
5.3.3 The DLMS/COSEM UDP-based transport layer protocol data unit.....	15
5.3.4 Reserved wrapper port numbers (wPorts).....	16
5.3.5 Protocol state machine	16
6 The DLMS/COSEM connection-oriented, TCP-based transport layer	16
6.1 General.....	16
6.2 Service specification for the DLMS/COSEM TCP-based transport layer	17
6.2.1 General	17
6.2.2 The TCP-CONNECT service.....	18
6.2.3 The TCP-DISCONNECT service	21
6.2.4 The TCP-ABORT service	23
6.2.5 The TCP-DATA service.....	24
6.3 Protocol specification for the DLMS/COSEM TCP-based transport layer	26
6.3.1 General	26
6.3.2 The wrapper protocol data unit (WPDU).....	26
6.3.3 The DLMS/COSEM TCP-based transport layer protocol data unit	27
6.3.4 Reserved wrapper port numbers	27
6.3.5 Definition of the procedures.....	27
Annex A (informative) Converting OSI-style TL services to and from RFC-style TCP function calls.....	32
A.1 Transport layer and TCP connection establishment.....	32
A.2 Closing a transport layer and a TCP connection.....	33
A.3 TCP connection abort	34
A.4 Data transfer using the TCP-DATA service	35
INDEX	37
Bibliography.....	38
Figure 1 – DLMS/COSEM as a standard Internet application protocol	9
Figure 2 – Transport layers of the DLMS/COSEM_on_IP profile.....	10
Figure 3 – Services of the DLMS/COSEM connection-less, UDP-based transport layer.....	12

Figure 4 – The wrapper protocol data unit (WPDU)	15
Figure 5 – The DLMS/COSEM connection-less, UDP-based transport layer PDU (UDP-PDU)	15
Figure 6 – Services of the DLMS/COSEM connection-oriented, TCP-based transport layer	18
Figure 7 – The TCP packet format	27
Figure 8 – TCP connection establishment	28
Figure 9 – TCP disconnection	29
Figure 10 – Data transfer using the DLMS/COSEM TCP-based transport layer	30
Figure 11 – High-level state transition diagram for the wrapper sublayer.....	31
Figure A.1 – TCP connection state diagram	32
Figure A.2 – MSC and state transitions for establishing a transport layer and TCP connection	33
Figure A.3 – MSC and state transitions for closing a transport layer and TCP connection	34
Figure A.4 – Polling the TCP sublayer for TCP abort indication.....	34
Figure A.5 – Sending an APDU in three TCP packets	35
Figure A.6 – Receiving the message in several packets.....	36
Table 1 – Reserved wrapper port numbers in the UDP-based DLMS/COSEM TL	16

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

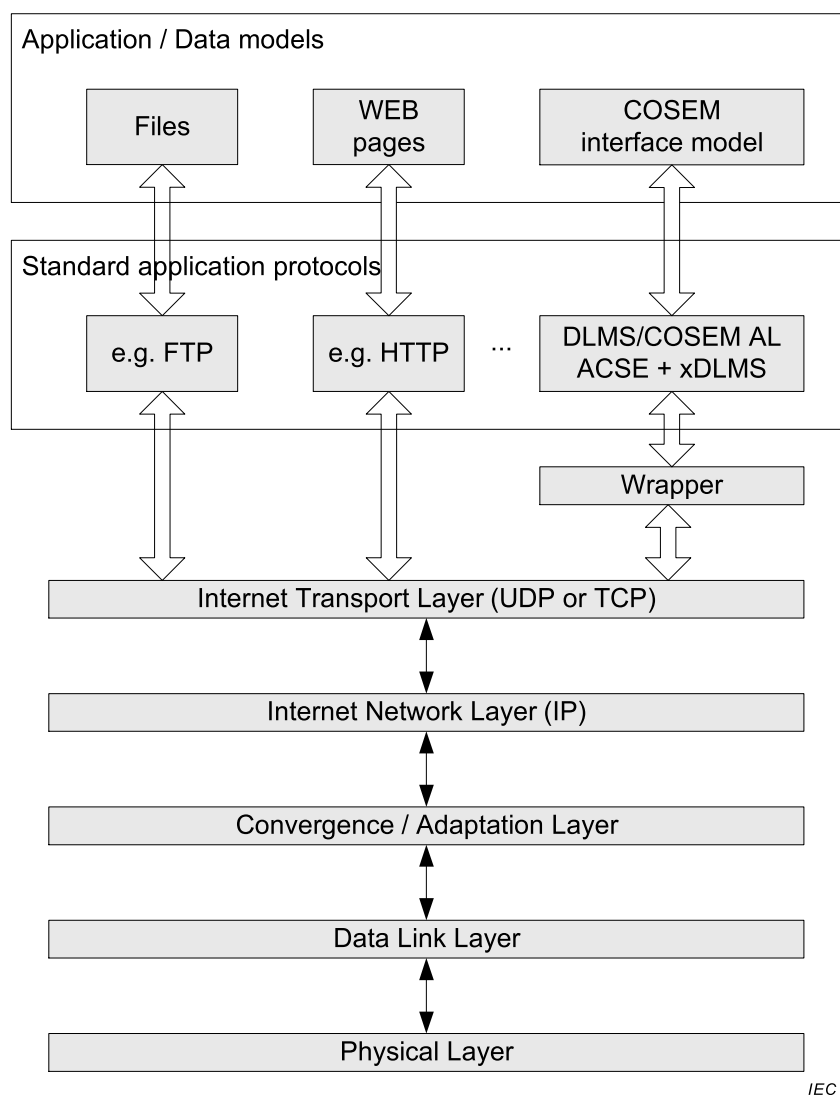
WPDU

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.



IEC

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

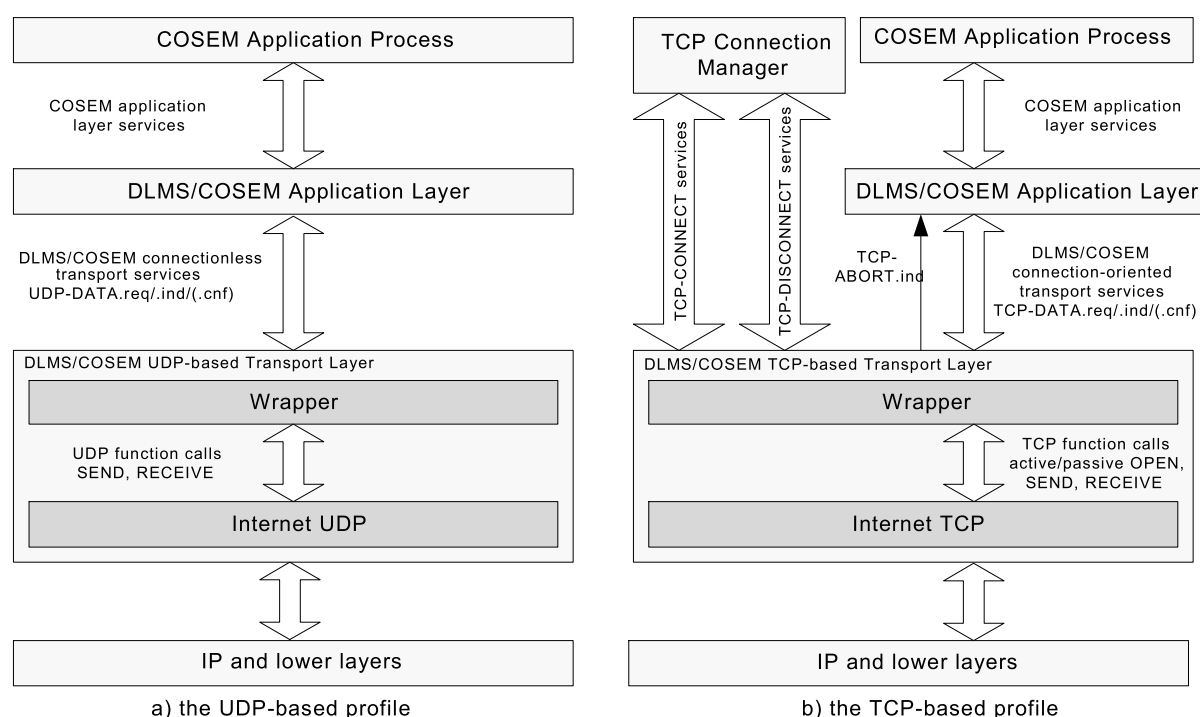
wrapper sublayer is a lightweight, nearly state-less entity: its main function is to adapt the OSI-style service set, provided by the DLMS/COSEM TL, to UDP or TCP function calls and vice versa.

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.



IEC

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.

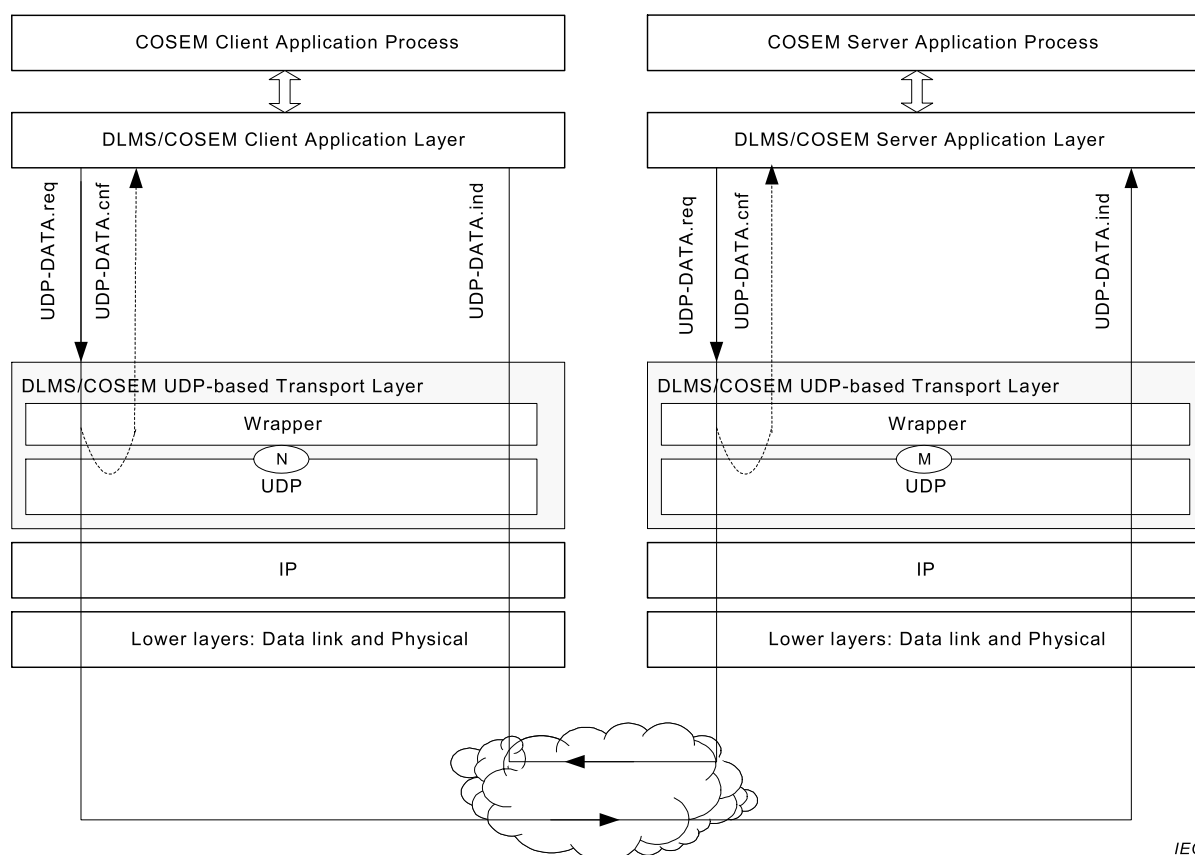
5.2 Service specification for the DLMS/COSEM UDP-based transport layer

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.



IEC

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)

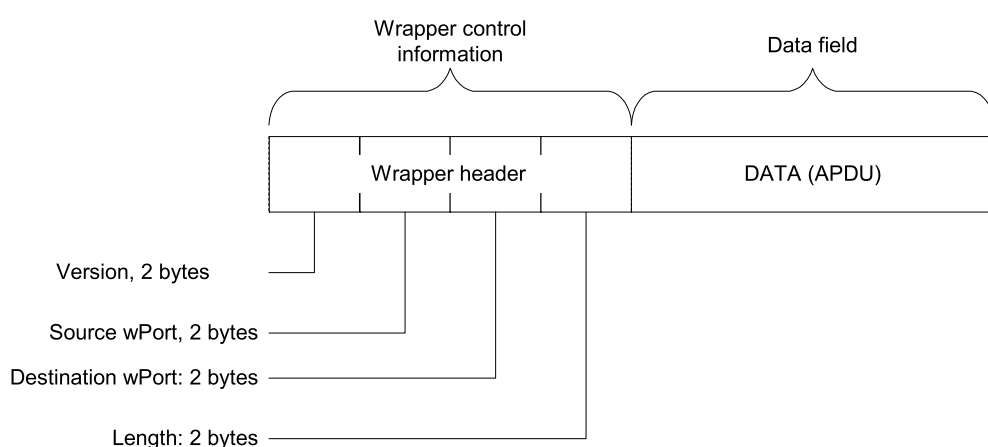
The WPDU consists of two parts:

- 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).



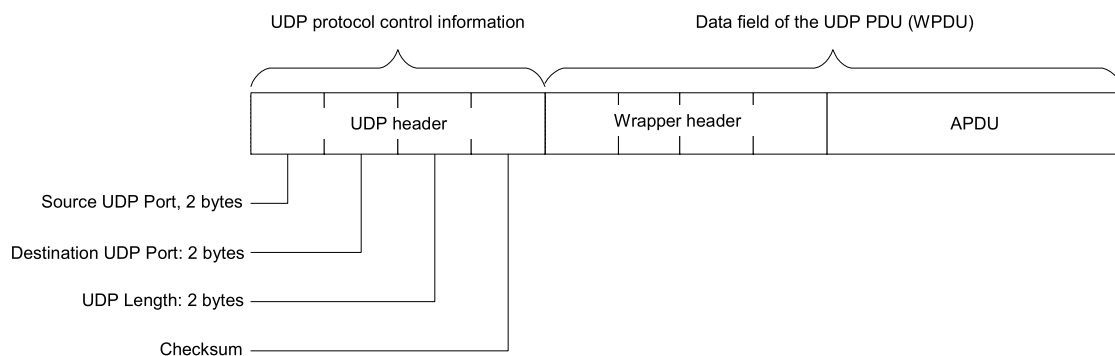
IEC

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, WPDU's shall be transmitted in UDP Datagrams, specified in Internet Standard STD 0006. They shall encapsulate the WPDU, as shown in Figure 5.



IEC

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
<i>Open for client SAP assignment</i>	0x02...0x0F
	0x11...0xFF
Server side reserved addresses	
	Wrapper Port Number
No-station	0x0000
Management Logical Device	0x0001
Reserved	0x0002..0x000F
<i>Open for server SAP assignment</i>	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.

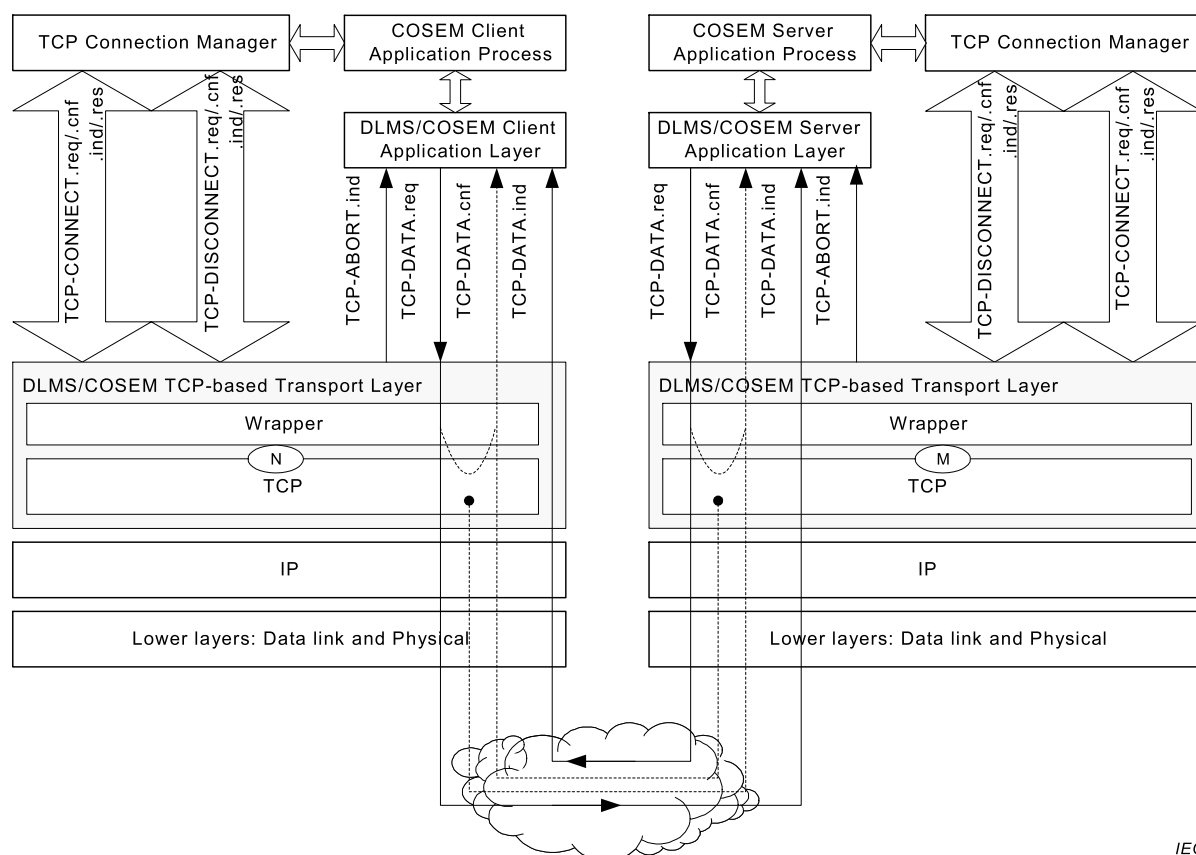
The DLMS/COSEM connection-oriented, TCP-based TL contains the same wrapper sublayer as the DLMS/COSEM UDP-based TL. In addition to transforming OSI-style services to and from TCP function calls, this wrapper provides additional addressing and length information.

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.



IEC

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.

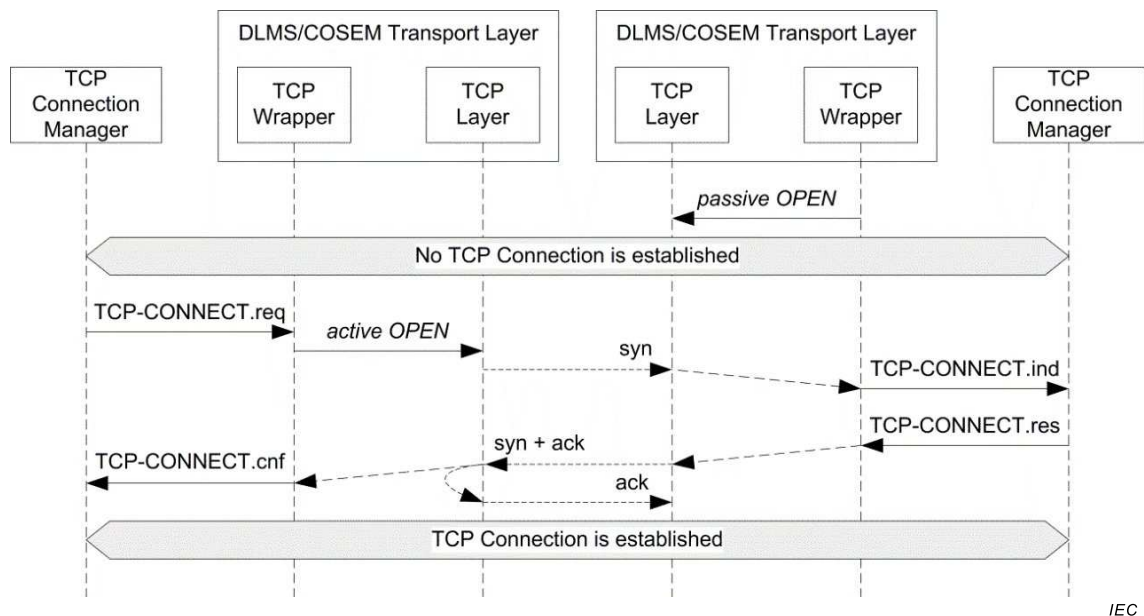
From the service user point of view, only the TCP-CONNECT service primitives are visible: according to this, the TCP connection establishment takes place as it is shown in Figure 8.

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.



IEC

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.

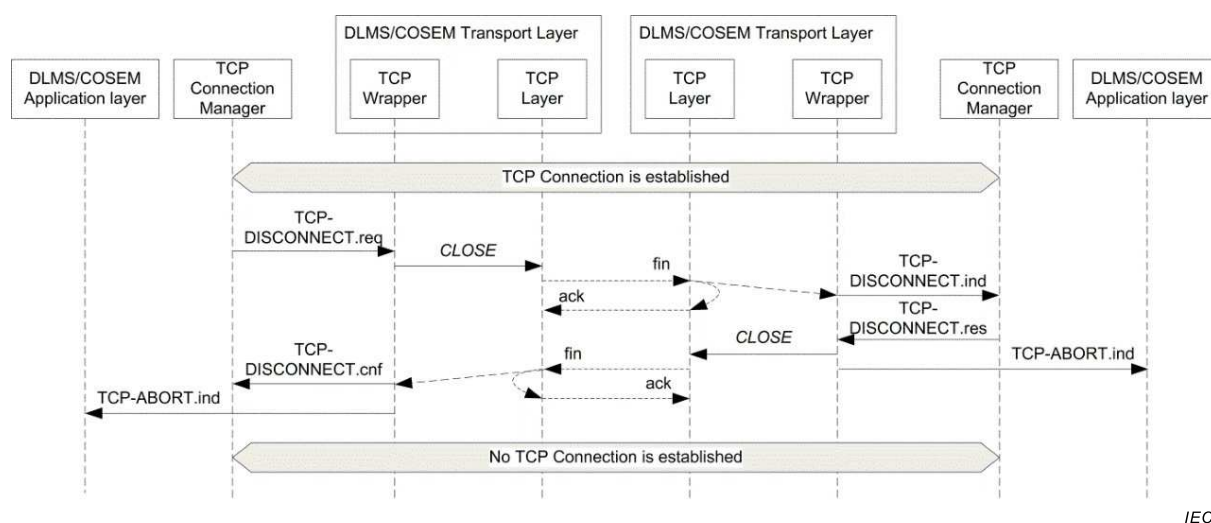


Figure 9 – TCP disconnection

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.

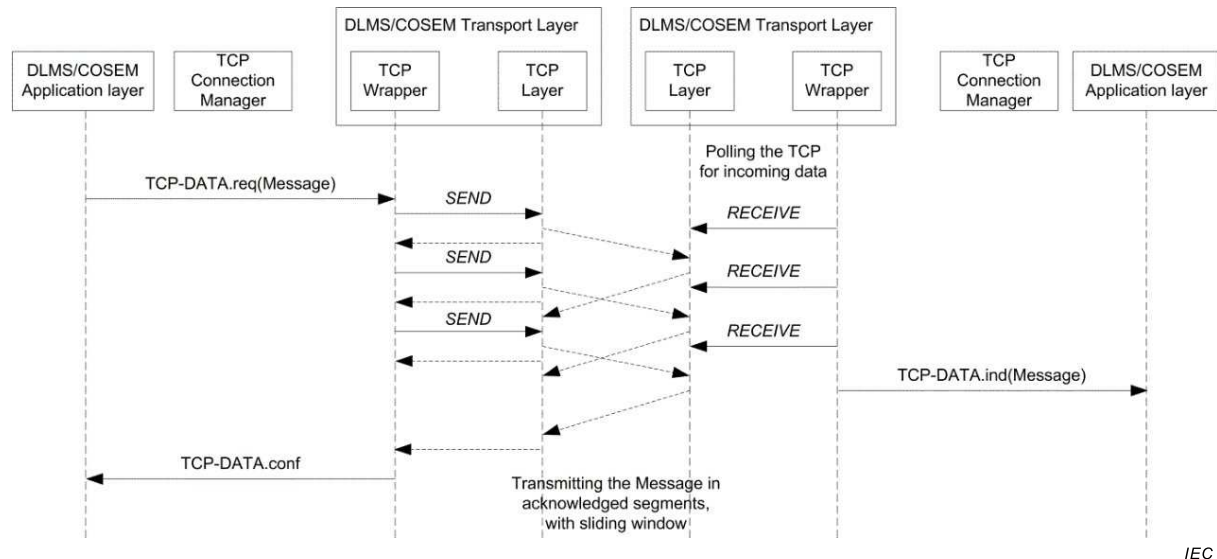


Figure 10 – Data transfer using the DLMS/COSEM TCP-based transport layer

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.

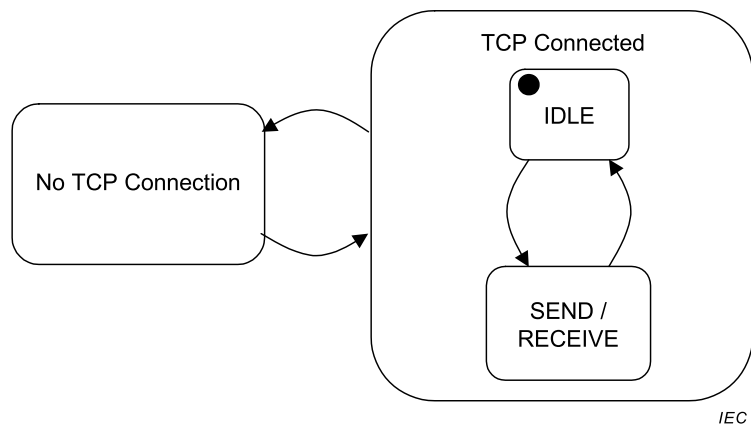


Figure 11 – High-level state transition diagram for the wrapper sublayer

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.

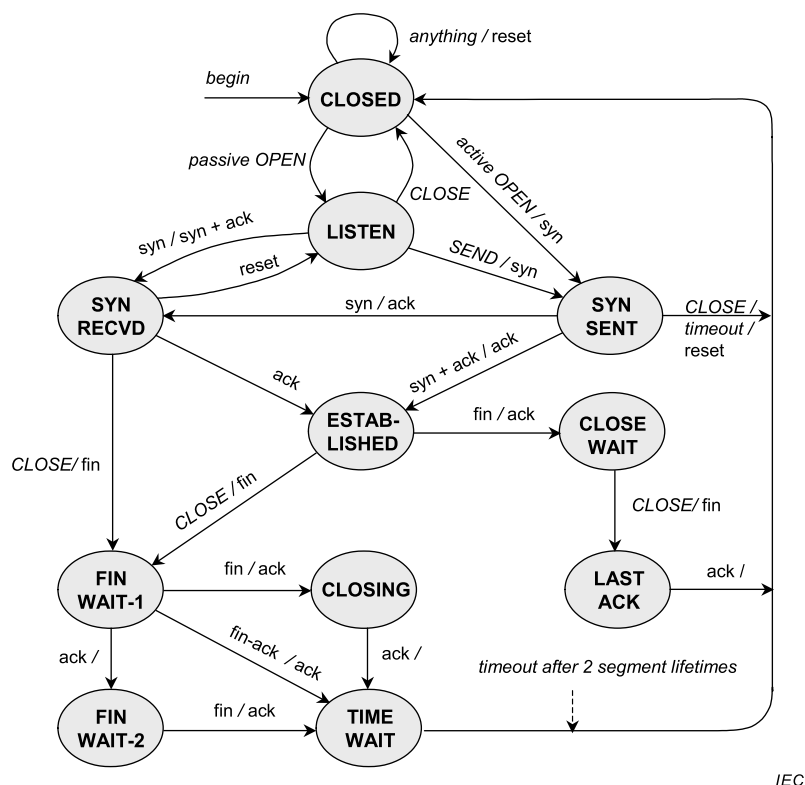
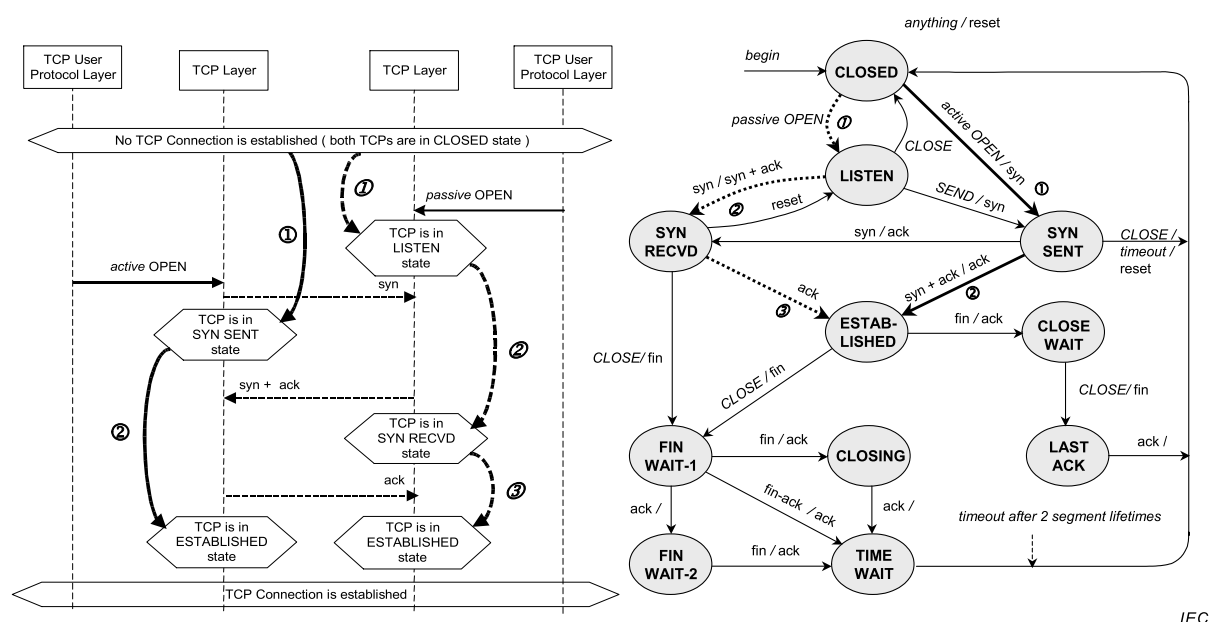


Figure A.1 – TCP connection state diagram

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.

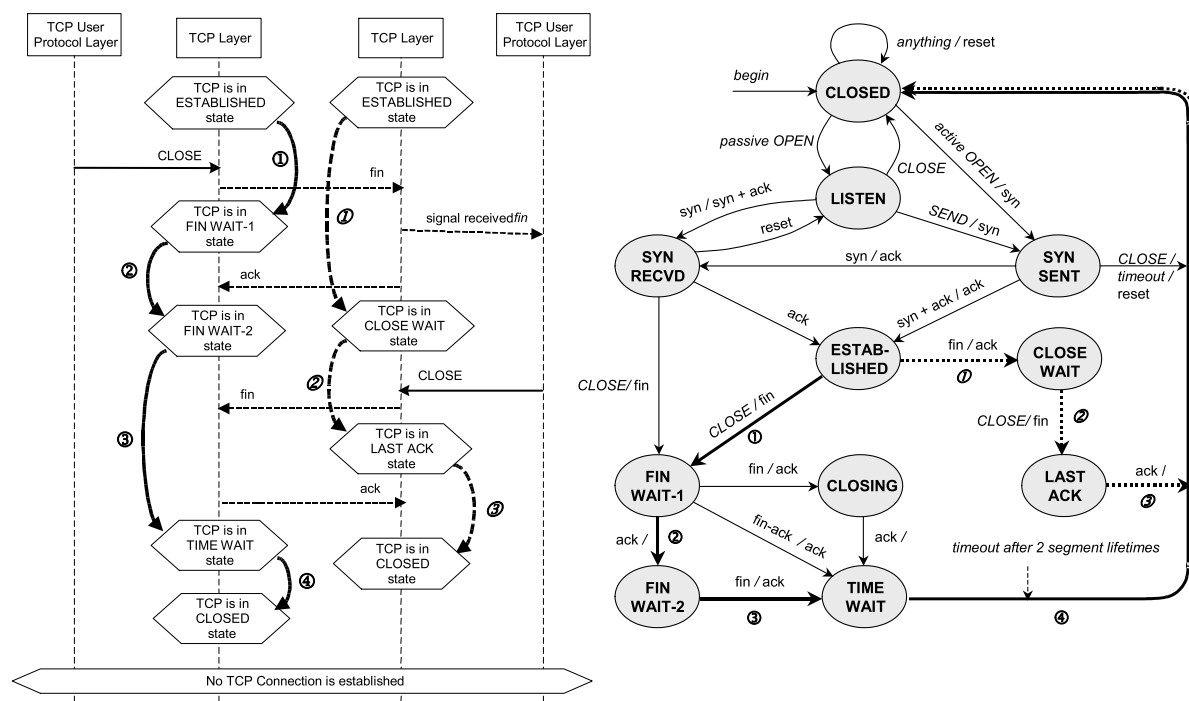
A.2 Closing a transport layer and a TCP connection

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.



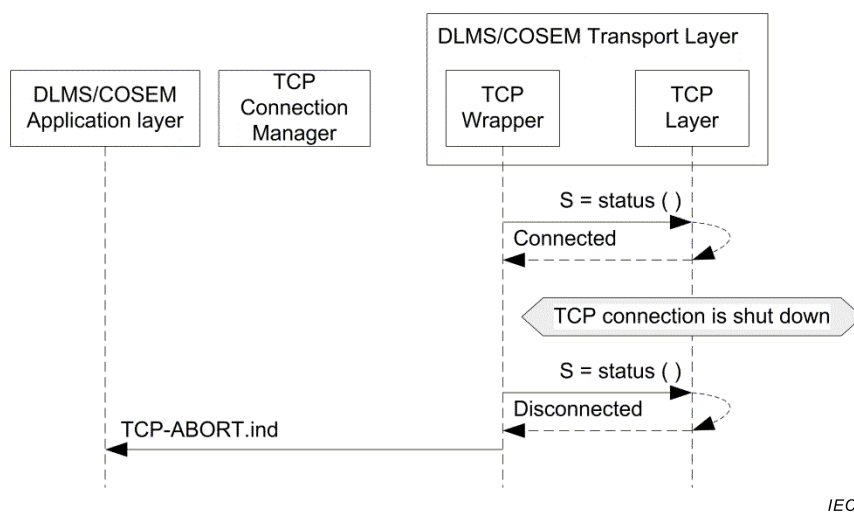
IEG

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



IEC

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

A.4 Data transfer using the TCP-DATA service

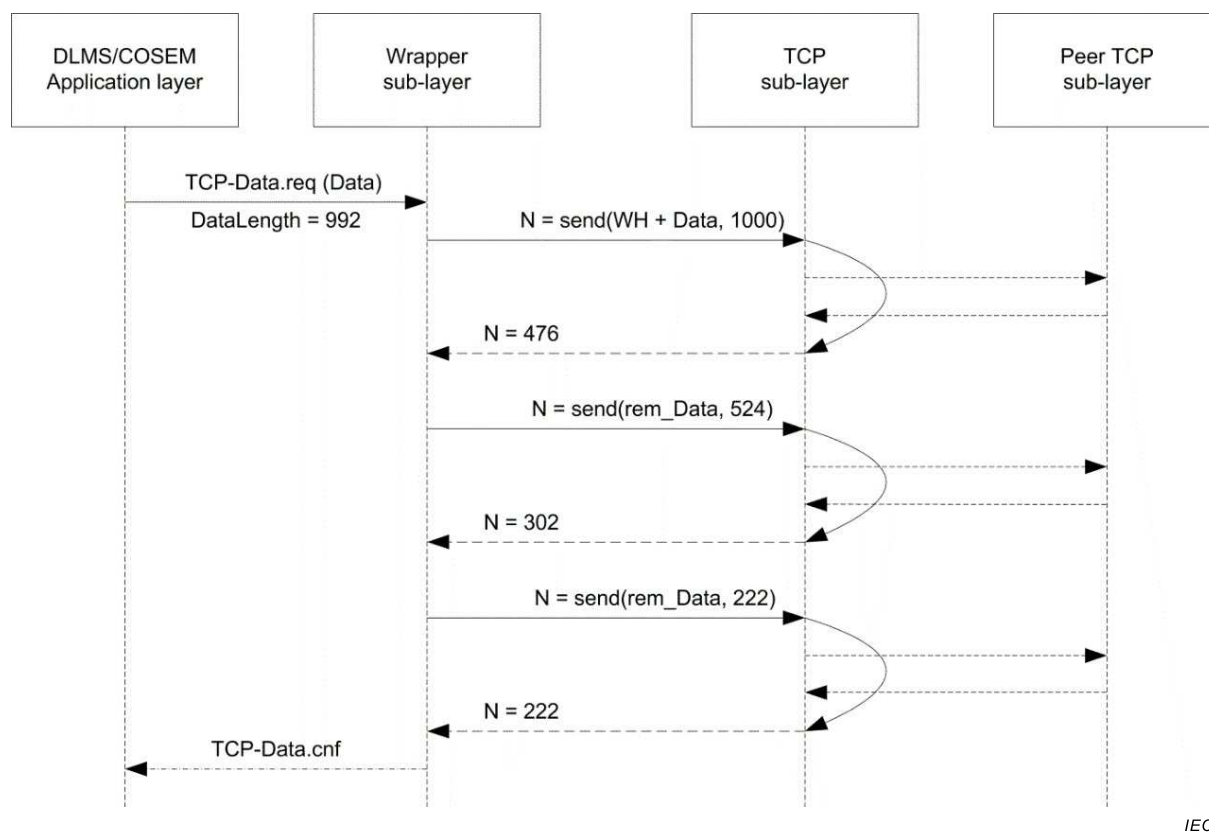
To send an APDU to the peer, the DLMS/COSEM AL simply invokes the TCP-DATA.request primitive of the DLMS/COSEM TCP-based TL. Also, when a complete APDU is received, this is indicated to the DLMS/COSEM AL with the help of the TCP-DATA.indication primitive. Thus, for the AL the TL behaves as if it would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



IEC

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

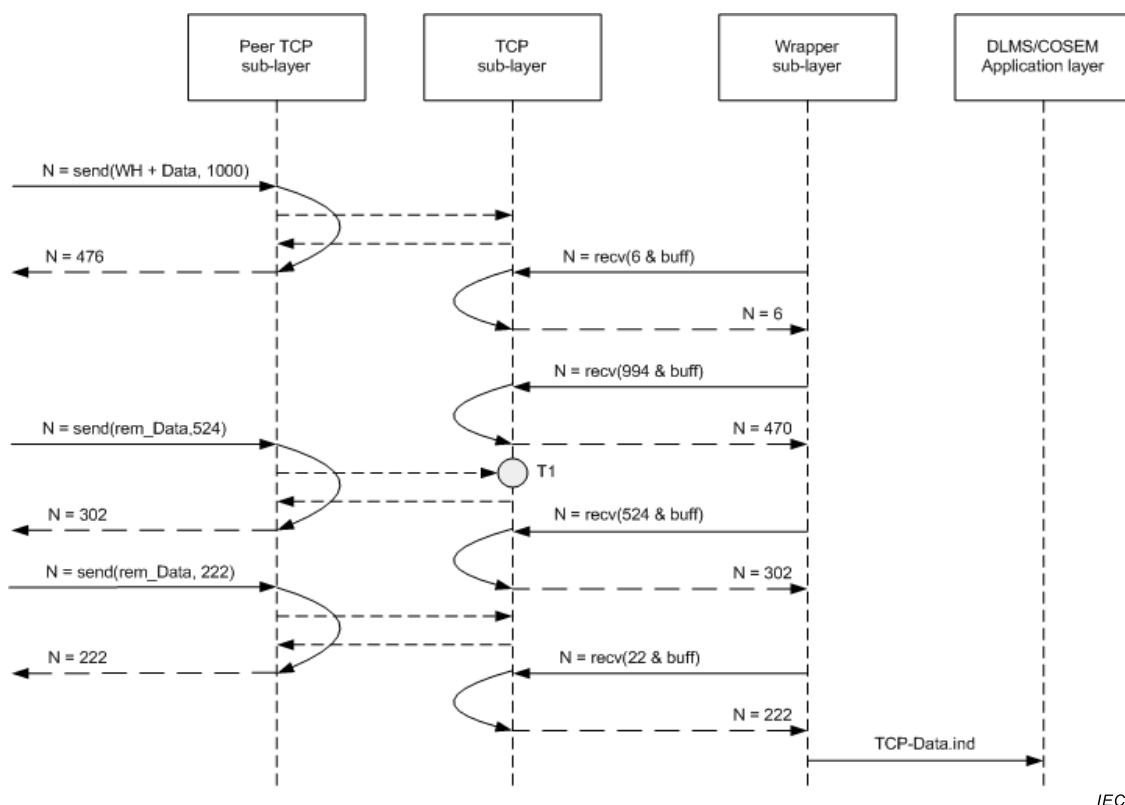
Upon the reception of this service invocation, the wrapper sublayer constructs the WPDU: it pre-fixes the APDU with the wrapper header (WH), including the local and remote wPort numbers and the APDU length. It calls then the SEND() function of the TCP sublayer, requesting to send the WPDU, which is now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



IEC

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

- active OPEN, 32
- Addressing capability (wPort), 10
- Application process identification, 26
- COSEM transport layer, state machine, 16
- COSEM transport layer, TCP based, protocol specification, 26
- COSEM transport layer, UDP based, 10
- COSEM transport layer, UDP based, protocol specification, 14
- Data length, 10, 15
- Data_Length, 13
- Destination wPort, 15
- DLMS/COSEM application layer, 9
- DLMS/COSEM transport layer, TCP based, 16
- DLMS/COSEM transport layer, TCP based, service specification, 17
- DLMS/COSEM transport layer, UDP based, service specification, 11
- Error detection, 17
- fin segment, 29
- Flow control, 17
- Full-duplex, 17
- IDLE sub-state, 30
- Initiator, 28
- Internet Protocol, 9
- Local_IP_Address, 12
- Local_TCP_Port, 19
- Local_UDP_Port, 12
- Local_wPort, 12
- Multi- and broadcasting using UDP, 13
- No TCP Connection, 30
- OSI-style services, 9
- Passive opening, 28
- Positive Acknowledgement with Retransmission, 16, 30
- Remote_IP_Address, 12
- Remote_TCP_Port, 19
- Remote_UDP_Port, 12
- Remote_wPort, 12
- Reserved wrapper port numbers, 16, 27
- Responder, 28
- SEND() function, 13
- SEND/RECEIVE state, 30
- Sequence numbers, 33
- Source UDP, 16
- Source wPort, 15
- TCP Connected, 30
- TCP connection, 27
- TCP connection abort, 29, 34
- TCP connection closing, 17, 33
- TCP connection establishment, 17, 32
- TCP connection manager process, 10, 18
- TCP data communication, 17
- TCP disconnection, 28
- TCP packets, 27
- TCP-ABORT service, 23
- TCP-ABORT.indication, 23
- TCP-CONNECT, 18
- TCP-CONNECT.confirm, 20
- TCP-CONNECT.indication, 19
- TCP-CONNECT.request, 18
- TCP-CONNECT.response, 19
- TCP-DATA service, 24, 35
- TCP-DATA services, 30
- TCP-DATA.confirm, 25
- TCP-DATA.indication, 25
- TCP-DATA.request, 24
- TCP-DISCONNECT services, 21
- TCP-DISCONNECT.confirm, 22
- TCP-DISCONNECT.indication, 21
- TCP-DISCONNECT.request, 21
- TCP-DISCONNECT.response, 22
- Three-way handshake, 27, 32
- Transmission Control Protocol, 7, 16
- UDP Datagram, 13
- UDP-DATA service, 11
- UDP-DATA.confirm, 14
- UDP-DATA.indication, 13
- UDP-DATA.request, 12
- User Datagram Protocol, 7, 10
- Valid wPort numbers, 13, 25
- Virtual circuit, 16
- Wrapper, 9
- Wrapper header, 14
- Wrapper protocol data unit, 14, 26
- Wrapper sublayer, 9, 14, 26
- Wrapper sublayer, state transition diagram, 30

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>

SOMMAIRE

AVANT-PROPOS	42
INTRODUCTION	44
1 Domaine d'application	45
2 Références normatives	45
3 Termes, définitions et abréviations	46
3.1 Termes et définitions	46
3.2 Abréviations	46
4 Présentation générale	47
5 Couche transport DLMS/COSEM sans connexion basée sur UDP	50
5.1 Généralités	50
5.2 Spécification de service pour la couche transport DLMS/COSEM basée sur UDP	51
5.2.1 Généralités	51
5.2.2 Services UDP-DATA	52
5.3 Spécification de protocole pour la couche transport DLMS/COSEM basée sur UDP	54
5.3.1 Généralités	54
5.3.2 Unité de données de protocole de l'enveloppe (WPDU)	54
5.3.3 Unité de données de protocole couche transport DLMS/COSEM basée sur UDP	55
5.3.4 Numéros de port de l'enveloppe réservés (wPorts)	56
5.3.5 Diagramme d'états de protocole	57
6 Couche transport DLMS/COSEM, orientée connexion, basée sur TCP	57
6.1 Généralités	57
6.2 Spécification de service pour la couche transport DLMS/COSEM basée sur TCP	57
6.2.1 Généralités	57
6.2.2 Service TCP-CONNECT	59
6.2.3 Service TCP-DISCONNECT	61
6.2.4 Service TCP-ABORT	63
6.2.5 Service TCP-DATA	64
6.3 Spécification de protocole pour la couche transport DLMS/COSEM basée sur TCP	66
6.3.1 Généralités	66
6.3.2 Unité de données de protocole de l'enveloppe (WPDU)	67
6.3.3 Unité de données de protocole couche transport DLMS/COSEM basée sur TCP	67
6.3.4 Numéros de port de l'enveloppe réservés	67
6.3.5 Définition des procédures	67
Annexe A (informative) Conversion de services de TL du modèle OSI vers et depuis des appels de fonction TCP type RFC	73
A.1 Établissement d'une connexion de couche transport et TCP	73
A.2 Fermeture d'une connexion de couche transport et TCP	74
A.3 Abandon de connexion TCP	76
A.4 Transfert de données utilisant le service TCP-DATA	76
INDEX	79
Bibliographie	80

Figure 1 – DLMS/COSEM en tant que protocole d'application Internet standard	48
Figure 2 – Couches transport du profil DLMS/COSEM_on_IP	50
Figure 3 – Services de la couche transport DLMS/COSEM sans connexion, basée sur UDP51	
Figure 4 – Unité de données de protocole de l'enveloppe (WPDU).....	55
Figure 5 – PDU (UDP-PDU) de couche transport DLMS/COSEM sans connexion, basée sur UDP	56
Figure 6 – Services de la couche transport DLMS/COSE orientée connexion basée sur TCP58	
Figure 7 – Format de paquet TCP	67
Figure 8 – Établissement de la connexion TCP	68
Figure 9 – Interruption de TCP.....	69
Figure 10 – Transfert de données utilisant la couche transport DLMS/COSEM basée sur TCP	71
Figure 11 – Diagramme de transition d'états de haut niveau de la sous-couche enveloppe	72
Figure A.1 – Diagramme d'états de connexion TCP	73
Figure A.2 – MSC et transitions d'états pour l'établissement d'une connexion de couche transport et TCP	74
Figure A.3 – MSC et transitions d'états pour la fermeture d'une connexion de couche transport et TCP	75
Figure A.4 – Demande d'indication d'abandon TCP à la sous-couche TCP	76
Figure A.5 – Envoi d'une APDU en trois paquets TCP.....	77
Figure A.6 – Réception du message en plusieurs paquets	78
Tableau 1 – Numéros de port de l'enveloppe réservés dans la TL DLMS/COSEM basée sur UDP.....	56

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

ÉCHANGE DES DONNÉES DE COMPTAGE DE L'ÉLECTRICITÉ – LA SUITE DLMS/COSEM –

Partie 4-7: Couche transport DLMS/COSEM pour réseaux IP

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

La Commission Electrotechnique Internationale (IEC) attire l'attention sur le fait qu'il est déclaré que la conformité avec les dispositions de la présente Norme internationale peut impliquer l'utilisation d'un service de maintenance concernant l'ensemble de protocoles sur lesquels la présente norme IEC 62056-4-7 est fondée.

L'IEC ne prend pas position quant à la preuve, à la validité et à la portée de ce service de maintenance.

Le fournisseur du service de maintenance a donné l'assurance à l'IEC qu'il consent à fournir des services, à des termes et conditions raisonnables et non discriminatoires pour les demandeurs du monde entier. À ce propos, la déclaration du fournisseur du service de maintenance est enregistrée à l'IEC. Des informations peuvent être demandées à:

DLMS User Association
Zug / Suisse
www.dlms.com

La Norme internationale IEC 62056-4-7 a été établie par le comité d'études 13 de l'IEC: Comptage et pilotage de l'énergie électrique.

Cette première édition annule et remplace l'IEC 62056-47 parue en 2006. Cette édition constitue une révision technique.

Cette édition inclut les modifications techniques majeures suivantes par rapport à l'édition précédente:

- a) La présente norme s'applique désormais aux deux réseaux IP4 et IPv6;
- b) Les dernières éditions de la série IEC 62056 sont référencées;
- c) Ajout des numéros de port assignés par l'IANA pour le DLMS/COSEM.

Le texte de cette norme est issu des documents suivants:

CDV	Rapport de vote
13/1570/CDV	13/1595/RVC

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

Cette publication a été rédigée selon les Directives ISO/IEC, Partie 2.

Une liste de toutes les parties de la série IEC 62056, publiées sous le titre général *Echange des données de comptage de l'électricité – La suite DLMS/COSEM*, peut être consultée sur le site web de l'IEC.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. À cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

INTRODUCTION

La présente norme spécifie la couche transport DLMS/COSEM pour les réseaux IP (IPv4 ou IPv6). Elle doit être lue conjointement avec l'IEC 62056-9-7:2013, *Échange des données de comptage de l'électricité – La suite DLMS/COSEM – Partie 9-7: Profil de communication pour réseaux TCP-UDP/IP*.

ÉCHANGE DES DONNÉES DE COMPTAGE DE L'ÉLECTRICITÉ – LA SUITE DLMS/COSEM –

Partie 4-7: Couche transport DLMS/COSEM pour réseaux IP

1 Domaine d'application

La présente partie de l'IEC 62056 spécifie une couche transport sans connexion et une couche transport orientée connexion pour les profils de communication DLMS/COSEM utilisés sur les réseaux IP.

Ces couches transport (TL – Transport Layer) offrent des services du modèle OSI à la couche application (AL – Application Layer) DLMS/COSEM utilisateur de service. La couche transport sans connexion est basée sur la norme Internet User Datagram Protocol (UDP, protocole de datagramme utilisateur). La couche transport orientée connexion est basée sur la norme Internet Transmission Control Protocol (TCP, protocole de contrôle de transmission).

La couche transport DLMS/COSEM comprend la couche transport UDP ou TCP et une sous-couche supplémentaire, appelée «enveloppe».

L'Annexe A montre comment les services de couche transport du modèle OSI peuvent être convertis vers et depuis les appels de fonction UDP et TCP.

2 Références normatives

Les documents suivants sont cités en référence de manière normative, en intégralité ou en partie, dans le présent document et sont indispensables pour son application. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 60050-300:2001, *Vocabulaire Électrotechnique International (VEI) – Mesures et appareils de mesure électriques et électroniques – Partie 311: Termes généraux concernant les mesures – Partie 312: Termes généraux concernant les mesures électriques – Partie 313: Types d'appareils électriques de mesure – Partie 314: Termes spécifiques selon le type d'appareil.*

IEC TR 62051:1999, *Electricity metering – Glossary of terms* (disponible en anglais seulement)

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* (disponible en anglais seulement)

IEC 62056-5-3:2013, *Échange des données de comptage de l'électricité – La suite DLMS/COSEM – Partie 5-3: Couche application DLMS/COSEM*¹

IEC 62056-6-2:2013, *Echange des données dans les équipements de comptage de l'énergie électrique – La suite DLMS/COSEM – Partie 6-2: Classes d'interfaces COSEM*²

¹ Edition 2 de l'IEC 62056-5-3 à publier.

² Edition 2 de l'IEC 62056-6-2 à publier.

IEC 62056-9-7:2013, *Echange de données de comptage de l'électricité - La suite DLMS/COSEM - Partie 9-7: Profil de communication pour réseaux TCP-UDP/IP*

STD 0006, *User Datagram Protocol (Protocole de datagramme utilisateur)*. Édité par Jon Postel, août 1980. Disponible à l'adresse suivante: <http://www.faqs.org/rfcs/std/std6.html>

STD 0007, *Transmission Control Protocol (Protocole de contrôle de transmission)*. Édité par Jon Postel, septembre 1981. Disponible à l'adresse suivante: <http://www.faqs.org/rfcs/std/std7.html>

NOTE Voir également la Bibliographie pour d'autres RFC Internet en rapport.

3 Termes, définitions et abréviations

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions donnés dans l'IEC 60050-300, l'IEC TR 62051 et l'IEC TR 62051-1 ainsi que les suivants s'appliquent.

3.1.1

processus d'application

élément d'un système ouvert réel effectuant un traitement d'information pour une application particulière

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

3.1.2

entité d'application

activités d'application indépendantes du système mises à disposition en tant que services d'application à l'agent d'application, par exemple un ensemble d'éléments de service d'application qui réunis réalisent tout ou partie des aspects de communication d'un processus d'application

3.2 Abréviations

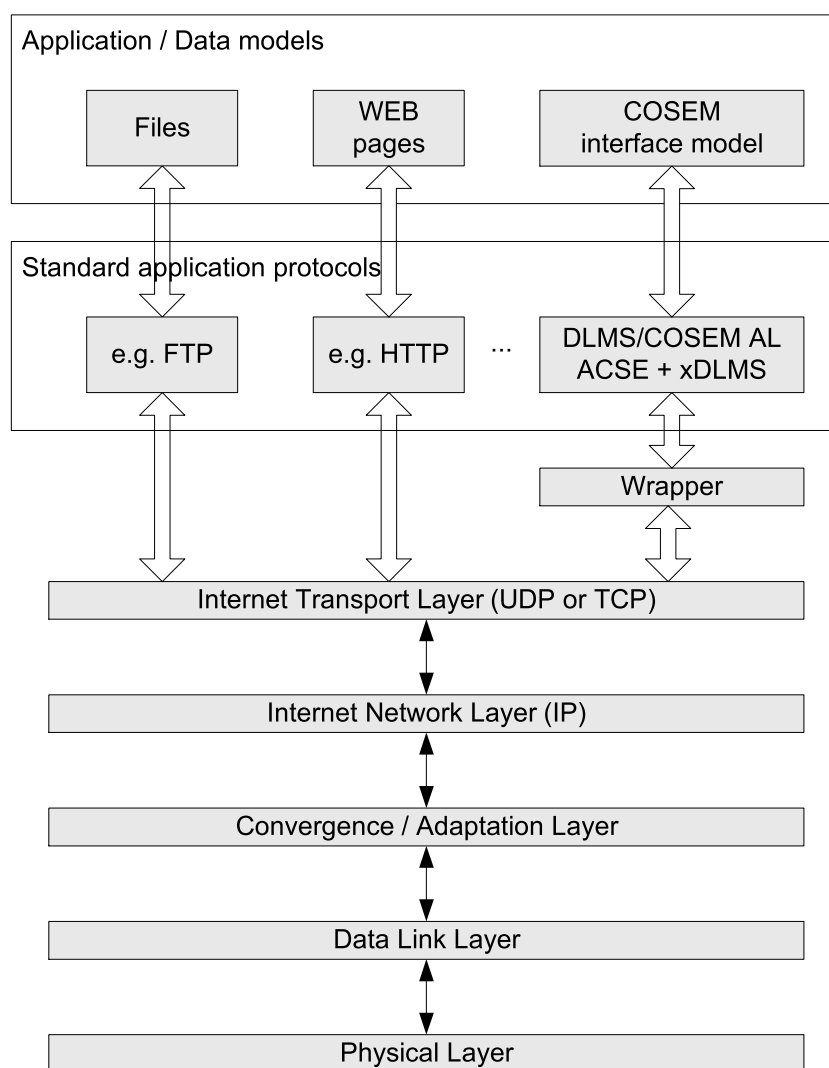
APDU	Application Layer Protocol Data Unit (Unité de données de protocole d'application)
AL	Application Layer (Couche application)
AP	Application Process (Processus d'application)
AE	Application Entity (Entité d'application)
COSEM	COmpanion Specification for Energy Metering (Spécification auxiliaire pour le comptage d'énergie)
DHCP	Dynamic Host Configuration Protocol (Protocole de configuration dynamique d'hôte)
DLMS	Device Language Message Specification (Spécification de message de langage de dispositif)
COSEM_on_IP	Profil de communication COSEM TCP-UDP/IP
FTP	File Transfer Protocol (Protocole de transfert de fichiers)
HTTP	HyperText Transfer Protocol (Protocole de transfert hypertexte)
IANA	Internet Assigned Numbers Authority
IP	Internet Protocol (Protocole Internet)
PDU	Protocol Data Unit (Unité de données de protocole)

PAR	Positive Acknowledgement with Retransmission (Acquittement positif avec retransmission)
SNMP	Simple Network Management Protocol (Protocole de gestion de réseau simple)
TCP	Transmission Control Protocol (Protocole de contrôle de transmission)
TFTP	Trivial File Transfer Protocol (Protocole trivial de transfert de fichier)
TL	Transport Layer (Couche transport)
UDP	User Datagram Protocol (Protocole de datagramme utilisateur)
WPDU	Wrapper Protocol Data Unit (Unité de données de protocole de l'enveloppe)

4 Présentation générale

Dans les profils DLMS/COSEM_on_IP, l'AL DLMS/COSEM utilise les services d'une de ces TL, qui utilisent les services de la couche réseau du protocole Internet (IP) pour communiquer avec les autres nœuds connectés au réseau IP abstrait.

Quand elle est utilisée dans ces profils, l'AL DLMS/COSEM peut être considérée comme un protocole d'application Internet standard (tel que les protocoles bien connus HTTP, FTP ou SNMP) et peut coexister avec d'autres protocoles d'application Internet, comme indiqué dans la Figure 1.



IEC

Anglais	Français
Application / Data models	Application / Modèles de données
Files	Fichiers
WEB pages	Pages Web
COSEM interface model	Modèle d'interface COSEM
Standard application protocols	Protocoles d'application standard
e.g.	par exemple
Wrapper	Enveloppe
Internet Transport Layer (UDP or TCP)	Couche transport Internet (UDP ou TCP)
Internet Network layer (IP)	Couche réseau Internet (IP)
Convergence / Adaptation Layer	Couche de convergence / d'adaptation
Data Link Layer	Couche liaison de données
Physical Layer	Couche physique

Figure 1 – DLMS/COSEM en tant que protocole d'application Internet standard

Pour DLMS/COSEM, les numéros de ports suivants ont été enregistrés par l'IANA. Voir <http://www.iana.org/assignments/port-numbers>.

- dlms/cosem 4059/TCP DLMS/COSEM

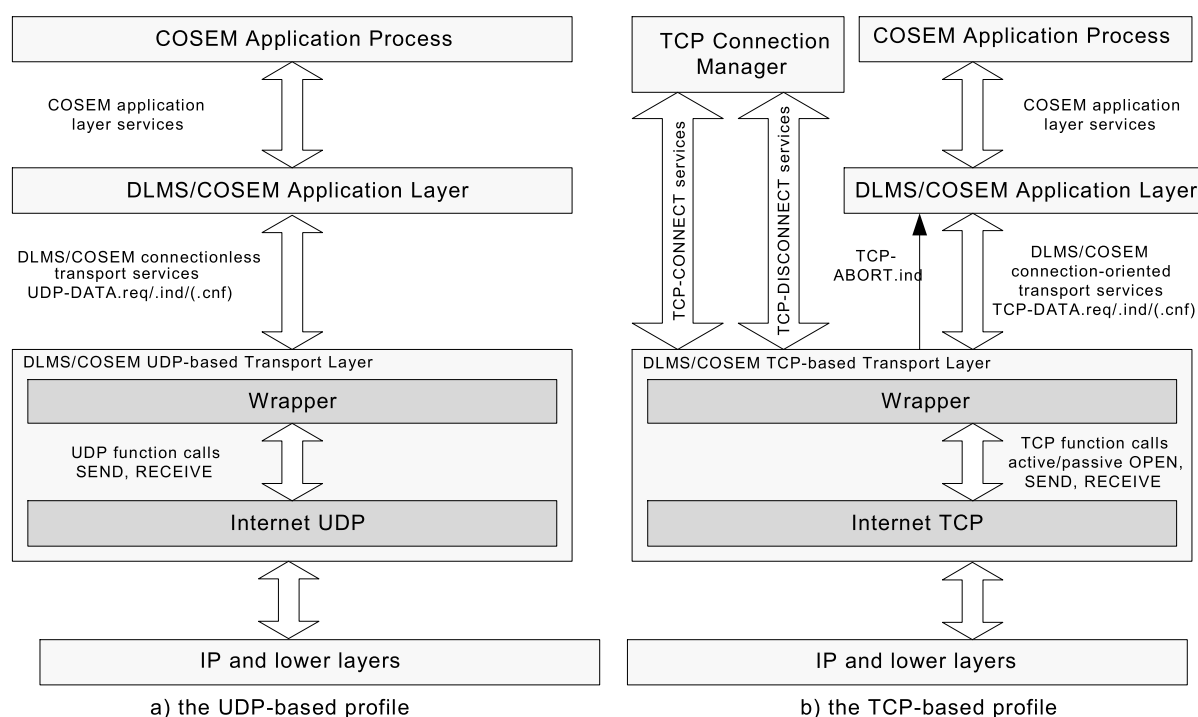
- dlms/cosem 4059/UDP DLMS/COSEM

Étant donné que l'AL DLMS/COSEM spécifiée dans l'IEC 62056-5-3 utilise et fournit des services du modèle OSI, une enveloppe a été introduite entre les couches UDP/TCP et l'AL DLMS/COSEM. Par conséquent, les TL DLMS/COSEM consistent en une sous-couche enveloppe et une TL UDP ou TCP. La sous-couche enveloppe est une entité légère, presque sans état: sa fonction principale consiste à adapter l'ensemble des services du modèle OSI fournis par la TL DLMS/COSEM aux appels de fonction UDP ou TCP et inversement.

En outre, la sous-couche enveloppe présente les fonctions suivantes:

- elle offre une capacité d'adressage supplémentaire (wPort) au-dessus du port UDP/TCP;
- elle fournit des informations relatives à la longueur des données transportées. Cette fonction aide l'expéditeur à envoyer et le destinataire à reconnaître la réception d'une APDU complète, pouvant être envoyée et reçue sous forme de paquets TCP multiples.

Comme spécifié à l'Article 6 de l'IEC 62056-9-7:2013, l'AL DLMS/COSEM n'écoute qu'un seul port UDP ou TCP. D'autre part, comme défini en 4.7 de l'IEC 62056-6-2:2013, un dispositif physique DLMS/COSEM peut héberger plusieurs AP client ou dispositifs logiques serveur. La capacité d'adressage supplémentaire fournie par la sous-couche enveloppe permet d'identifier ces AP.



IEC

La structure de la TL DLMS/COSEM et sa place dans COSEM-on_IP sont indiquées dans la Figure 2.

Anglais	Français
COSEM Application Process	Processus d'application COSEM
DLMS/COSEM application layer	Couche application DLMS/COSEM
TCP Connection Manager	Gestionnaire de connexion TCP
COSEM application layer services	Services de couche application COSEM
DLMS/COSEM connectionless transport services	Services de transport sans connexion DLMS/COSEM
DLMS/COSEM connection-oriented transport services	Services de transport orientés connexion DLMS/COSEM

Anglais	Français
TCP-CONNECT services	Services TCP-CONNECT
TCP-DISCONNECT services	Services TCP-DISCONNECT
DLMS/COSEM UDP-based Transport Layer	Couche transport DLMS/COSEM basée sur UDP
DLMS/COSEM TCP-based Transport Layer	Couche transport DLMS/COSEM basée sur TCP
Wrapper	Enveloppe
UDP Function calls SEND, RECEIVE	Appels de fonction UDP SEND, RECEIVE
TCP Function calls active/passive OPEN, SEND, RECEIVE	Appels de fonction TCP active/passive OPEN, SEND, RECEIVE
Internet UDP	UDP Internet
Internet TCP	TCP Internet
IP and lower layers	IP et couches inférieures
a) the UDP-based profile	Profil UDP
b) the TCP-based profile	Profil TCP

Figure 2 – Couches transport du profil DLMS/COSEM_on_IP

L'utilisateur des services UDP-DATA et TCP-DATA est l'AL DLMS/COSEM. D'autre part, l'utilisateur des services TCP-CONNECT et TCP-DISCONNECT est le processus gestionnaire de connexion TCP. La TL DLMS/COSEM basée sur TCP fournit également un service TCP-ABORT à l'AL DLMS/COSEM utilisateur de service.

5 Couche transport DLMS/COSEM sans connexion basée sur UDP

5.1 Généralités

La TL DLMS/COSEM sans connexion est basée sur le protocole de datagramme utilisateur (UDP) tel que spécifié dans le document STD 0006.

L'UDP fournit une procédure pour que les programmes d'application envoient des messages à d'autres programmes en utilisant un minimum de mécanisme de protocole. D'une part, le protocole est orienté transaction et la livraison, comme la protection double, ne sont pas garanties. D'autre part, l'UDP est simple, il limite l'augmentation de surdébit, et s'avère efficace et facile d'emploi. Plusieurs applications Internet bien connues, comme SNMP, DHCP, TFTP, etc., bénéficient de ces avantages en termes de performance, soit parce que certaines applications de datagramme n'ont pas besoin d'être fiables, soit parce que le mécanisme de fiabilité requis est assuré par l'application elle-même. Les applications de type demande/réponse, comme par exemple une association d'application COSEM confirmée, établie sur la TL DLMS/COSEM basée sur UDP, qui sollicite ensuite des services de transfert de données xDLMS confirmés, sont un bon exemple de cette seconde catégorie. Un autre avantage de l'UDP tient au fait qu'en tant que protocole sans connexion, il est facilement capable d'assurer la diffusion et la multidiffusion.

Pour l'essentiel, l'UDP fournit une interface supérieure à la couche IP, avec une capacité d'identification supplémentaire, à savoir le numéro de port UDP. Cette caractéristique permet de faire la distinction entre les AP, hébergés dans le même dispositif physique et identifiés par l'adresse IP.

NOTE Le schéma d'adressage/identification pour les profils COSEM_on_IP est défini à l'Article 6 de l'IEC 62056-9-7.

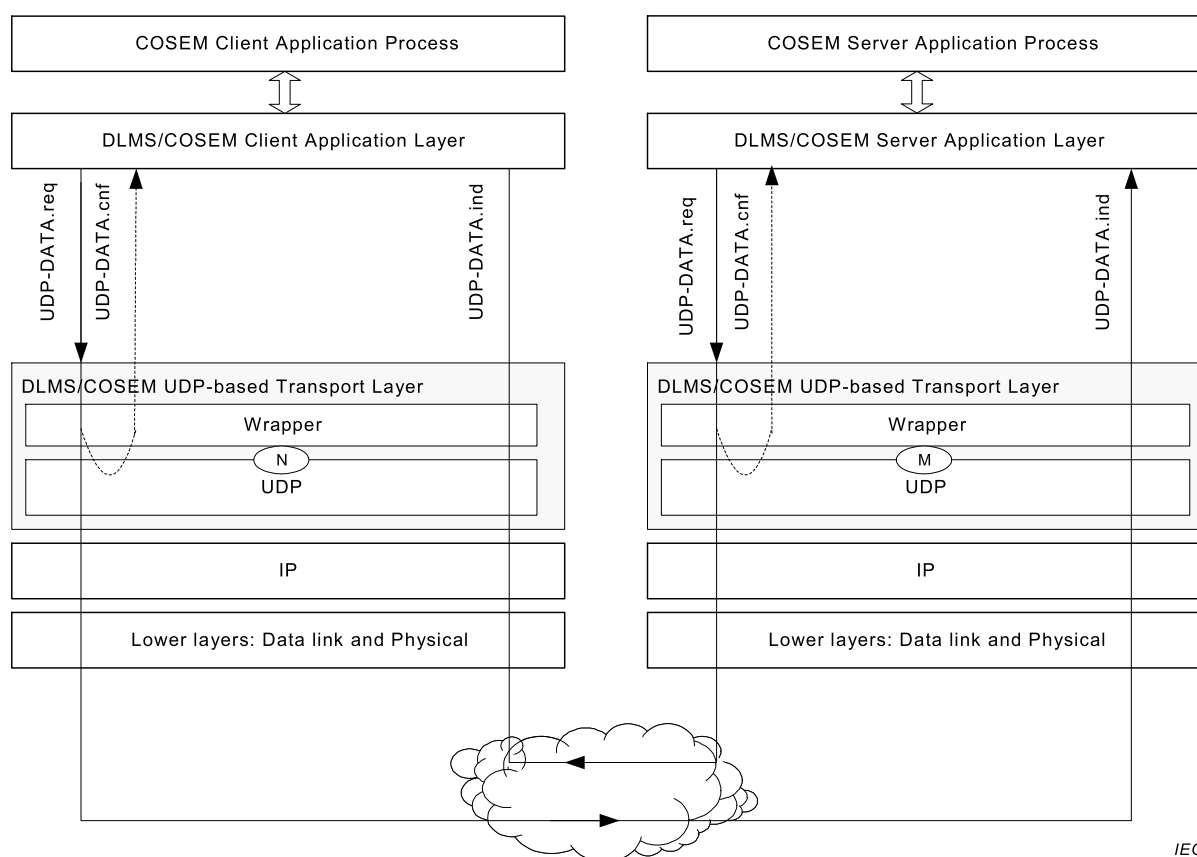
5.2 Spécification de service pour la couche transport DLMS/COSEM basée sur UDP

5.2.1 Généralités

La TL DLMS/COSEM basée sur UDP fournit uniquement un service de transfert de données: le service UDP-DATA sans connexion. Par conséquent, la spécification de service pour ce service est identique pour les TL client et serveur, comme indiqué dans la Figure 3.

Les primitives de service .request et .indication sont obligatoires. La mise en œuvre de la primitive de service locale .confirm est facultative.

L'APDU xDLMS prédéterminée avec l'en-tête de l'enveloppe doit s'adapter dans un datagramme UDP unique.



Anglais	Français
COSEM Client Application process	Processus d'application client COSEM
COSEM Server Application process	Processus d'application serveur COSEM
DLMS/COSEM Client Application layer	Couche application client DLMS/COSEM
DLMS/COSEM Server Application layer	Couche application serveur DLMS/COSEM
DLMS/COSEM UDP-based Transport Layer	Couche transport DLMS/COSEM basée sur UDP
Wrapper	Enveloppe
Lower Layers: Data Link and Physical	Couches inférieures: Liaison de données et Physique

Figure 3 – Services de la couche transport DLMS/COSEM sans connexion, basée sur UDP

5.2.2 Services UDP-DATA

5.2.2.1 UDP-DATA.request

Fonction

Cette primitive est la primitive de demande de service pour le service de transfert de données en mode sans connexion.

Sémantique de la primitive de service

La primitive doit fournir les paramètres comme suit:

```

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

Les paramètres Local_wPort, Local_UDP_Port et Local_IP_Address indiquent les paramètres numéro de port de l'enveloppe, numéro de port UDP et adresse IP appartenant au dispositif/AE DLMS/COSEM demandant l'envoi des données. Les paramètres Remote_wPort, Remote_UDP_Port et Remote_IP_Address indiquent les paramètres numéro de port de l'enveloppe, numéro de port UDP et adresse IP appartenant au dispositif/AE DLMS/COSEM auquel les données sont à transmettre.

Le paramètre Data_Length indique la longueur du paramètre Data en octets.

Le paramètre Data contient l'APDU xDLMS à transférer à l'AL homologue.

Utilisation

La primitive UDP-DATA.request est sollicitée par l'AL DLMS/COSEM client ou serveur pour demander l'envoi d'une APDU à une AL homologue unique, ou, dans le cas de la diffusion ou multidiffusion, à des AL homologues multiples.

La réception de cette primitive de service doit inciter la sous-couche enveloppe à prédéterminer l'en-tête de l'enveloppe à l'APDU reçue, et à appeler ensuite la fonction SEND() de la sous-couche UDP avec la WDPDU formée correctement, voir en 5.3.2, en tant que DATA. La sous-couche UDP doit transmettre la WDPDU à la sous-couche enveloppe homologue comme décrit dans le document STD 0006.

5.2.2.2 UDP-DATA.indication

Fonction

Cette primitive est la primitive d'indication de service pour le service de transfert de données en mode sans connexion.

Sémantique de la primitive de service

La primitive doit fournir les paramètres comme suit:

```

UDP-DATA.indication (
    
```

```

Local_wPort,
Remote_wPort,
Local_UDP_Port,
Remote_UDP_Port,
Local_IP_Address,
Remote_IP_Address,
Data_Length,
Data
)

```

Les paramètres Local_wPort, Local_UDP_Port et Local_IP_Address indiquent les paramètres numéro de port de l'enveloppe, numéro de port UDP et adresse IP appartenant au dispositif/AE DLMS/COSEM recevant les données. Les paramètres Remote_wPort, Remote_UDP_Port et Remote_IP_Address indiquent les paramètres numéro de port de l'enveloppe, numéro de port UDP et adresse IP appartenant au dispositif/AE qui a envoyé les données.

Le paramètre Data_Length indique la longueur du paramètre Data en octets.

Le paramètre Data contient l'APDU xDLMS reçue de la part de l'AL homologue.

Utilisation

La primitive UDP-DATA.indication est générée par la TL DLMS/COSEM basée sur UDP pour indiquer à l'AL DLMS/COSEM utilisateur de service qu'une APDU a été reçue de la part de l'entité couche homologue.

La primitive est générée à la suite de la réception d'un datagramme UDP par la sous-couche UDP, si les paramètres Local_UDP_Port et Local_wPort du message reçu contiennent tous les deux des numéros de port valides, indiquant qu'une AE DLMS/COSEM est présente dans le dispositif de réception lié aux numéros de ports fournis. Si ce n'est pas le cas, le message reçu doit simplement être éliminé.

5.2.2.3 UDP-DATA.confirm

Fonction

Cette primitive est la primitive de confirmation de service facultative pour le service de transfert de données en mode sans connexion.

Sémantique de la primitive de service

La primitive doit fournir les paramètres comme suit:

```

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

```

Les paramètres Local_wPort, Remote_wPort, Local_UDP_Port, Remote_UDP_Port, Local_IP_Address et Remote_IP_Address comportent les mêmes valeurs que le service UDP-DATA.request correspondant qui est confirmé.

La valeur du paramètre Result indique si la TL DLMS/COSEM basée sur UDP a été capable d'envoyer le datagramme UDP demandé (OK) ou s'il n'en a pas été capable (NOK).

Utilisation

La primitive UDP-DATA.confirm est facultative. Si elle est mise en œuvre, elle est générée par la TL DLMS/COSEM pour confirmer à l'AL DLMS/COSEM utilisateur de service le résultat de la demande UDP-DATA.request précédente. Elle est générée localement et indique uniquement si les données dans la primitive .request ont pu être envoyées ou non. En d'autres termes, UDP-DATA.confirm avec le paramètre Result == OK signifie uniquement que les données ont été envoyées et non qu'elles ont été (ou seront) livrées avec succès à leur destinataire.

5.3 Spécification de protocole pour la couche transport DLMS/COSEM basée sur UDP

5.3.1 Généralités

Comme indiqué dans la Figure 2, la TL DLMS/COSEM basée sur UDP comprend la couche UDP Internet standard, telle que spécifiée dans la norme Internet STD 0006, et la sous-couche enveloppe légère spécifique DLMS/COSEM.

Dans ce profil de communication, la sous-couche enveloppe est une entité sans état: son unique rôle consiste à assurer l'identification de l'AE DLMS/COSEM source et destination utilisant des numéros wPort et à effectuer une conversion entre les demandes de service UDP-DATA.xxx du modèle OSI et les fonctions d'interface SEND() et RECEIVE() fournies par l'UDP standard.

Bien que ce ne soit pas nécessaire dans le profil UDP, pour que les informations de contrôle du protocole de l'enveloppe – autrement dit l'en-tête de l'enveloppe – soient identiques dans les deux TL, la sous-couche enveloppe doit également inclure les informations relatives à la longueur de données dans l'unité de données de protocole de l'enveloppe.

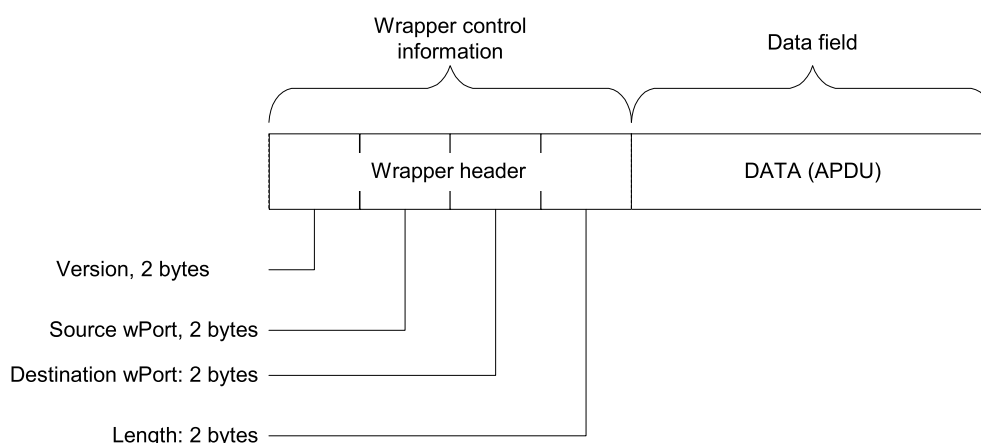
5.3.2 Unité de données de protocole de l'enveloppe (WPDU)

La WPDU contient deux parties:

- la partie en-tête de l'enveloppe, qui comprend les informations de contrôle de l'enveloppe; et
- la partie données, qui comprend le paramètre DATA – l'APDU xDLMS – de la demande de service UDP-DATA.xxx correspondante.

L'en-tête de l'enveloppe comporte quatre champs, voir Figure 4. Chaque champ est une valeur entière non signée de 16 bits.

- Version: contient la version de l'enveloppe. Sa valeur est contrôlée par la DLMS UA. La valeur actuelle est 0x0001. Noter que dans les versions ultérieures, l'en-tête de l'enveloppe peut présenter une structure différente;
- Source wPort: contient le numéro wPort identifiant l'AE DLMS/COSEM émettrice.
- Destination wPort: contient le numéro wPort identifiant l'AE DLMS/COSEM destinataire;
- Data length: indique la longueur du champ DATA de la WPDU (l'APDU xDLMS transportée).



IEC

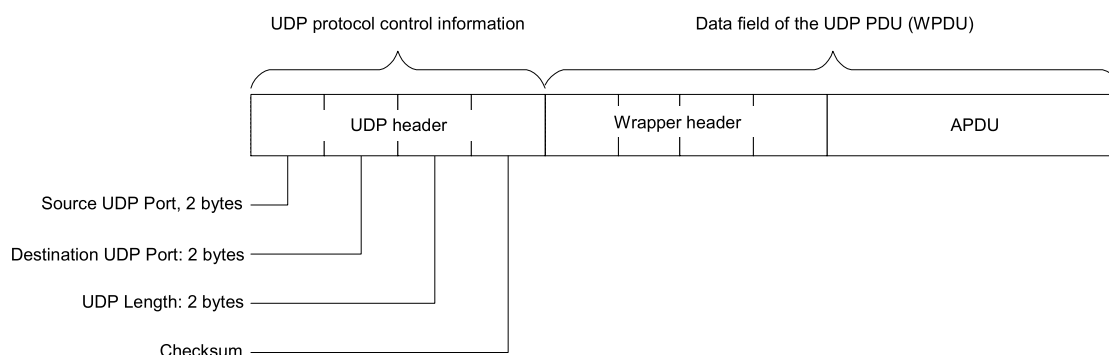
NOTE Il convient que la longueur maximale de l'APDU soit de huit octets de moins que la longueur maximale du datagramme UDP.

Anglais	Français
Wrapper control information	Information de contrôle de l'enveloppe
Data field	Champ de données
Wrapper header	En-tête de l'enveloppe
DATA (APDU)	DATA (APDU)
Version, 2 bytes	Version, 2 octets
Source wPort, 2 bytes	Source wPort, 2 octets
Destination wPort, 2 bytes	Destination wPort, 2 octets
Length: 2 bytes	Longueur: 2 octets

Figure 4 – Unité de données de protocole de l'enveloppe (WPDU)

5.3.3 Unité de données de protocole couche transport DLMS/COSEM basée sur UDP

Dans ce profil, les WPDU doivent être transmises dans les datagrammes UDP spécifiés dans la norme Internet STD 0006. Ils doivent encapsuler la WPDU, comme indiqué dans la Figure 5.



IEC

Anglais	Français
UDP protocol control information	Information de contrôle du protocole UDP
Data field of the UDP PDU (WPDU)	Champ de données de l'UDP PDU (WPDU)
UDP Header	En-tête UDP

Anglais	Français
Wrapper Header	En-tête de l'enveloppe
Source UDP Port, 2 bytes	Port UDP source, 2 octets
Destination UDP Port, 2 bytes	Port UDP destination, 2 octets
UDP Length: 2 bytes	Longueur UDP: 2 octets
Checksum	Somme de contrôle

Figure 5 – PDU (UDP-PDU) de couche transport DLMS/COSEM sans connexion, basée sur UDP

Du point de vue externe, la PDU de TL DLMS/COSEM sans connexion est un datagramme UDP ordinaire: tout élément spécifique DLMS/COSEM, y compris l'en-tête spécifique à l'enveloppe, se trouve à l'intérieur du champ de données UDP. Par conséquent, les mises en œuvre UDP standard peuvent être (ré)utilisées pour mettre en œuvre facilement cette TL.

Les ports UDP Source et Destination peuvent faire référence aux ports UDP aussi bien locaux que distants en fonction du sens du transfert de données: du point de vue du dispositif émetteur, le port UDP Source dans un datagramme correspond au paramètre de service Local_UDP_port, mais du point de vue du dispositif destinataire, le port UDP Source dans un datagramme correspond au paramètre de service Remote_UDP_Port.

Conformément à la spécification UDP, compléter les champs Source UDP Port (port UDP source) et Checksum (somme de contrôle) avec des données réelles est facultatif. Une valeur zéro, quand tous les bits sont égaux à zéro, de ces champs indique que dans le datagramme UDP concerné, le champ n'est pas utilisé. Cependant, dans le profil DLMS/COSEM_on_IP, le champ Source UDP Port doit toujours être complété à l'aide du numéro de port Source UDP.

5.3.4 Numéros de port de l'enveloppe réservés (wPorts)

Les numéros wPort réservés sont spécifiés dans le Tableau 1:

Tableau 1 – Numéros de port de l'enveloppe réservés dans la TL DLMS/COSEM basée sur UDP

Adresses réservées côté client	
	Numéro de port de l'enveloppe
No-station	0x0000
Client Management Process	0x0001
Public Client	0x0010
Open for client SAP assignment	0x02...0x0F
	0x11...0xFF
Adresses réservées côté serveur	
	Numéro de port de l'enveloppe
No-station	0x0000
Management Logical Device	0x0001
Reserved	0x0002..0x000F
Open for server SAP assignment	0x0010,,,0x007E
All-station (Broadcast)	0x007F

5.3.5 Diagramme d'états de protocole

Étant donné que la sous-couche enveloppe dans ce profil est sans état, pour toutes les autres questions relatives au protocole – diagramme d'états de protocole, etc. – les règles applicables sont telles que spécifiées dans la norme Internet STD 0006. La seule règle supplémentaire concerne l'élimination des messages inappropriés: les messages sans numéro wPort de destination valide, impliquant qu'aucune AE DLMS/COSEM n'est présente dans le dispositif destinataire lié à ce numéro wPort, doivent être éliminés par la sous-couche enveloppe.

6 Couche transport DLMS/COSEM, orientée connexion, basée sur TCP

6.1 Généralités

La TL DLMS/COSEM orientée connexion est basée sur le protocole de transport Internet orienté connexion, appelé Protocole de contrôle de transmission (TCP). TCP est un protocole fiable de bout en bout. Cette fiabilité est assurée par un "circuit virtuel" conceptuel utilisant une méthode appelée "acquittement positif avec retransmission" (PAR). Il permet la livraison de données avec acquittement, la détection d'erreur, la retransmission des données quand le temps d'attente d'acquittement est écoulé, etc. Par conséquent, il gère les lots de données perdus, retardés, dupliqués ou erronés. De plus, le TCP fournit un mécanisme de contrôle de flux efficace, ainsi qu'un fonctionnement bidirectionnel simultané.

Le TCP, en tant que protocole de transfert orienté connexion, implique trois phases: l'établissement de la connexion, l'échange de données et la déconnexion. La TL DLMS/COSEM basée sur TCP fournit donc des services du modèle OSI à l'utilisateur (aux utilisateurs) de service pour chacune des trois phases:

- pour la phase d'établissement de la connexion, le service TCP-CONNECT est fourni au processus gestionnaire de connexion TCP utilisateur de service;
- pour la phase de transfert des données, le service TCP-DATA est fourni à l'AL DLMS/COSEM utilisateur de service;
- pour la phase de fermeture de la connexion, le service TCP-DISCONNECT est fourni au processus gestionnaire de connexion TCP utilisateur de service;
- en outre, un service TCP-ABORT est fourni à l'AL DLMS/COSEM utilisateur de service.

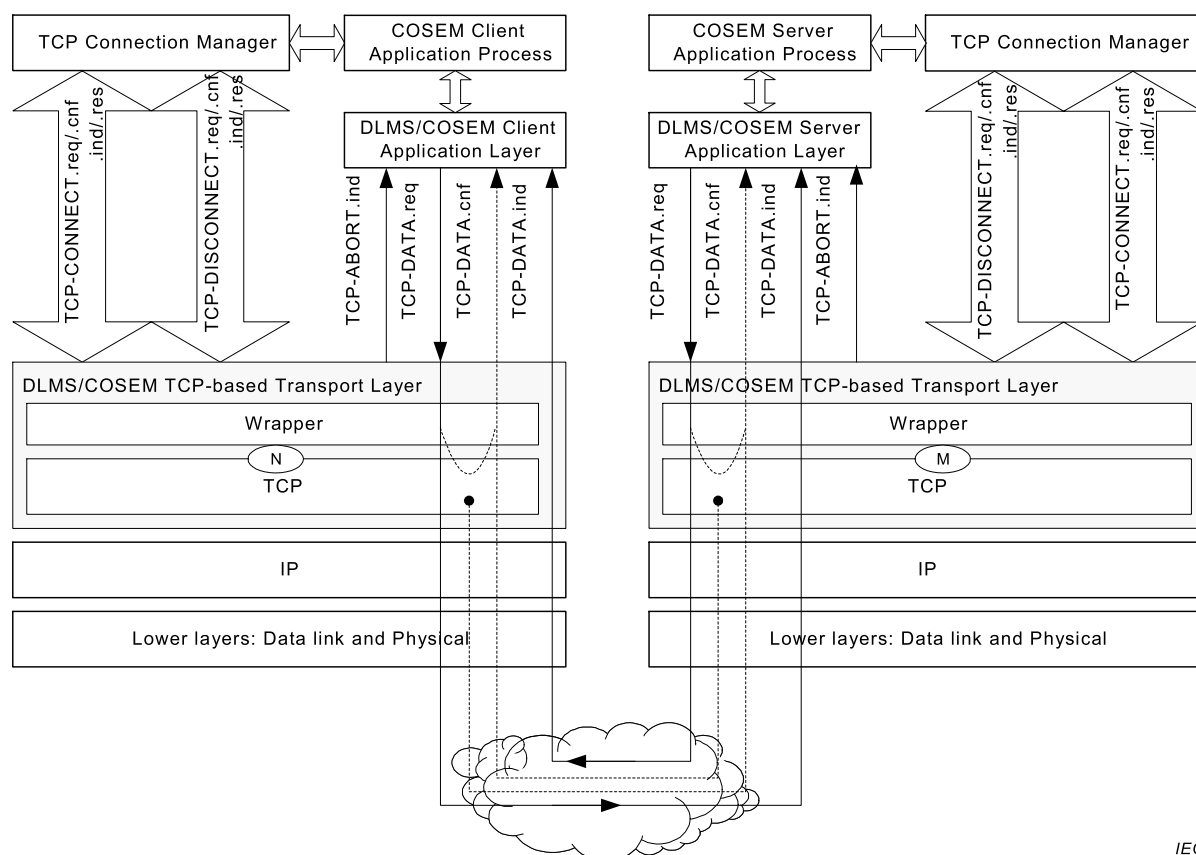
La TL DLMS/COSEM orientée connexion basée sur TCP contient la même sous-couche enveloppe que la TL DLMS/COSEM basée sur UDP. En plus de convertir les services du modèle OSI vers et depuis des appels de fonction TCP, cette enveloppe fournit des informations d'adressage et de longueur supplémentaires.

La TL DLMS/COSEM orientée connexion basée sur TCP est spécifiée en termes de services et de protocoles. La conversion entre services du modèle OSI et appels de fonction TCP est présentée à l'Annexe A.

6.2 Spécification de service pour la couche transport DLMS/COSEM basée sur TCP

6.2.1 Généralités

La TL DLMS/COSEM orientée connexion basée sur TCP fournit le même ensemble de services côté client et côté serveur, comme indiqué dans la Figure 6.



IEC

Anglais	Français
TCP Connection Manager	Gestionnaire de connexion TCP
COSEM Client Application process	Processus d'application client COSEM
COSEM Server Application process	Processus d'application serveur COSEM
DLMS/COSEM Client Application layer	Couche application client DLMS/COSEM
DLMS/COSEM Server Application layer	Couche application serveur DLMS/COSEM
DLMS/COSEM TCP-based Transport Layer	Couche transport DLMS/COSEM basée sur TCP
Wrapper	Enveloppe
Lower Layers: Data Link and Physical	Couches inférieures: Liaison de données et Physique

Figure 6 – Services de la couche transport DLMS/COSE orientée connexion basée sur TCP

Dans ce profil de communication, l'ensemble complet des primitives de service des services de gestion de connexion TCP (TCP-CONNECT et TCP-DISCONNECT) est fourni côté client comme côté serveur. Il s'agit de permettre au serveur de lancer et d'interrompre une connexion TCP.

NOTE L'établissement d'une association d'application est effectué par l'AE client.

L'utilisateur des services de gestion de connexion TCP n'est pas l'AL DLMS/COSEM, mais le processus gestionnaire de connexion TCP. La spécification de ce processus n'entre pas dans le domaine d'application de cette spécification auxiliaire. Cependant, l'AL DLMS/COSEM définit certaines exigences correspondantes. Voir 9.1 de IEC 62056-9-7:2013.

Un service COSEM-ABORT supplémentaire est fourni pour indiquer à l'AL DLMS/COSEM que la connexion TCP concernée est interrompue ou désactivée.

Comme dans la TL DLMS/COSEM basée sur UDP, la primitive de service TCP-DATA.confirm est également facultative. Cependant, le service TCP-DATA.request peut être confirmé soit localement, soit à distance.

6.2.2 Service TCP-CONNECT

6.2.2.1 TCP-CONNECT.request

Fonction

Cette primitive est la primitive de demande de service pour le service d'établissement d'une connexion.

Sémantique de la primitive de service

La primitive doit fournir les paramètres comme suit:

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

Les paramètres Local_TCP_Port et Remote_TCP_Port identifient respectivement les ports TCP local et distant. Les paramètres Local_IP_Address et Remote_IP_Address indiquent l'adresse IP du dispositif physique demandant la connexion TCP et du dispositif physique cible, avec lequel la connexion TCP demandée est à établir.

Utilisation

Le processus gestionnaire de connexion TCP utilisateur de service demande la primitive TCP-CONNECT.request pour établir une connexion avec la TL DLMS/COSEM homologue basée sur TCP.

6.2.2.2 TCP-CONNECT.indication

Fonction

Cette primitive est la primitive d'indication de service pour le service d'établissement d'une connexion.

Sémantique de la primitive de service

La primitive doit fournir les paramètres comme suit:

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

Les paramètres Local_TCP_Port et Remote_TCP_Port indiquent les deux ports TCP entre lesquels la connexion TCP demandée est à établir. Les paramètres Local_IP_Address et Remote_IP_Address indiquent les adresses IP des deux dispositifs participant à la connexion TCP.

Utilisation

La primitive TCP-CONNECT.indication est générée par la TL DLMS/COSEM basée sur TCP à la suite de la réception d'un paquet TCP, indiquant au processus gestionnaire de connexion TCP qu'un dispositif distant demande une nouvelle connexion TCP.

6.2.2.3 TCP-CONNECT.response

Fonction

Cette primitive est la primitive de réponse de service pour le service d'établissement d'une connexion.

Sémantique de la primitive de service

La primitive doit fournir les paramètres comme suit:

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

Les paramètres Local_TCP_Port et Remote_TCP_Port indiquent les deux ports TCP entre lesquels la connexion est établie. Les paramètres Local_IP_Address et Remote_IP_Address indiquent les adresses IP des deux dispositifs physiques participant à la connexion TCP.

Le paramètre Result indique que le gestionnaire de connexion TCP utilisateur de service a accepté la demande de connexion TCP. Sa valeur est toujours SUCCESS.

Utilisation

Le processus gestionnaire de connexion TCP demande la primitive TCP-CONNECT.response pour indiquer à la TL DLMS/COSEM basée sur TCP si la demande de connexion TCP précédente a été acceptée. Le gestionnaire de connexion TCP ne peut pas rejeter une demande de connexion.

6.2.2.4 TCP-CONNECT.confirm

Fonction

Cette primitive est la primitive de confirmation de service pour le service d'établissement d'une connexion.

Sémantique de la primitive de service

La primitive doit fournir les paramètres comme suit:

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

Les paramètres Local_TCP_Port et Remote_TCP_Port indiquent les deux ports TCP entre lesquels la connexion est établie. Les paramètres Local_IP_Address et Remote_IP_Address indiquent les adresses IP des deux dispositifs physiques participant à cette connexion TCP.

Le paramètre Result indique si la connexion TCP demandée est établie ou non. A noter que cette primitive de service correspond normalement au résultat d'une confirmation distante, et dans la mesure où une demande de connexion TCP ne peut pas être rejetée, le paramètre Result doit toujours indiquer SUCCESS.

Cependant, le paramètre Result peut également indiquer FAILURE, lorsqu'il est confirmé localement. Dans ce cas, le paramètre Reason_of_Failure indique la raison de l'échec.

Utilisation

La primitive TCP-CONNECT.confirm est générée par la TL DLMS/COSEM basée sur TCP pour indiquer au processus gestionnaire de connexion TCP utilisateur de service le résultat d'une demande de service TCP-CONNECT.request reçue précédemment.

6.2.3 Service TCP-DISCONNECT

6.2.3.1 TCP-DISCONNECT.request

Fonction

Cette primitive est la primitive de demande de service pour le service d'interruption d'une connexion.

Sémantique de la primitive de service

La primitive doit fournir les paramètres comme suit:

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

Les paramètres de service sont les identificateurs de la connexion TCP à libérer. Les paramètres Local_TCP_Port et Local_IP_Address désignent le port TCP et l'adresse IP locaux du dispositif et de l'application demandant la connexion, les paramètres Remote_IP_Address et Remote_TCP_Port font référence au dispositif et à l'application distants.

Utilisation

La primitive TCP-DISCONNECT.request est demandée par le processus gestionnaire de connexion TCP utilisateur de service pour demander l'interruption d'une connexion TCP existante.

6.2.3.2 TCP-DISCONNECT.indication

Fonction

Cette primitive est la primitive d'indication de service pour le service d'interruption d'une connexion.

Sémantique de la primitive de service

La primitive doit fournir les paramètres comme suit:

```
TCP-DISCONNECT.indication (
                                Local_TCP_Port,
```

```
Remote_TCP_Port,
Local_IP_Address,
Remote_IP_Address,
Reason
```

)

Les paramètres Local_TCP_Port, Remote_TCP_Port, Local_IP_Address, Remote_IP_Address identifient la connexion TCP, soit dont la libération a été demandée par le dispositif homologue, soit qui a été abandonnée.

Le paramètre Reason indique si le service est sollicité parce que le dispositif homologue a demandé une déconnexion TCP (Reason == REMOTE_REQ), ou s'il est généré localement à la suite de la détection d'un événement devant entraîner l'interruption de la connexion TCP (Reason == ABORT).

NOTE La couche transport DLMS/COSEM peut fournir des informations plus détaillées sur les raisons d'ABORT par l'intermédiaire des services de gestion de couche. Cependant, ces services ne relèvent pas du domaine d'application de la présente norme internationale.

Utilisation

La primitive TCP-DISCONNECT.indication est générée par la TL DLMS/COSEM basée sur TCP pour indiquer au processus gestionnaire de connexion TCP utilisateur de service que l'entité homologue a demandé l'interruption d'une connexion TCP existante. La même primitive est également utilisée pour indiquer si la TL a détecté une interruption non demandée d'une connexion TCP existante (par exemple, lorsque la connexion physique est coupée).

6.2.3.3 TCP-DISCONNECT.response

Fonction

Cette primitive est la primitive de réponse de service pour le service d'interruption d'une connexion.

Sémantique de la primitive de service

La primitive doit fournir les paramètres comme suit:

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

Les paramètres Local_TCP_Port et Remote_TCP_Port identifient les deux ports TCP entre lesquels la connexion TCP est à interrompre. Les paramètres Local_IP_Address et Remote_IP_Address indiquent les adresses IP des deux dispositifs physiques participant à la connexion TCP à interrompre.

Le paramètre Result indique que le processus gestionnaire de connexion TCP utilisateur de service a accepté la déconnexion de la connexion TCP concernée. La valeur de ce paramètre est toujours SUCCESS.

Utilisation

La primitive TCP-DISCONNECT.response est demandée par le processus gestionnaire de connexion TCP pour indiquer à la TL DLMS/COSEM basée sur TCP si la déconnexion TCP demandée précédemment est acceptée. À noter que le processus gestionnaire de connexion

TCP ne peut pas rejeter la demande de déconnexion. Cette primitive de service est sollicitée uniquement si le service TCP-DISCONNECT.indication correspondant a indiqué une demande de déconnexion initiée à distance (Reason == REMOTE_REQ).

6.2.3.4 TCP-DISCONNECT.confirm

Fonction

Cette primitive est la primitive de confirmation de service pour le service d'interruption d'une connexion.

Sémantique de la primitive de service

La primitive doit fournir les paramètres comme suit:

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

Les paramètres Local_TCP_Port et Remote_TCP_Port identifient les deux ports TCP entre lesquels la connexion TCP est à interrompre. Les paramètres Local_IP_Address et Remote_IP_Address indiquent les adresses IP des deux dispositifs physiques participant à la connexion TCP à interrompre.

Le paramètre Result indique si l'interruption de la connexion TCP concernée a réussi ou non. Normalement, cette primitive de service est sollicitée en tant que résultat d'une confirmation distante, et dans la mesure où une demande de déconnexion TCP ne peut pas être rejetée, la valeur du paramètre Result est toujours SUCCESS.

Cependant, le paramètre Result peut également indiquer FAILURE, lorsqu'il est confirmé localement. Dans ce cas, le paramètre Reason_of_Failure indique la raison de l'échec.

Utilisation

La primitive TCP-DISCONNECT.confirm est demandée par la TL DLMS/COSEM basée sur TCP pour confirmer au gestionnaire de connexion TCP utilisateur de service le résultat d'une demande de service TCP-DISCONNECT.request précédente.

6.2.4 Service TCP-ABORT

6.2.4.1 TCP-ABORT.indication

Fonction

Cette primitive est la primitive d'indication de service pour le service d'interruption d'une connexion.

Sémantique de la primitive de service

La primitive doit fournir les paramètres comme suit:

```
TCP-ABORT.indication (
                        Local_TCP_Port,
                        Remote_TCP_Port,
```

```

Local_IP_Address,
Remote_IP_Address,
Reason
)

```

Les paramètres Local_TCP_Port et Remote_TCP_Port identifient les deux ports TCP entre lesquels la connexion a été abandonnée. Les paramètres Local_IP_Address et Remote_IP_Address indiquent les adresses IP des deux dispositifs physiques ayant participé à la connexion TCP abandonnée.

Le paramètre Reason indique la raison de l'abandon TCP. Ce paramètre est facultatif.

Utilisation

La primitive TCP-ABORT.indication est générée par la TL DLMS/COSEM basée sur TCP pour indiquer à l'AL DLMS/COSEM utilisateur de service que la connexion TCP concernée subit une interruption non sollicitée.

Quand cette indication est reçue, l'AL DLMS/COSEM doit libérer toutes les AA établies utilisant cette connexion TCP, et doit l'indiquer à l'AP COSEM utilisant la primitive de service COSEM-ABORT.indication. Voir également 6.4 de l'IEC 62056-5-3:2013.

6.2.5 Service TCP-DATA

6.2.5.1 TCP-DATA.request

Fonction

Cette primitive est la primitive de demande de service pour le service de transfert de données en mode connexion.

Sémantique de la primitive de service

La primitive doit fournir les paramètres comme suit:

```

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

```

Les paramètres Local_wPort, Local_TCP_Port et Local_IP_Address indiquent les paramètres numéro de port de l'enveloppe, numéro de port TCP et adresse IP du dispositif/AE DLMS/COSEM demandant l'envoi des données. Les paramètres Remote_wPort, Remote_TCP_Port et Remote_IP_Address indiquent les paramètres numéro de port de l'enveloppe, numéro de port TCP et adresse IP appartenant au dispositif/AE DLMS/COSEM auquel les données sont à transmettre.

Le paramètre Data_Length indique la longueur du paramètre Data en octets.

Le paramètre Data contient l'APDU xDLMS à transférer à l'AL homologue.

Utilisation

La primitive TCP-DATA.request est demandée par l'AL DLMS/COSEM client ou serveur pour demander l'envoi d'une APDU à une application homologue unique.

La réception de cette primitive doit faire en sorte que la sous-couche enveloppe prédétermine les champs spécifiques de l'enveloppe (Local_wPort, Remote_wPort et Data_Length) à l'APDU xDLMS reçue, puis appelle la fonction SEND() de la sous-couche TCP avec la WPDU, correctement formée, voir 5.3.2, en tant que DATA. La sous-couche TCP doit transmettre la WPDU à la sous-couche TCP homologue tel que décrit dans le document STD 0007.

6.2.5.2 TCP-DATA.indication

Fonction

Cette primitive est la primitive d'indication de service pour le service de transfert de données en mode connexion.

Sémantique de la primitive de service

La primitive doit fournir les paramètres comme suit:

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

Les paramètres Local_wPort, Local_TCP_Port et Local_IP_Address indiquent les paramètres numéro de port de l'enveloppe, numéro de port TCP et adresse IP appartenant au dispositif/AE DLMS/COSEM recevant les données. Les paramètres Remote_wPort, Remote_TCP_Port et Remote_IP_Address indiquent les paramètres numéro de port de l'enveloppe, numéro de port TCP et adresse IP appartenant au dispositif/AE DLMS/COSEM qui a envoyé les données.

Le paramètre Data_Length indique la longueur du paramètre Data en octets.

Le paramètre Data contient l'APDU xDLMS reçue de la part de l'AL homologue.

Utilisation

La primitive TCP-DATA.indication est générée par la TL DLMS/COSEM pour indiquer à l'AL DLMS/COSEM utilisateur de service qu'une APDU xDLMS a été reçue de la part d'un dispositif distant. Elle est générée à la suite de la réception d'une APDU complète (en un paquet TCP ou plus) par la TL DLMS/COSEM basée sur TCP, si les paramètres Local_TCP_Port et Local_wPort dans le(s) paquet(s) TCP transportant l'APDU contiennent tous deux des numéros de port valides, ce qui signifie qu'une AE DLMS/COSEM est présente dans le dispositif destinataire lié aux numéros de ports donnés. Si ce n'est pas le cas, le message reçu doit simplement être éliminé.

6.2.5.3 TCP-DATA.confirm

Fonction

Cette primitive est la primitive de confirmation de service facultative pour le service de transfert de données en mode connexion.

Sémantique de la primitive de service

La primitive doit fournir les paramètres comme suit:

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

Les paramètres Local_wPort, Remote_wPort, Local_TCP_Port, Remote_TCP_Port, Local_IP_Address et Remote_IP_Address comportent les mêmes valeurs que le service TCP-DATA.request correspondant confirmé.

Le paramètre Confirmation_Type indique si le service de confirmation constitue une confirmation LOCAL ou REMOTE.

La valeur du paramètre Result indique le résultat du service TCP-DATA.request précédent. Sa valeur est soit OK soit NOK, mais sa signification dépend de la mise en œuvre de la primitive .confirm. Voir 6.3.5.4.

Utilisation

La primitive TCP-DATA.confirm est facultative. Si elle est mise en œuvre, elle est générée par la TL DLMS/COSEM pour confirmer à l'AL DLMS/COSEM utilisateur de service le résultat de l'exécution de la primitive .request précédente.

6.3 Spécification de protocole pour la couche transport DLMS/COSEM basée sur TCP

6.3.1 Généralités

Comme indiqué à la Figure 2, la TL DLMS/COSEM orientée connexion (CO) basée sur TCP comprend la couche TCP Internet standard telle que spécifiée dans le document STD 0007 et la sous-couche enveloppe spécifique DLMS/COSEM.

Dans cette TL basée sur TCP, la sous-couche enveloppe est plus complexe que dans la TL basée sur UDP. D'une part, à l'instar de la TL basée sur UDP, son rôle principal consiste également à assurer l'identification des AE DLMS/COSEM source et destination utilisant les numéros wPort, et à convertir les primitives de service TCP-DATA du modèle OSI vers et depuis les fonctions d'interface SEND() et RECEIVE() fournies par le TCP standard. D'autre part, la sous-couche enveloppe dans la TL basée sur TCP a également pour fonction de permettre aux AL DLMS/COSEM utilisateur de service d'échanger des APDU complètes.

Le TCP est un protocole de "diffusion en flux" ce qui signifie qu'il ne préserve pas les frontières des données. Sans entrer dans les détails (voir Article A.4 pour de plus amples informations), ceci signifie que les appels de fonction SEND() et RECEIVE() de la sous-couche TCP reviennent avec succès même si le nombre d'octets réellement envoyés/reçus est inférieur au nombre d'octets requis pour l'envoi/la réception. Il incombe à la sous-couche enveloppe de connaître la quantité de données à envoyer/recevoir, de conserver une trace de la quantité de données réellement envoyées/reçues, et de renouveler l'opération jusqu'à ce que l'APDU complète soit transmise.

Par conséquent, la sous-couche enveloppe dans la TL DLMS/COSEM basée sur TCP n'est pas une entité sans état: elle assure le suivi décrit ci-dessus, en renouvelant la procédure

pour rendre la nature “diffusion en flux” du TCP transparente à l’AL DLMS/COSEM utilisateur de service.

6.3.2 Unité de données de protocole de l’enveloppe (WPDU)

L’unité de données de protocole de l’enveloppe est telle que spécifiée en 5.3.2.

6.3.3 Unité de données de protocole couche transport DLMS/COSEM basée sur TCP

Les WPDU sont transmises dans un ou plusieurs paquets TCP. Le paquet TCP est spécifié dans le document STD 0007 et encapsule une partie de la WPDU dans son champ Data, comme indiqué dans la Figure 7. La raison de n’avoir qu’une partie de la WPDU dans un paquet TCP est la nature “diffusion en flux” du TCP déjà mentionné.

0	4	8	10	16	24	31
Port TCP source				Port TCP destination		
Numéro de séquence						
Numéro d'acquittement						
Décalage de données	Réservé		Drapeaux	Fenêtre		
Somme de contrôle				Pointeur urgent		
Options TCP					Remplissage	
Data (partie de la WPDU)						
Data (partie de la WPDU)						

IEC

Figure 7 – Format de paquet TCP

Du point de vue externe, la PDU de TL DLMS/COSEM basée sur TCP est un paquet TCP ordinaire: tout élément spécifique DLMS/COSEM, y compris l’en-tête spécifique à l’enveloppe dans le premier paquet TCP, se trouve à l’intérieur du champ Data du paquet.

Les ports TCP source et destination peuvent faire référence aux ports TCP aussi bien locaux que distants en fonction du sens du transfert de données (c’est-à-dire que du point de vue du dispositif émetteur, le port TCP source dans un paquet TCP correspond au paramètre de service Local_TCP_port, mais du point de vue du dispositif destinataire, le port TCP source d’un datagramme correspond au paramètre de service Remote_TCP_Port).

6.3.4 Numéros de port de l’enveloppe réservés

Les numéros wPort réservés sont spécifiés dans le Tableau 1.

6.3.5 Définition des procédures

6.3.5.1 Connexion TCP

L’établissement d’une connexion TCP est initié par la demande de service TCP-CONNECT.request. Bien que ce service – comme tous les services de TL DLMS/COSEM – soit fourni à l’entité utilisateur de service par la sous-couche enveloppe, la connexion TCP est établie entre les deux sous-couches TCP (locale et distante). Le rôle de l’enveloppe dans cette procédure consiste uniquement à convertir les primitives de service TCP-CONNECT (.request, .indication, .response et .confirm) vers et depuis les appels de fonction TCP.

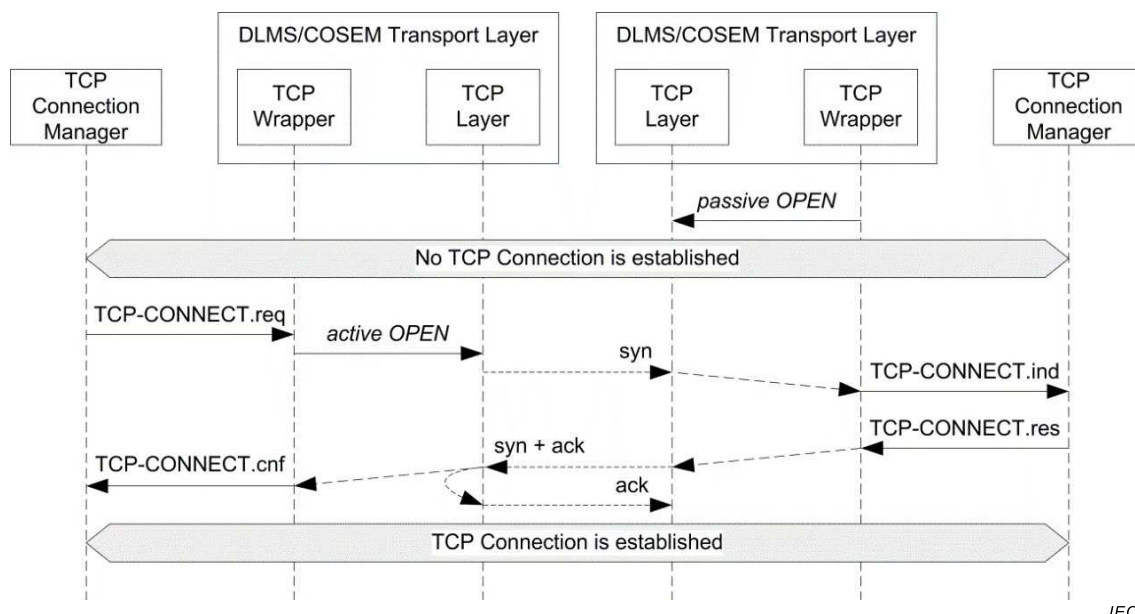
Du point de vue de l’utilisateur de service, seules les primitives de service TCP-CONNECT sont visibles: c’est pourquoi l’établissement de la connexion TCP se produit comme indiqué dans la Figure 8.

La connexion TCP est établie à l'aide d'un mécanisme triple prise de contact, comme décrit dans le document STD 0007. Ce mécanisme requiert l'échange des trois messages indiqué ci-dessus et garantit à la fois que chacun des deux côtés sait que l'autre est prêt à transmettre et que les deux côtés sont synchronisés: les numéros de séquence initiaux sont décidés en commun.

Les processus gestionnaire de connexion TCP côté client et côté serveur sont autorisés à initier la connexion TCP. Pour établir la connexion, l'un d'eux joue le rôle d'initiateur, et l'autre celui de répondeur.

Afin d'être capable de répondre, le répondeur est tenu d'effectuer une ouverture «passive» avant de recevoir le premier paquet SYN. Pour ce faire, il est tenu de contacter le système d'exploitation local (OS) pour indiquer qu'il est prêt à accepter les demandes de connexion entrantes. En réponse à cette prise de contact, l'OS attribue un numéro de port TCP à ce point final de la connexion et réserve les ressources nécessaires pour une connexion future – mais aucun message n'est envoyé.

NOTE Dans le cas de la couche transport DLMS/COSEM, la mise en œuvre force l'OS à assigner le numéro de port TCP/UDP requis au point final local de la connexion.



IEC

Anglais	Français
TCP Connection Manager	Gestionnaire de connexion TCP
TCP Wrapper	Enveloppe TCP
TCP Layer	Couche TCP
DLMS/COSEM Transport Layer	Couche transport DLMS/COSEM
No TCP connection is established	Aucune connexion TCP n'est établie
TCP connection is established	La connexion TCP est établie

Figure 8 – Établissement de la connexion TCP

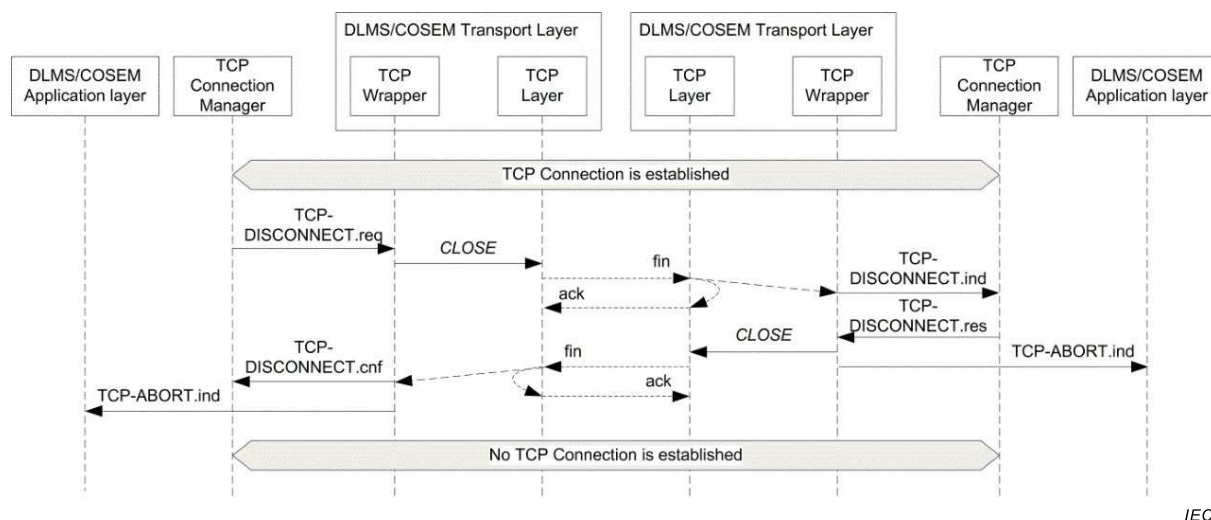
Dans le cas de la TL DLMS/COSEM basée sur TCP, la sous-couche enveloppe initie cette ouverture passive de manière autonome au cours de l'initialisation du système. Autrement dit, comme cette ouverture passive relève de la responsabilité de la sous-couche enveloppe, aucun service n'est fourni à une entité externe pour initier l'ouverture passive.

Étant donné que les processus gestionnaire de connexion TCP côté client comme côté serveur sont autorisés à jouer le rôle d'application «répondeur», les TL des deux côtés doivent exécuter une ouverture passive au cours de l'initialisation du système.

Plus de détails concernant l'établissement de la connexion TCP sont fournis à l'Article A.1.

6.3.5.2 Interruption de TCP

Le TCP est déconnecté à l'aide du service TCP-DISCONNECT, comme indiqué dans la Figure 9.



Anglais	Français
DLMS/COSEM Application Layer	Couche application DLMS/COSEM
TCP Connection Manager	Gestionnaire de connexion TCP
TCP Wrapper	Enveloppe TCP
TCP Layer	Couche TCP
DLMS/COSEM Transport Layer	Couche transport DLMS/COSEM
TCP connection is established	La connexion TCP est établie
No TCP connection is established	Aucune connexion TCP n'est établie

Figure 9 – Interruption de TCP

La procédure peut être initiée par le processus gestionnaire de connexion TCP soit côté client soit côté serveur, en sollicitant la primitive TCP-DISCONNECT.request. Cette demande est transformée par l'«enveloppe» en appel de fonction CLOSE () vers l'interface TCP.

Le TCP envoie un segment FIN, qui est acquitté par le TCP homologue.

NOTE Le TCP utilise une triple prise de contact améliorée pour libérer une connexion, et garantir qu'une duplication ou un retard éventuel – introduit par la couche IP non fiable – ne pose pas de problème. De plus amples informations concernant cette procédure sont disponibles dans le document STD 0007.

Au même moment, à travers l'enveloppe, la primitive TCP-DISCONNECT.indication est générée, pour informer le gestionnaire de connexion TCP utilisateur que la connexion est en cours de fermeture. Pour suspendre progressivement la connexion, le gestionnaire de connexion répond au moyen d'une primitive TCP-DISCONNECT.response. L'enveloppe TCP appelle la fonction CLOSE et le TCP envoie son segment FIN. Au même moment, l'enveloppe TCP indique la fermeture de la connexion TCP à l'AL DLMS/COSEM à l'aide de la primitive TCP-ABORT.indication.

Du côté de la demande, le TCP envoie un acquittement et à la réception de cet acquittement par le protocole homologue, la connexion TCP est supprimée. Au même moment, l'enveloppe génère la primitive TCP-DISCONNECT.confirm pour informer le processus gestionnaire de connexion que la demande de déconnexion a été acceptée. Comme dans le cas du protocole homologue, la déconnexion TCP est également indiquée à l'AL DLMS/COSEM à l'aide de la primitive COSEM-ABORT.indication.

De plus amples informations concernant la déconnexion TCP sont fournies à l'Article A.2.

6.3.5.3 Abandon de connexion TCP

La TL DLMS/COSEM basée sur TCP indique à l'AL DLMS/COSEM la rupture ou la déconnexion de la connexion TCP qui le supporte, à l'aide de la primitive TCP-ABORT.indication. À noter qu'il s'agit de l'unique service de gestion de connexion TCP fourni à l'AL DLMS/COSEM.

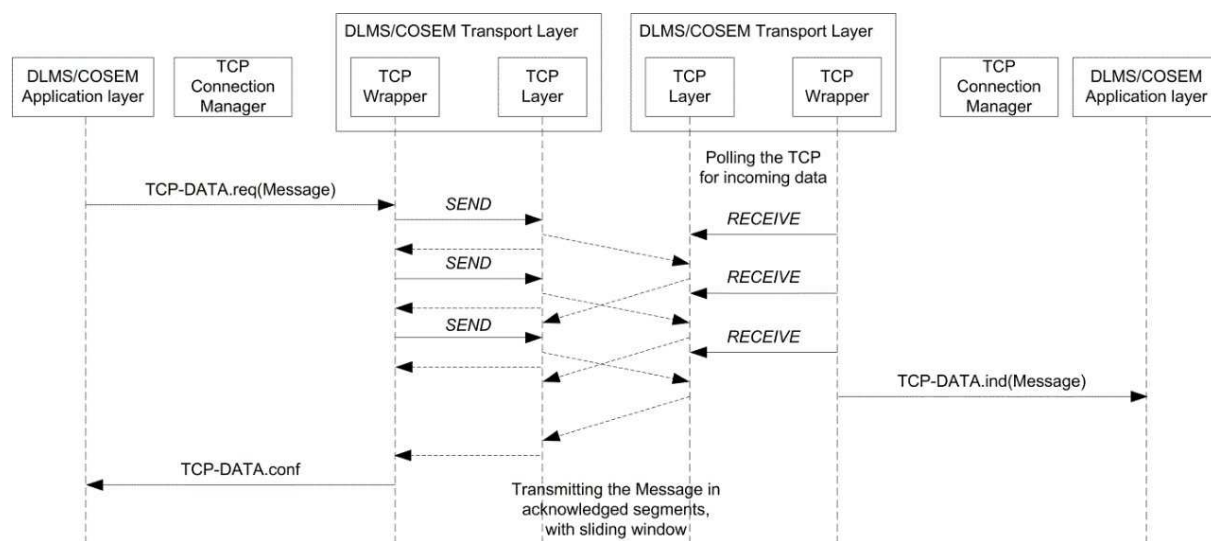
Le service est sollicité soit lors de l'interruption de la connexion TCP par le processus gestionnaire de connexion TCP – en cas de déconnexion progressive – soit lorsque la déconnexion TCP se produit de manière non sollicitée, par exemple quand la sous-couche TCP détecte une erreur insoluble ou quand la connexion physique est coupée.

L'objectif de ce service est d'informer l'AL DLMS/COSEM de l'interruption de la connexion TCP, de sorte qu'elle puisse libérer toutes les AA en cours.

6.3.5.4 Transfert de données utilisant le service TCP-DATA

Un transfert de données fiable peut être exécuté par l'intermédiaire de la connexion TCP, dès lors que cette dernière est établie. Bien que la fourniture de ce transfert de données fiable représente une opération relativement complexe impliquant des mécanismes de fiabilité tels qu'un Acquittement Positif avec Retransmission (PAR) ou un contrôle de flux avec fenêtres dynamiques (fourni par le TCP et spécifié dans le document STD 0007), la couche TL DLMS/COSEM basée sur TCP fournit uniquement un service de transfert de données, le service TCP-DATA, comme indiqué dans la Figure 10.

L'utilisation du service TCP-DATA est identique côté client et côté serveur.



IEC

Anglais	Français
DLMS/COSEM Application Layer	Couche application DLMS/COSEM
DLMS/COSEM Transport Layer	Couche transport DLMS/COSEM

Anglais	Français
TCP connection manager	Gestionnaire de connexion TCP
TCP Wrapper	Enveloppe TCP
TCP Layer	Couche TCP
Polling the TCP for incoming data	Interrogation du TCP pour les données entrantes
Transmitting the Message in acknowledged segments, with sliding window	Transmission du Message en segments avec acquittement, à l'aide de fenêtre dynamique

Figure 10 – Transfert de données utilisant la couche transport DLMS/COSEM basée sur TCP

La primitive facultative TCP-DATA.confirm indique le résultat de la primitive TCP-DATA.request sollicitée précédemment, qui est soit OK soit NOK. Cependant, la signification de ce résultat dépend de la mise en œuvre. Quand la primitive .confirm est mise en œuvre en tant que confirmation locale, le résultat indique si la TL DLMS/COSEM a été capable de mettre en mémoire tampon pour envoi ou d'envoyer l'APDU ou non. Lorsqu'elle est mise en œuvre en tant que confirmation distante, le résultat indique si l'APDU a été livrée avec succès au destinataire ou non.

Comme indiqué dans la Figure 10, le message (une WPDU) peut être transporté (envoyé/reçu) dans plusieurs paquets TCP, en raison de l'envoi par le TCP d'un flux d'octets, sans préserver les frontières des données. Il incombe à la sous-couche enveloppe de cacher cette propriété de la sous-couche TCP à l'AL DLMS/COSEM utilisateur de service. L'enveloppe côté émetteur conserve une trace de la quantité de données envoyées avec un appel de fonction SEND() et renouvelle l'opération jusqu'à ce que la WPDU complète soit envoyée. L'enveloppe côté destinataire continue à recevoir les paquets TCP entrants jusqu'à ce qu'une WPDU complète soit reçue. Pour plus de détails, voir Article A.4.

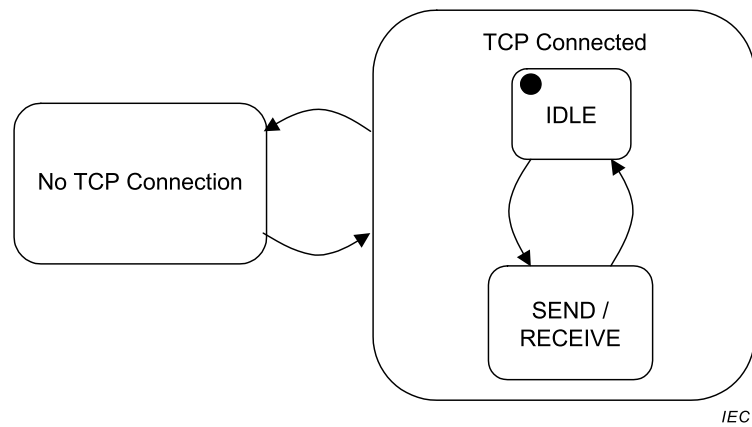
6.3.5.5 Diagramme de transition d'états de haut niveau de la sous-couche enveloppe

Le diagramme d'états de haut niveau de la sous-couche enveloppe est indiqué dans la Figure 11.

Dans les deux macroétats (No TCP Connection et TCP Connected), l'enveloppe continue d'interroger la couche TCP sur son statut de connexion et transite dans l'autre macroétat si le statut a changé.

L'enveloppe entre toujours dans le sous-état IDLE de l'état TCP Connected et transite vers l'état composite SEND/RECEIVE soit sur un paramètre TCP-DATA.request, soit à la réception d'un paquet TCP. Dans cet état, l'enveloppe envoie et/ou reçoit les WPDU, comme décrit à l'Annexe A.

NOTE Un TCP au-dessus d'une pile de protocoles de couche bidirectionnelle simultanée peut envoyer et recevoir des données simultanément.



Anglais	Français
No TCP Connection	Pas de connexion TCP
TCP Connected	TCP Connecté
IDLE	IDLE
SEND/RECEIVE	SEND/RECEIVE

Figure 11 – Diagramme de transition d'états de haut niveau de la sous-couche enveloppe

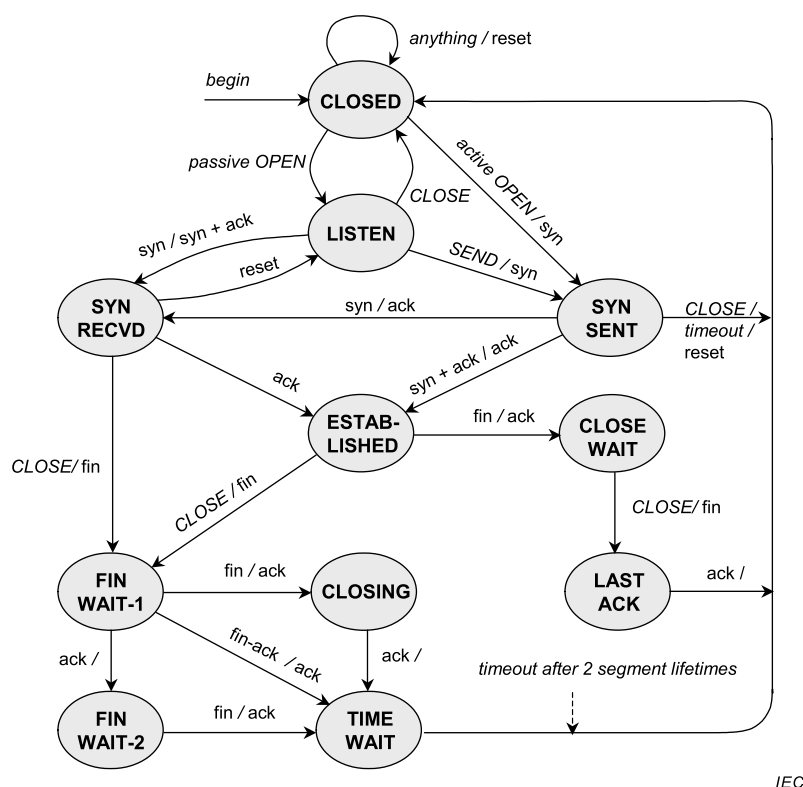
Annexe A (informative)

Conversion de services de TL du modèle OSI vers et depuis des appels de fonction TCP type RFC

A.1 Établissement d'une connexion de couche transport et TCP

Comme spécifié dans le document STD 0007 une connexion TCP est établie en sollicitant la fonction OPEN. Cette fonction peut être sollicitée de manière active ou passive.

Conformément au diagramme d'états de connexion TCP (voir Figure A.1), une fonction passive OPEN prend le dispositif appelant à l'état LISTEN, et attend une demande de connexion de tout TCP ou port distant.



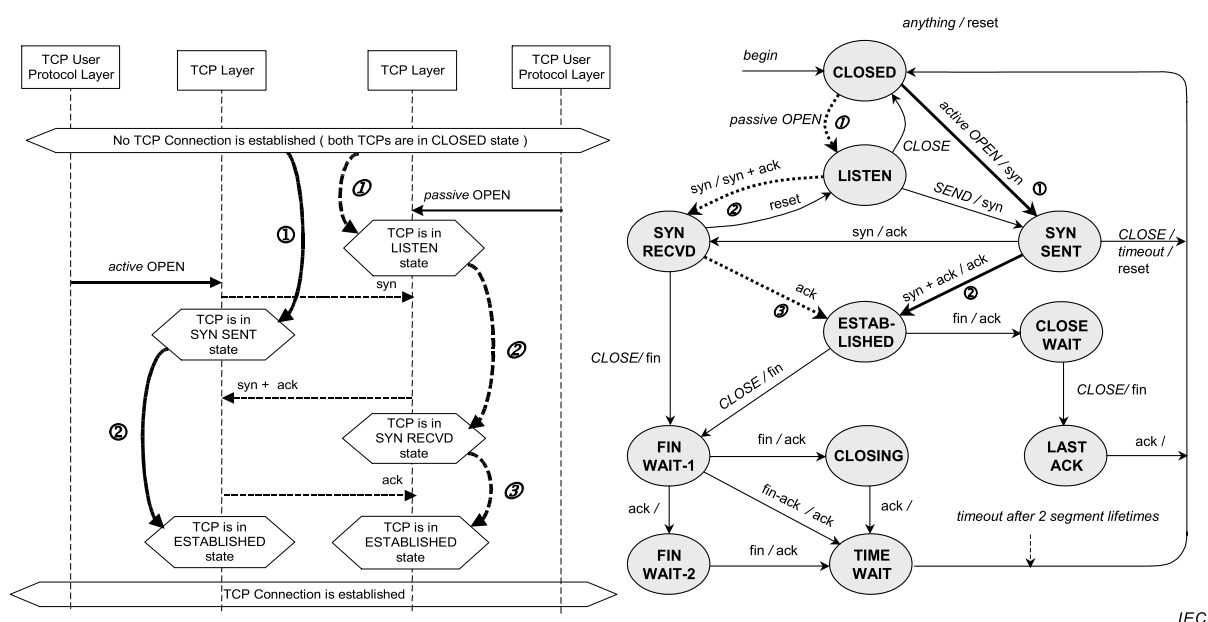
Anglais	Français
Timeout after 2 segment lifetimes	Temporisation après 2 durées de vie de segment

Figure A.1 – Diagramme d'états de connexion TCP

Un appel active OPEN fait établir au TCP la connexion avec un TCP distant.

L'établissement d'une connexion TCP est effectué à l'aide de la procédure dite «Triple prise de contact», qui est initiée par un TCP appelant une fonction active OPEN et à laquelle répond un autre TCP, celui qui a déjà appelé une fonction passive OPEN et se trouve donc en état LISTEN.

La séquence de message et les transitions d'états correspondant à cet échange de messages pour la procédure «triple prise de contact» sont indiquées à la Figure A.2.



NOTE Dans le cas de la couche transport DLMS/COSEM, la couche de protocole utilisateur de TCP est la sous-couche enveloppe.

Anglais	Français
TCP User Protocol Layer	Couche de protocole utilisateur de TCP
TCP Layer	Couche TCP
No TCP connection is established (both TCPs are in CLOSED state)	Aucune connexion TCP n'est établie (les deux TCP sont à l'état CLOSED)
TCP is in LISTEN state	Le TCP est à l'état LISTEN
TCP is in SYN SENT state	Le TCP est à l'état SYN SENT
TCP is in SYN RECVD state	Le TCP est à l'état SYN RECVD
TCP is in ESTABLISHED state	Le TCP est à l'état ESTABLISHED
TCP connection is established	La connexion TCP est établie
Timeout after 2 segment lifetimes	Temporisation après 2 durées de vie de segment

Figure A.2 – MSC et transitions d'états pour l'établissement d'une connexion de couche transport et TCP

Ce processus, constitué de trois messages, établit la connexion TCP et «synchronise» les numéros de séquence initiaux des deux côtés. Ce mécanisme a été soigneusement conçu pour garantir que chacun des deux côtés est prêt à transmettre des données et sait que l'autre côté l'est également. À noter que cette procédure fonctionne également si deux TCP l'initient simultanément.

NOTE Les numéros de séquence font partie du paquet TCP, et sont fondamentaux pour un transfert de données fiable. Pour plus de détails sur les numéros de séquence (ou toute autre question relative au TCP), se référer à STD 0007.

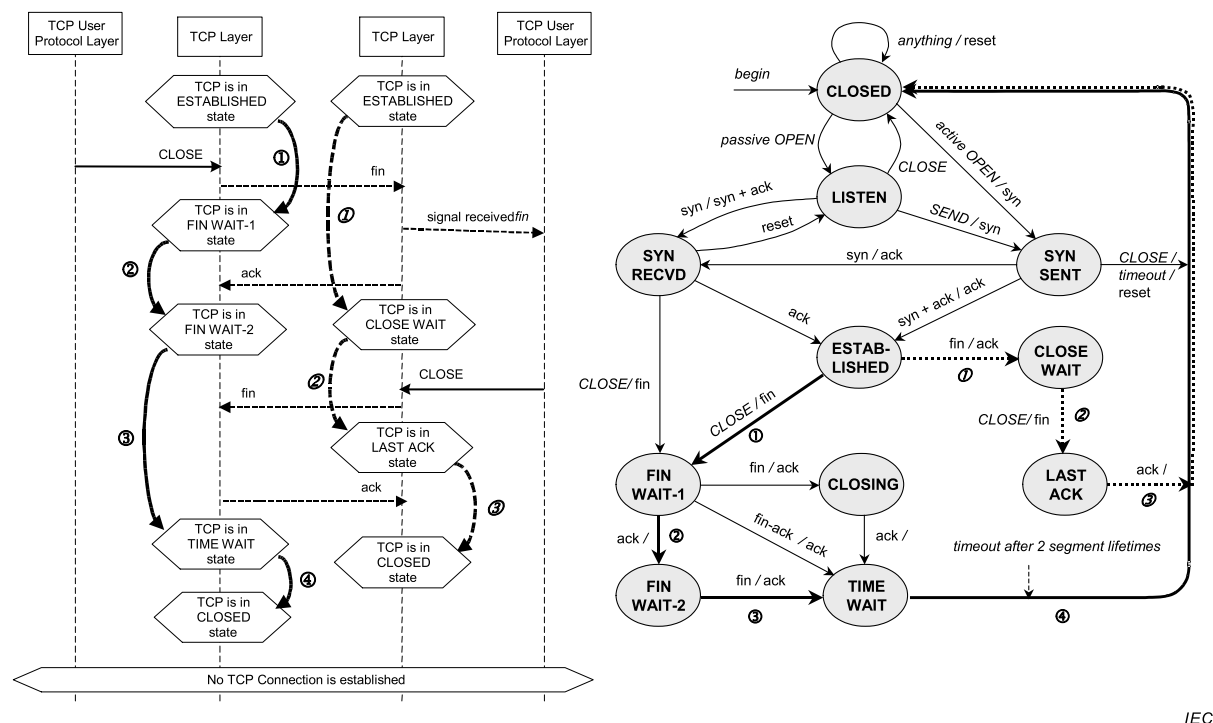
A.2 Fermeture d'une connexion de couche transport et TCP

La fermeture d'une connexion TCP est effectuée en appelant la fonction CLOSE, généralement quand il ne reste plus aucune donnée à envoyer.

Lors de l'appel de la primitive de service TCP-DISCONNECT.request par le processus gestionnaire de connexion TCP, la sous-couche enveloppe sollicite la fonction CLOSE de la sous-couche TCP.

Cependant, comme la connexion TCP est bidirectionnelle simultanée, l'autre côté peut encore avoir des données à envoyer. Par conséquent, après avoir sollicité la fonction CLOSE, le transport basé sur TCP peut ultérieurement continuer à recevoir des données et à en envoyer à l'AL DLMS/COSEM, jusqu'à ce qu'il soit informé que l'autre côté a également CLOSED. À ce moment-là, il génère la primitive COSEM-ABORT.indication, et toutes les AA sont libérées.

Le diagramme séquentiel des messages et les transitions d'états correspondant à une connexion TCP libérée avec succès sont indiqués dans la Figure A.3.



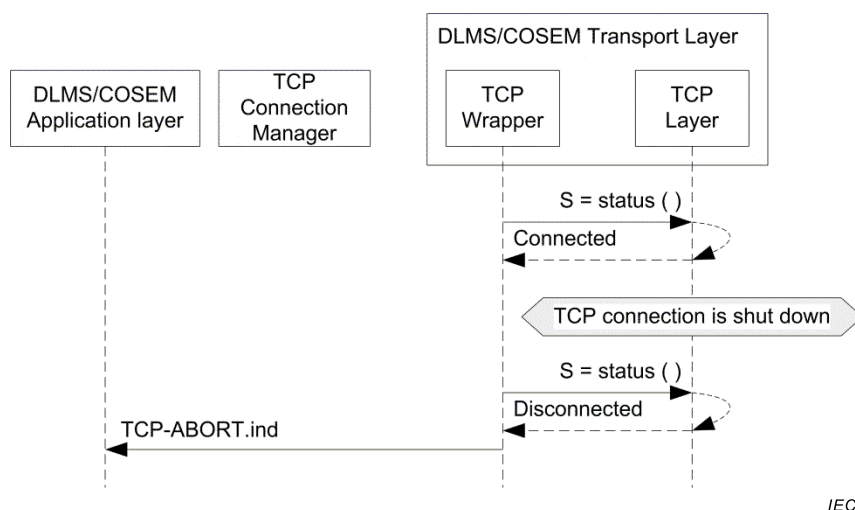
NOTE Dans le cas de la TL DLMS/COSEM, la couche de protocole utilisateur de TCP est la sous-couche enveloppe.

Anglais	Français
TCP User Protocol Layer	Couche de protocole utilisateur de TCP
TCP Layer	Couche TCP
TCP is in ESTABLISHED state	Le TCP est à l'état ESTABLISHED
TCP is in FIN WAIT-1 state	Le TCP est à l'état FIN WAIT-1
signal received	signal reçu
TCP is in FIN WAIT-2 state	Le TCP est à l'état FIN WAIT-2
TCP is in CLOSE WAIT state	Le TCP est à l'état CLOSE WAIT
TCP is in LAST ACK state	Le TCP est à l'état LAST ACK
TCP is in TIME WAIT state	Le TCP est à l'état TIME WAIT
TCP is in CLOSED state	Le TCP est à l'état CLOSED
No TCP connection is established	Aucune connexion TCP n'est établie
Timeout after 2 segment lifetimes	Temporisation après 2 durées de vie de segment

Figure A.3 – MSC et transitions d'états pour la fermeture d'une connexion de couche transport et TCP

A.3 Abandon de connexion TCP

Le document STD 0007 ne spécifie pas de fonction standard pour indiquer un abandon inattendu au niveau du TCP. Cependant, un abandon de connexion peut être détecté par l'entité utilisateur de TCP en interrogeant le statut du TCP à l'aide de la fonction STATUS(), comme indiqué dans la Figure A.4.



Anglais	Français
DLMS/COSEM Application Layer	Couche application DLMS/COSEM
TCP connection manager	Gestionnaire de connexion TCP
DLMS/COSEM transport Layer	Couche transport DLMS/COSEM
TCP Wrapper	Enveloppe TCP
TCP Layer	Couche TCP
Connected	Connecté
disconnected	déconnecté
Status	Statut
TCP connection is shut down	La connexion TCP est interrompue

Figure A.4 – Demande d'indication d'abandon TCP à la sous-couche TCP

A.4 Transfert de données utilisant le service TCP-DATA

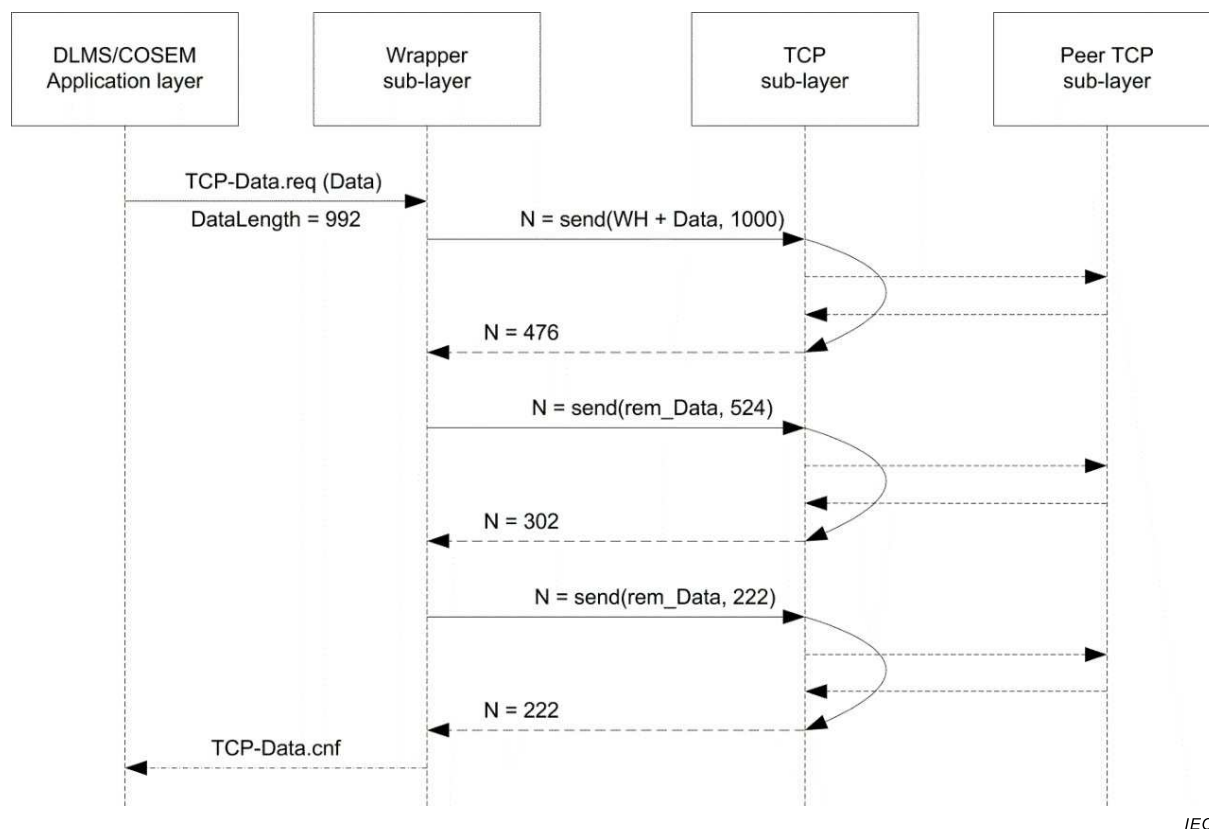
Pour envoyer une APDU à l'homologue, l'AL DLMS/COSEM appelle simplement la primitive TCP-DATA.request de la TL DLMS/COSEM basée sur TCP. De même, lorsqu'une APDU complète est reçue, cela est indiqué à l'AL DLMS/COSEM à l'aide de la primitive TCP-DATA.indication. Ensuite, pour l'AL, la TL se comporte comme si elle allait transporter l'APDU complète en une seule fois.

Cependant, comme le TCP est un protocole de «diffusion en flux» qui ne préserve pas les frontières des données comme décrit en 6.3.1, rien ne garantit qu'une APDU soit réellement transmise en un seul paquet TCP. Comme cela est déjà mentionné en 6.3.5.4, dans la TL DLMS/COSEM basée sur TCP, il incombe à la sous-couche enveloppe de «cacher» la nature «diffusion en flux» de la sous-couche TCP.

Les exemples suivants montrent comment la sous-couche enveloppe exécute cette tâche. Supposons qu'une entité AL souhaite envoyer une APDU contenant 992 octets par l'intermédiaire de la TL DLMS/COSEM basée sur TCP.

NOTE Les deux AL côté client comme côté serveur peuvent jouer le rôle d'émetteur comme de récepteur.

Elle appelle le service TCP-DATA.request avec cette APDU constituant le paramètre de service DATA, comme indiqué à la Figure A.5



Anglais	Français
DLMS/COSEM Application Layer	Couche application DLMS/COSEM
Wrapper sub-layer	Sous-couche enveloppe
TCP sub-layer	Sous-couche TCP
Peer TCP sub-layer	Sous-couche TCP homologue

Figure A.5 – Envoi d'une APDU en trois paquets TCP

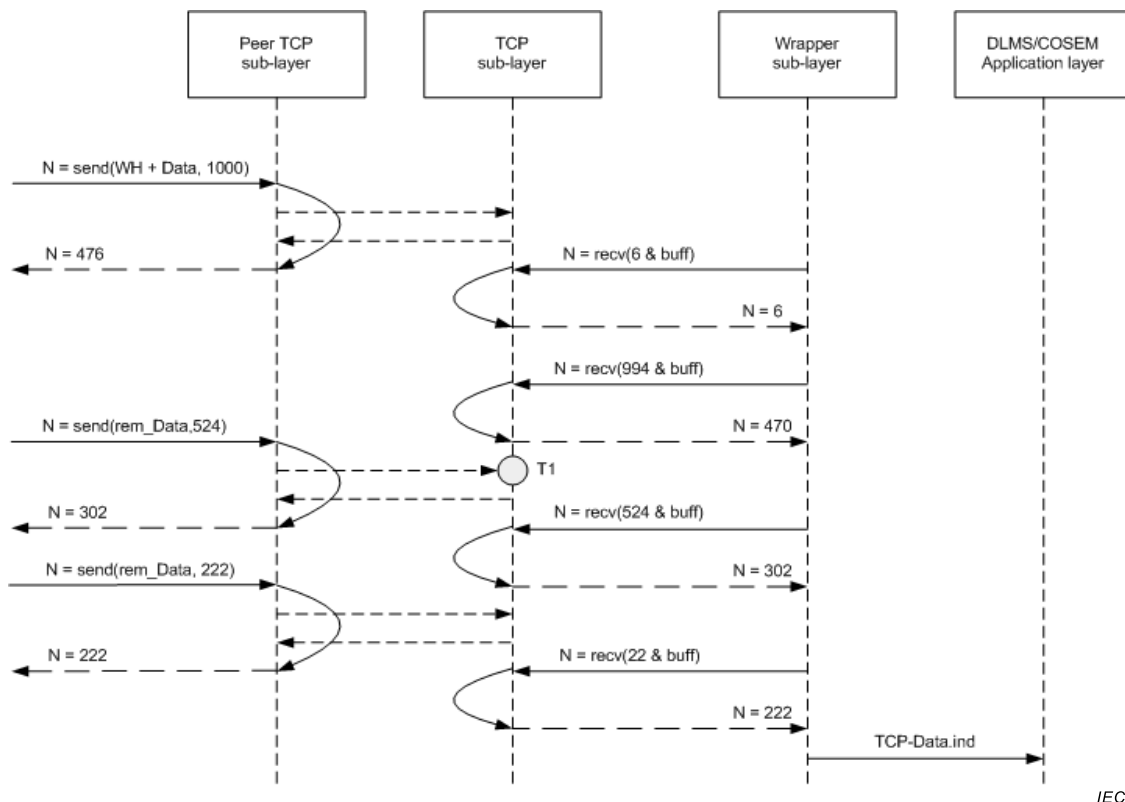
A la réception de cet appel de service, la sous-couche enveloppe construit la WPDU: elle prédétermine l'APDU avec l'en-tête de l'enveloppe (WH), y compris les numéros wPort local et distant et la longueur de l'APDU. Elle appelle ensuite la fonction SEND() de la sous-couche TCP, et lui demande d'envoyer la WPDU, qui fait maintenant 1 000 octets de long: 8 octets d'en-tête de l'enveloppe plus 992 octets d'APDU.

La fonction SEND() renvoie le nombre d'octets envoyés ou une erreur (une valeur négative). Supposons qu'aucune erreur ne se produise, et que la fonction SEND() renvoie les données avec succès, avec la valeur 476. Ce nombre désigne la quantité d'octets envoyés. Ceci illustre également le sens de la nature «diffusion en flux» du TCP: de fait, la fonction SEND() renvoie les données avec succès même si le nombre d'octets envoyés est inférieur au nombre d'octets à envoyer. À partir de la valeur renvoyée, l'enveloppe sait que la WPDU complète n'a pas été envoyée. Elle appelle à nouveau la fonction SEND() pour la partie restante de la WPDU, et ainsi de suite, jusqu'à ce que la WPDU complète soit envoyée.

Comme déjà mentionné en 6.3.5.4, en fonction de la mise en œuvre, le renvoi réussi de la fonction SEND() peut aussi ne pas signifier qu'un élément a véritablement été envoyé sur le réseau, et signifier uniquement que la mise en œuvre du protocole a pris les données et les a

prises en mémoire tampon. Il peut arriver que la mise en œuvre du protocole retarde la transmission pour se conformer aux conventions du protocole ou aux algorithmes relatifs au trafic sur le réseau.

Côté récepteur, il incombe également à la sous-couche enveloppe d'assembler l'APDU complète avant d'appeler la primitive TCP-DATA.indication, ce qui peut être fait en utilisant les octets de l'en-tête de la WPDU. L'enveloppe renouvelle les appels RECEIVE() jusqu'à ce que le nombre d'octets indiqués dans l'en-tête de la WPDU soit reçu. Ce protocole est indiqué à la Figure A.6.



IEC

NOTE 1 Étant donné que l'appel de la fonction RECEIVE() est asynchrone en ce qui concerne les communications TCP, il est tout à fait possible que le récepteur appelle la fonction RECEIVE() à un moment où la réception d'un paquet TCP est en cours (T1 dans la Figure ci-dessus), comme à un moment où aucun caractère n'a été reçu depuis le dernier appel RECEIVE(). Cette possibilité n'induit pas une réception erronée, mais augmente seulement le nombre d'appels de fonction RECEIVE() nécessaires pour obtenir le message complet.

NOTE 2 Il est également possible qu'un ou plusieurs appels SEND() entraînent l'envoi de plusieurs paquets TCP, ce qui n'induit pas non plus une réception erronée: tôt ou tard le récepteur obtient le message complet.

Anglais	Français
Peer TCP sub-layer	Sous-couche TCP homologue
TCP sub-layer	Sous-couche TCP
Wrapper sub-layer	Sous-couche enveloppe
DLMS/COSEM Application Layer	Couche application DLMS/COSEM

Figure A.6 – Réception du message en plusieurs paquets

Tous ces appels SEND() et RECEIVE() sont internes à la TL DLMS/COSEM. L'AL DLMS/COSEM utilisateur de service utilise simplement les services TCP-DATA, et offre un service de transfert de données fiable en préservant les frontières des données de l'APDU.

INDEX

- Abandon d'une connexion TCP, 76
- Abandon de connexion TCP, 70
- Acquittement positif avec retransmission, 57, 70
- active OPEN, 73
- Capacité d'adressage (wPort), 49
- Circuit virtuel, 57
- Communication des données TCP, 57
- Confirmation locale, 71
- Confirmation_Type, 66
- contrôle de flux, 57
- couche application DLMS/COSEM, 47
- Couche transport COSEM basée sur UDP, 50
- Couche transport COSEM, diagramme d'états, 57
- Couche transport DLMS/COSEM, basée sur TCP, 57
- Data length, 52, 54
- Datagramme UDP, 53
- Destination wPort, 54
- Détection d'erreur, 57
- Diagramme de transition d'états de la sous-couche enveloppe, 71
- Diffusion ou multidiffusion, 52
- En-tête enveloppe, 54
- Enveloppe, 49
- Etablissement d'une connexion TCP, 73
- établissement de la connexion, 57
- Etablissement de la connexion TCP, 67
- état SEND/RECEIVE, 71
- Fermeture d'une connexion TCP, 74
- Fermeture de la connexion TCP, 57
- Fonction SEND(), 52
- Fonctionnement bidirectionnel simultané, 57
- Identification de processus d'application, 66
- Initiateur, 68
- Interruption de TCP, 69
- Local_IP_Address, 52
- Local_TCP_Port, 59
- Local_UDP_Port, 52
- Local_wPort, 52
- Longueur des données, 49
- mécanisme triple prise de contact, 68
- No TCP Connection, 71
- Numéros de port d'enveloppe réservés, 67
- Numéros de port de l'enveloppe réservés, 56
- Numéros de séquence, 74
- numéros wPort valides, 65
- Numéros wPort valides, 53
- ouverture passive, 68
- Paquets TCP, 67
- Processus gestionnaire de connexion TCP, 50, 58
- protocole de contrôle de transmission, 45
- Protocole de contrôle de transmission, 57
- protocole de datagramme utilisateur, 45
- Protocole de datagramme utilisateur, 50
- Protocole Internet, 47
- Remote_IP_Address, 52
- Remote_TCP_Port, 59
- Remote_UDP_Port, 52
- Remote_wPort, 52
- Répondeur, 68
- segment FIN, 69
- Service TCP-ABORT, 63
- Service TCP-DATA, 64, 76
- Service UDP-DATA, 51
- Services du modèle OSI, 49
- Services TCP-DATA, 70
- services TCP-DISCONNECT, 61
- Source UDP, 56
- Source wPort, 54
- Sous-couche enveloppe, 49, 54, 66
- sous-état IDLE, 71
- Spécification de protocole pour la couche transport COSEM basée sur TCP, 66
- Spécification de protocole pour la couche transport COSEM basée sur UDP, 54
- Spécification de service pour la couche transport DLMS/COSEM basée sur TCP, 57
- Spécification de service pour la couche transport DLMS/COSEM basée sur UDP, 51
- TCP Connected, 71
- TCP-ABORT.indication, 63
- TCP-CONNECT, 59
- TCP-CONNECT.confirm, 60
- TCP-CONNECT.indication, 59
- TCP-CONNECT.request, 59
- TCP-CONNECT.response, 60
- TCP-DATA.confirm, 65
- TCP-DATA.indication, 65
- TCP-DATA.request, 64
- TCP-DISCONNECT.confirm, 63
- TCP-DISCONNECT.indication, 61
- TCP-DISCONNECT.request, 61
- TCP-DISCONNECT.response, 62
- Triple prise de contact, 73
- UDP-DATA.confirm, 53
- UDP-DATA.indication, 52
- UDP-DATA.request, 52
- Unité de données de protocole de l'enveloppe, 54, 67

Bibliographie

RFC 0768, *User Datagram Protocol (Également: IETF STD0006)*. août 1980. Disponible à l'adresse: <http://www.ietf.org/rfc/rfc768.txt>

RFC 0791, *Internet Protocol (Également: IETF STD 0005)*. Édité par J. Postel, septembre 1981. Disponible à l'adresse <http://www.ietf.org/rfc/rfc791.txt>

RFC 0792 *Internet Control Message Protocol (Également: IETF STD 0005)*, Édité par J. Postel, septembre 1981. Disponible à l'adresse <http://www.ietf.org/rfc/rfc792.txt>

RFC 0793, *Transmission Control Protocol (Également: IETF STD 0005)*. Édité par J. Postel, septembre 1981. Disponible à l'adresse <http://www.ietf.org/rfc/rfc793.txt>

RFC 0919, *Broadcasting Internet Datagrams (Également: IETF STD 0005)*. Édité par J. Mogul, octobre 1984. Disponible à l'adresse <http://www.ietf.org/rfc/rfc919.txt>

RFC 0922, *Broadcasting Internet datagrams in the presence of subnets (Également: IETF STD 0005)*. Édité par J. Mogul, octobre 1984. Disponible à l'adresse <http://www.ietf.org/rfc/rfc922.txt>

RFC 0950, *Internet Standard Subnetting Procedure (Également: IETF STD 0005)*. Édité par J. Mogul, J. Postel août 1985. Disponible à l'adresse <http://www.ietf.org/rfc/rfc950.txt>

RFC 1112, *Host extensions for IP multicasting (Également: IETF STD 0005)*. Édité par S. Deering août 1989. Disponible à l'adresse <http://www.ietf.org/rfc/rfc1112.txt>

RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*. Édité par S. Deering and R. Hinden. décembre 1998. Disponible à l'adresse: <http://www.ietf.org/rfc/rfc2460.txt>

RFC 3513, *Internet Protocol Version 6 (IPv6) Addressing Architecture*. Édité par R. Hinden S. Deering, avril 2003. Disponible à l'adresse: <http://www.ietf.org/rfc/rfc3513.txt>

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch