BS EN 62056-8-3:2013

BSI Standards Publication

Electricity metering data exchange — The DLMS/COSEM suite

Part 8-3: Communication profile for PLC S-FSK neighbourhood networks

National foreword

This British Standard is the UK implementation of EN 62056-8-3:2013. It is identical to IEC 62056-8-3:2013.

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

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

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

Published by BSI Standards Limited 2013

ISBN 978 0 580 75067 0
ICS 17.220.01; 35.110; 91.140.50

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

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 September 2013.

Amendments/corrigenda issued since publication

Date	Text affected

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN 62056-8-3

August 2013

ICS 17.220; 35.110; 91.140.50

English version

Electricity metering data exchange - The DLMS/COSEM suite - Part 8-3: Communication profile for PLC S-FSK neighbourhood networks
(IEC 62056-8-3:2013)

Echange des données de comptage de l'électricité -
La suite DLMS/COSEM -
Partie 8-3: Profil de communication pour réseaux de voisinage CPL S-FSK
(CEI 62056-8-3:2013)

Datenkommunikation der elektrischen Energiemessung -
DLMS/COSEM -
Teil 8-3: PLC S-FSK Spezifikation für Areal-Netze
(IEC 62056-8-3:2013)

This European Standard was approved by CENELEC on 2013-06-20. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

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

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

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

CENELEC

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

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

Foreword

The text of document 13/1526/FDIS, future edition 1 of IEC 62056-8-3, prepared by IEC/TC 13 "Electrical energy measurement, tariff- and load control" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 62056-8-3:2013.

The following dates are fixed:

• latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement	(dop)	2014-03-20
• latest date by which the national standards conflicting with the document have to be withdrawn	(dow)	2016-06-20

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

Endorsement notice

The text of the International Standard IEC 62056-8-3:2013 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following note has to be added for the standard indicated:

IEC 61334-4-512:2001 NOTE Harmonized as EN 61334-4-512:2002 (not modified).

Annex ZA
(normative)

Normative references to international publications with their corresponding European publications

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

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

Publication	Year	Title	EN/HD	Year
IEC 60050	Series	International Electrotechnical Vocabulary (IEV)	-	-
IEC 61334-4-1	1996	Distribution automation using distribution line carrier systems - Part 4: Data communication protocols - Section 1: Reference model of the communication system	EN 61334-4-1	1996
IEC 61334-4-32	1996	Distribution automation using distribution line carrier systems - Part 4: Data communication protocols - Section 32: Data link layer - Logical link control (LLC)	EN 61334-4-32	1996
IEC 61334-4-511	2000	Distribution automation using distribution line carrier systems - Part 4-511: Data communication protocols - Systems management - CIASE protocol	EN 61334-4-511	2000
IEC 61334-5-1	2001	Distribution automation using distribution line carrier systems - Part 5-1: Lower layer profiles - The spread frequency shift keying (S-FSK) profile	EN 61334-5-1	2001
IEC/TR 62051	1999	Electricity metering - Glossary of terms	-	-
IEC/TR 62051-1 + corr. June	2004 2005	Electricity metering - Data exchange for meter reading, tariff and load control - Glossary of terms - Part 1: Terms related to data exchange with metering equipment using DLMS/COSEM	-	-
IEC 62056-5-3	2013	Electricity metering data exchange - The DLMS/COSEM suite - Part 5-3: DLMS/COSEM application layer	EN 62056-5-3	2013
IEC 62056-6-2	2013	Electricity metering data exchange - The DLMS/COSEM suite - Part 6-2: COSEM interface classes	EN 62056-6-2	2013
IEC 62056-46 + A1	2002 2006	Electricity metering - Data exchange for meter reading, tariff and load control - Part 46: Data link layer using HDLC protocol	EN 62056-46 + A1	2002 2007
ISO/IEC 8802-2 + corr. October	1998 2000	Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 2: Logical link control	-	-

CONTENTS

ELECTRICITY METERING DATA EXCHANGE – THE DLMS/COSEM SUITE –

Part 8-3: Communication profile for PLC S-FSK neighbourhood networks

1 Scope

This part of IEC 62056 specifies the DLMS/COSEM PLC S-SFK communication profile for neighbourhood networks.

It uses standards established by IEC TC 57 in the IEC 61334 series, *Distribution automation using distribution line carrier systems* and it specifies extensions to some of those standards.

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 (all parts), *International Electrotechnical Vocabulary* (available at http://www.electropedia.org)

IEC 61334-4-1:1996, *Distribution automation using distribution line carrier systems – Part 4: Data communication protocols – Section 1: Reference model of the communication system*

IEC 61334-4-32:1996, *Distribution automation using distribution line carrier systems – Part 4: Data communication protocols – Section 32: Data link layer – Logical link control (LLC)*

IEC 61334-4-511:2000, *Distribution automation using distribution line carrier systems – Part 4-511: Data communication protocols – Systems management – CIASE protocol*

IEC 61334-5-1:2001, *Distribution automation using distribution line carrier systems – Part 5-1: Lower layer profiles – The spread frequency shift keying (S-FSK) profile*

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 62056-46:2002, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 46: Data link layer using HDLC protocol*
Amendment 1:2006

IEC 62056-5-3:—, *Electricity metering data exchange – The DLMS/COSEM suite – Part 5-3: DLMS/COSEM application layer*[2]

[2] To be published simultaneously with this part of IEC 62056.

IEC 62056-6-2:—, *Electricity metering data exchange – The DLMS/COSEM suite – Part 6-2: COSEM interface classes*[3]

ISO/IEC 8802-2:1998, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 2: Logical link control*

NOTE See also the Bibliography.

3 Terms, definitions and abbreviations

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

Where there is a difference between the definitions in the glossary and those contained in product standards produced by TC 13, then the latter shall take precedence in applications of the relevant standard.

3.1 Terms and definitions

3.1.1
initiator
user-element of a client System Management Application Entity (SMAE). It uses the CIASE and xDLMS ASE and it is identified by its system title

[SOURCE: IEC 61334-4-511:2000, 3.8.1, modified]

3.1.2
active initiator
initiator, which issues or has last issued a CIASE Register request when the server is in the unconfigured state

[SOURCE: IEC 61334-4-511:2000, 3.9.1]

3.1.3
new system
server system, which is in the unconfigured state: its MAC address equals "NEW-address"

[SOURCE: IEC 61334-4-511:2000, 3.9.3]

3.1.4
new system title
system-title of a new system

Note 1 to entry: This is the system title of a system, which is in the new state.

[SOURCE: IEC 61334-4-511:2000, 3.9.4, modified]

3.1.5
registered system
server system, which has an individual, valid MAC address

Note 1 to entry: Therefore, this MAC address is different from "NEW Address", see IEC 61334-5-1: Medium Access Control.

[SOURCE: IEC 61334-4-511:2000, 3.9.5, modified]

---

[3] To be published simultaneously with this part of IEC 62056.

3.1.6
reporting system
server system, which issues a DiscoverReport

[SOURCE: IEC 61334-4-511:2000, 3.9.6, modified]

3.1.7
sub-timeslot
the time needed to transmit two bytes by the physical layer

Note 1 to entry: Timeslots are divided to sub-slots in the RepeaterCall mode of the physical layer.

3.1.8
timeslot
the time needed to transmit a physical frame

Note 1 to entry: As specified in IEC 61334-5-1:2001, 3.3.1, a physical frame comprises 2 bytes preamble, 2 bytes start subframe delimiter, 38 bytes PSDU and 3 bytes pause.

3.2 Abbreviations

.cnf	.confirm service primitive
.ind	.indication service primitive
.req	.request service primitive
.res	.response service primitive
AA	Application Association
AARE	A-Associate Response – an APDU of the ACSE
AARQ	A-Associate Request – an APDU of the ACSE
ACSE	Association Control Service Element
AES	Advanced Encryption Standard
AL	Application Layer
AP	Application Process
APDU	Application Layer Protocol Data Unit
ASE	Application Service Element
ASO	Application service Object
A-XDR	Adapted Extended Data Representation
CIASE	Configuration Initiation Application Service Element
CI-PDU	CIASE PDU
Client	A station, asking for services. In the case of the 3-layer, CO HDLC based profile it is the master station
COSEM	Companion Specification for Energy Metering
DA	Destination Address
DLMS	Device Language Message Specification
DLMS UA	DLMS User Association
FCS	Frame Check Sequence
GCM	Galois/Counter Mode, an algorithm for authenticated encryption with associated data
HCS	Header Check Sequence
HDLC	High-level Data Link Control
HES	(Metering) Head End System
ISO	International Organization for Standardization

LLC	Logical Link Control (Sublayer)
LN	Local Network
LNAP	Local Network Access Point
L-SAP	LLC sublayer Service Access Point
LSDU	LLC Service Data Unit
LV	Low voltage
MAC	Medium Access Control (sublayer)
MPDU	MAC Layer Protocol Data Unit
MSC	Message Sequence Chart
NN	Neighbourhood Network
NNAP	Neighbourhood Network Access Point
NS	Number of subframes (S-FSK MAC sublayer)
OSI	Open System Interconnection
PDU	Protocol Data Unit
PhL	Physical Layer
PLC	Power Line Carrier
PSDU	Physical Layer Service Data Unit
RDR	Reply Data on Request (used in IEC 61334-4-32)
RLRE	A-Release Response – an APDU of the ACSE
RLRQ	A-Release Request – an APDU of the ACSE
SA	Source Address
SAP	Service Access Point
SDN	Send Data Non-acknowledged (used in IEC 61334-4-32)
SDU	Service Data Unit
SMAE	Systems Management Application Entity
SMAP	Systems Management Application Process
SNRM	Set Normal Response Mode (a HDLC frame type)

4 Targeted communication environments

The DLMS/COSEM PLC S-FSK communication profile is intended for remote data exchange on Neighbourhood Networks (NN) between Neighbourhood Network Access Points (NNAP) *and* Local Network Access Points (LNAPs) or End Devices using S-FSK power line carrier technology over the low voltage electricity distribution network as a communication medium. The functional reference architecture is shown in Figure 1.

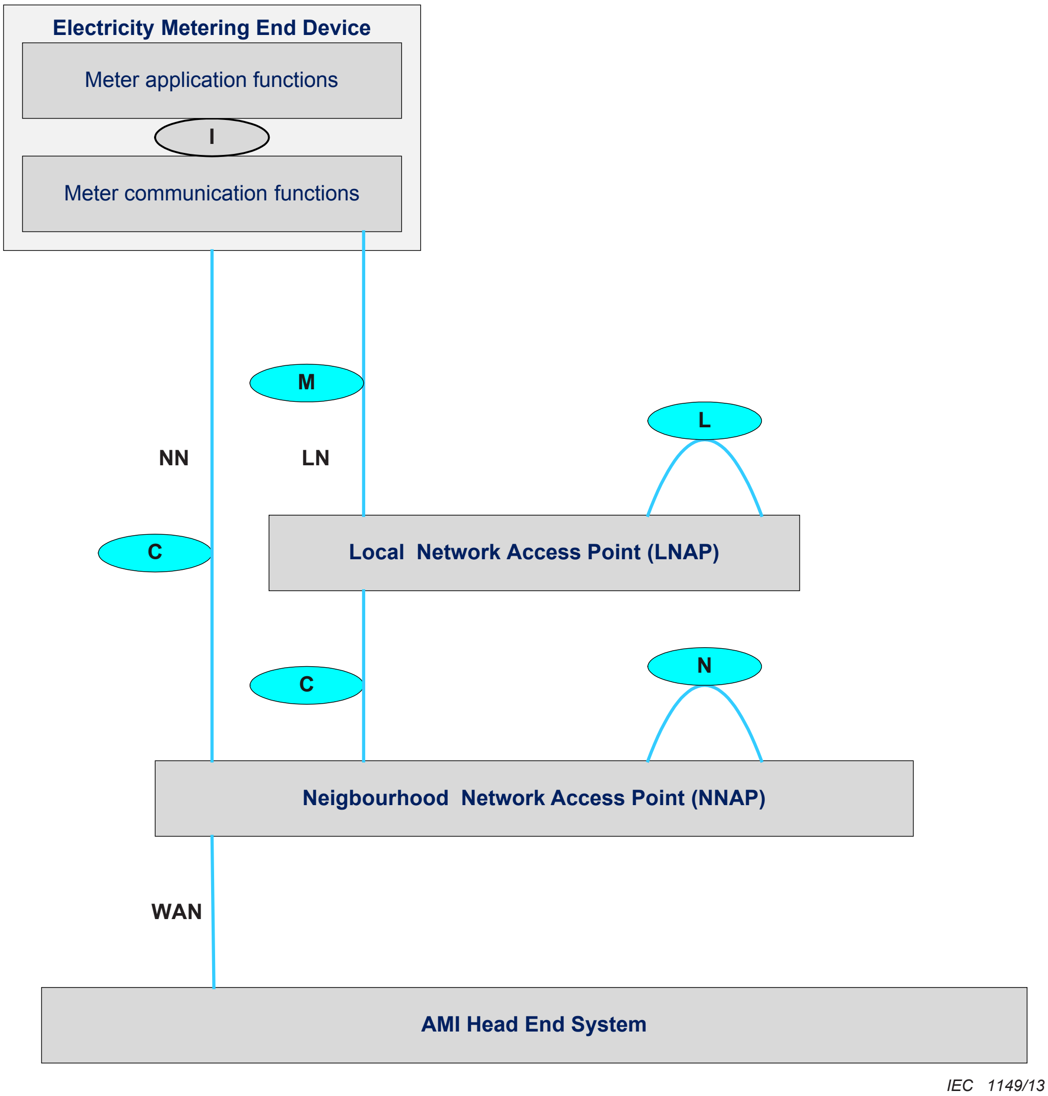


Figure 1 – Communication architecture

End devices – typically electricity meters – comprise application functions and communication functions. They may be connected directly to the NNAP via the C interface, or to an LNAP via an M interface, while the LNAP is connected to the NNAP via the C interface. The LNAP function may be co-located with the metering functions.

A NNAP comprises gateway functions and it may comprise concentrator functions. Upstream, it is connected to the Metering Head End System (HES) using suitable communication media and protocols.

End devices and LNAPs may communicate to different NNAPs, but to one NNAP only at a time. From the PLC communication point of view, the NNAP acts as an initiator while end devices and LNAPs act as responders.

NNAPs and similarly LNAPs may communicate to each other, but this is out of the scope of this standard, which covers the C interface only.

When the NNAP has concentrator functions, it acts as a COSEM client. When the NNAP has gateway functions only, then the HES acts as a COSEM client. The end devices or the LNAPs act as COSEM servers.

5 Reference model

NOTE This clause is partly based on IEC 61334-4-1:1996, Clause 3.

The reference model of the *DLMS/COSEM PLC S-FSK communication profile is* shown in Figure 2. It is based on a simplified – or collapsed – three-layer OSI architecture. The layers are the *physical layer*, the *data link layer* and the *application layer*. The data link layer is split to the *MAC sublayer* and the *LLC sublayer*.

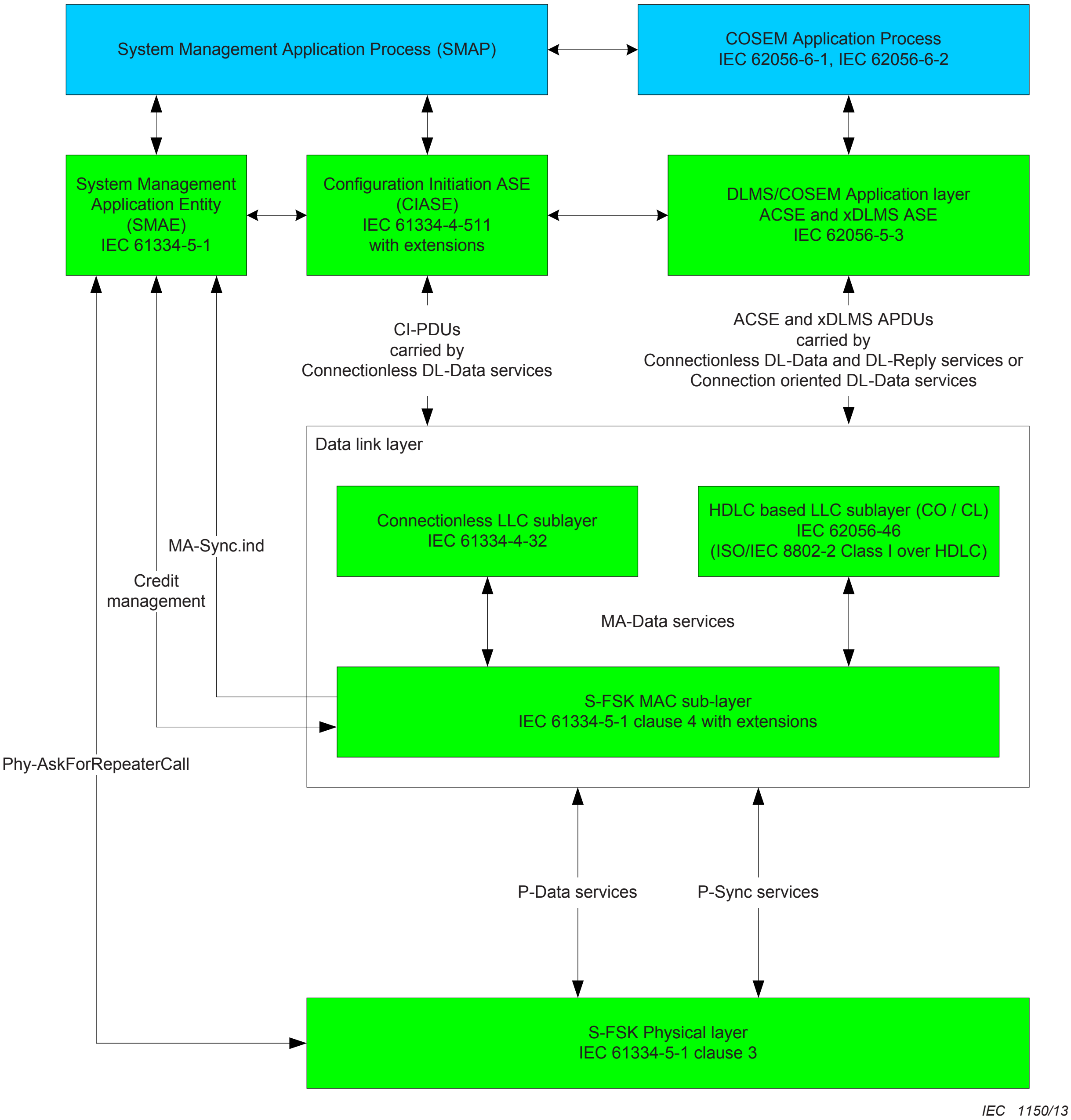


Figure 2 – The DLMS/COSEM S-FSK PLC communication profile

6 The physical layer (PhL)

The PhL provides the interface between the equipment and the physical transmission medium that is the distribution network. It transports binary information from the source to the destination.

The PhL in this profile is as specified in IEC 61334-5-1:2001, Clause 3. It provides the following services to its service user MAC sublayer:

- P-Data services to transfer MPDUs to (a) peer MAC sublayer entity(ies) using the LV distribution network as the transport medium;
- P-Sync services to allow the MAC sublayer entity to ask for a new synchronization and to be informed of a change in the synchronization state of the PL. These services are used locally by the MAC sublayer.

See IEC 61334-5-1:2001, 3.4.

7 The data link layer

7.1 General

The data link layer consists of two sublayers: the Medium Access Control (MAC) and the Logical Link Control (LLC) sublayer.

The MAC sublayer handles access to the physical medium and provides physical device addressing. The decision to access the medium is made by the initiator, directly for its own MAC sublayer, or indirectly for other MAC sublayers that are requested to transmit a response to a request sent previously by the initiator.

The LLC sublayer controls the logical links.

There are two LLC sublayer alternatives available:

- the connectionless LLC sublayer, as specified in IEC 61334-4-32;
- the LLC sublayer using the HDLC based data link layer, as specified in IEC 62056-46.

7.2 The MAC sublayer

The MAC sublayer of the DLMS/COSEM S-FSK PLC communication profile is as specified in IEC 61334-5-1:2001, Clause 4. It provides the following services to its service user LLC sublayer:

- the MA-Data services. These services allow the LLC sublayer entity to exchange LLC data units with peer LLC sublayer entities. See IEC 61334-5-1:2001, 4.1.3.1;
- the MA-Sync.indication service. This allows the SMAE entity to be informed of the synchronization and configuration status of the device. See IEC 61334-5-1:2001, 4.1.3.2.

7.3 The connectionless LLC sublayer

The connectionless LLC sublayer is as specified in IEC 61334-4-32. It is derived from ISO/IEC 8802-2 – similar to Class III operation – and it performs the following functions:

- addressing of application entities within the equipment;
- sending data with no acknowledgement (SDN);
- reply data on request (RDR).

It provides the following services:

- DL-Data services for transporting CI-PDUs, ACSE APDUs and client-server type xDLMS APDUs;
- DL-Reply services for asking the remote LLC sublayer entity to send a previously prepared LSDU;
- DL-Update-Reply services to prepare the LSDUs to be transferred using the DL-Reply services.

For more details, see IEC 61334-4-32:1996, 2.1.

7.4 The HDLC based LLC sublayer

The HDLC based LLC sublayer is as specified in IEC 62056-46.

As explained in IEC 62056-46:2002, 4.1 and 4.2, this sublayer can also be divided to two sublayers:

- the LLC sublayer based on ISO/IEC 8802-2. Here, it is used in an extended Class I operation. The only role of this sublayer is to select the DLMS/COSEM Application layer by using a specific LLC address. The LLC services are provided by the HDLC based MAC sublayer;
- the MAC sublayer, based on the HDLC protocol. It provides addressing of application entities within the equipment.

NOTE In this profile, there are two MAC sublayers. The HDLC MAC sublayer provides reliable LLC data transport and segmentation. The Medium Access Control functionality is provided by the S-FSK MAC sublayer specified in 7.2.

The HDLC based LLC sublayer provides the following services:

- DL-Connect services to connect and to disconnect the data link layer;
- connectionless DL-Data services for transporting CI-PDUs, ACSE APDUs and xDLMS APDUs;
- connection oriented DL-Data services for transporting ACSE APDUs and xDLMS APDUs. These services provide reliable data transport and support segmentation to carry long messages, in a transparent manner for the application layer.

7.5 Co-existence of the connectionless and the HDLC based LLC sublayers

The frames of the connectionless LLC sublayer and the HDLC based LLC sublayer can be distinguished from each other as shown in Figure 3. This allows systems using the two profiles to co-exist on the same network.

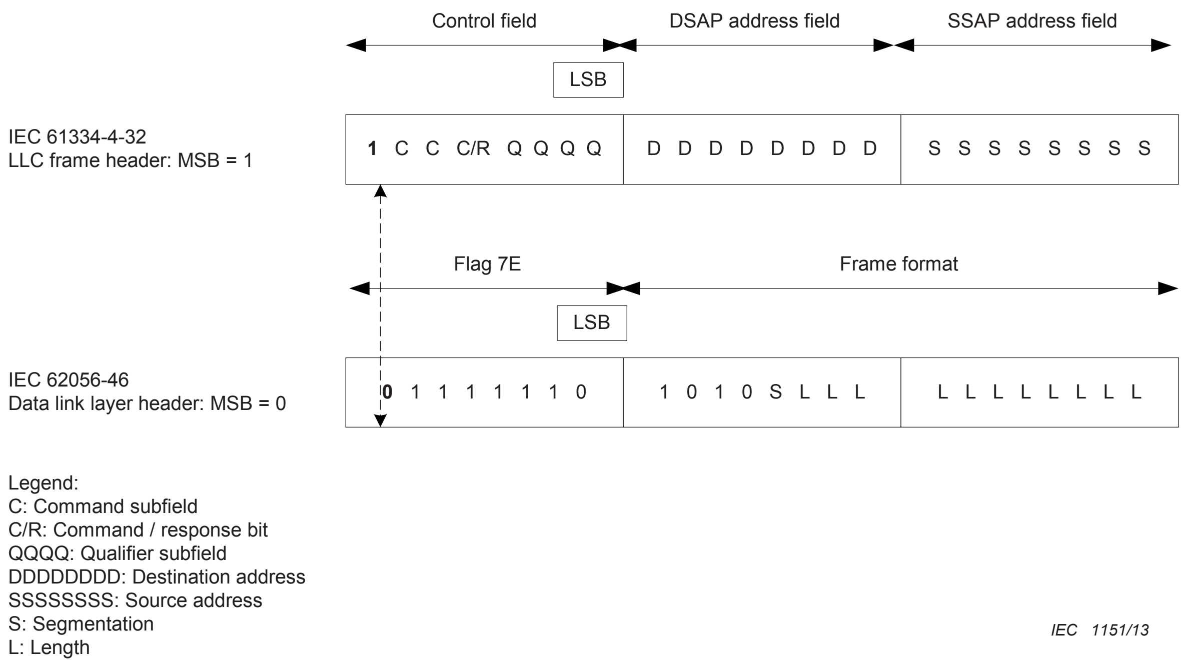


Figure 3 – Co-existence of the connectionless and the HDLC based LLC sublayers

8 The application layer (AL)

Concerning the application layer, the DLMS/COSEM Application layer as specified in IEC 62056-5-3 applies. It provides services to the COSEM application process (AP) and uses the services of the connectionless or the HDLC based LLC sublayer.

9 The application process (AP)

On the server side, the COSEM device- and object model – as specified in IEC 62056-6-2 – applies. Each logical device represents an AP.

The client side APs make use of the resources of the server side AP. A physical device may host one or more client APs.

10 The Configuration Initiation Application Service Element (CIASE)

NOTE This clause is based on IEC 61334-4-511 and constitutes an extension to it.

10.1 Overview

One of the activities of systems management is open system initialisation and / or modification. This is provided by the Configuration Initiation ASE (CIASE). It is specified in IEC 61334-4-511 with the extensions specified below.

The CIASE services are the following:

- the Discover service;
- the Register service;
- the PING service;
- the RepeaterCall service; and
- the ClearAlarm service.

The three latter services, together with the Intelligent Search Initiator process specified in 10.7, constitute upper compatible functional extensions to IEC 61334-4-511.

The CIASE uses the connectionless DL-Data services of the LLC sublayer.

10.2 The Discover service

NOTE In this document, the description of the CIASE services follows the presentation style used for the DLMS/COSEM services. For the notation used, see IEC 62056-5-3:—, 6.1.

The Discover service is used to discover new systems or systems, which are in alarm state. It is specified in IEC 61334-4-511:2000, 7.1. The Discover service primitives shall provide the parameters as shown in Table 1.

Table 1 – Service parameters of the Discover service primitives

	Discover		DiscoverReport	
	.request	.indication	.response	.confirm
Argument	M	M (=)	–	–
Response_Probability	M	M (=)		
Allowed_Time_Slots	M	M (=)		
DiscoverReport_Initial_Credit	M	M (=)		
IC_Equal_Credit	M	M (=)	–	–
Result (+)	–	–	S	S (=)
System_Title {System_Title}			M	M (=)
Alarm_Descriptor			C	C (=)
Result (-)	–	–	S	S (=)
Argument_Error(s)			M	M (=)
NOTE This Table 1 is included here for completeness and to correct some editorial errors in IEC 61334-4-511:2000, 7.1. For the description of the service parameters, see the clause referenced here.				

10.3 The Register service

The Register service is used to perform system configuration. It assigns a MAC address to a new system identified by its system title. It is specified in IEC 61334-4-511:2000, 7.2. The Register service primitives shall provide the parameters as shown in Table 2.

Table 2 – Service parameters of the Register service primitives

	.request	.indication
Argument		
Active_Initiator_System_Title	M	M (=)
List_Of_Correspondence	M	M (=)
New_System_Title	M	M (=)
MAC_Address	M	M (=)
Result (+)	S	S
Result (–)	S	S
Argument_Error(s)	M	M (=)
NOTE This Table 2 is included here for completeness. For the description of the service parameters, see IEC 61334-4-511:2000, 7.2.		

NOTE 1 If a server in NEW state receives a correct Register service with its own server system title in it, it will be registered, even if it did not receive a Discover service before.

NOTE 2 Only those servers in the NEW state can be registered.

10.4 The Ping Service

Function

The Ping service is used to check that a server system already registered is still present on the network. It also allows verifying that the right physical device is linked to the right MAC address. It also allows preventing the *time_out_not_addressed* timer to expire.

The process begins with a Ping.request service primitive issued by the active initiator. The service contains the system title of the physical device pinged. The PingRequest CI-PDU is carried by a DL-Data.request service primitive and it is sent to the MAC address assigned to this system and to the server CIASE L-SAP.

If the system title carried by the Ping.indication service primitive is equal to the system title of the server, the server shall respond with a Ping.response service primitive, carrying the system title of the server. It is sent to the initiator CIASE L-SAP.

Semantics

The PING service primitives shall provide parameters as shown in Table 3.

Table 3 – Service parameters of the PING service primitives

	.request	.indication	.response	.confirm
Argument System_Title_Server	M	M (=)	-	-
Result (+) System_Title_Server	–	–	S M	S (=) M (=)
Result (–) Argument_Error(s)	–	–	S M	S M (=)

The System_Title_Server service parameter allows identifying a physical device concerned by the Ping service. The destination MAC address in the DL-Data.request service primitive is equal to the MAC address that has been assigned to this system using the Register service.

The Ping.response service primitive returns with « Result (+) » if the Ping.request service has succeeded, i.e. the system title of the physical device at the given MAC address is equal to the “System_Title_Server” carried by the Ping.request service primitive.

Otherwise, no response is sent by the server.

Use – Client side

The Ping.request service primitive is issued by the active initiator.

If the “System_Title_Server” service parameter is not valid, a local confirmation is sent immediately with a negative result indicating the problem encountered (Ping-system-title-nok).

Otherwise, the CIASE forms a DL-Data.request PDU containing a PingRequest CI-PDU that carries the System_Title_Server requested. It is sent to the physical device concerned by the request.

Once the transmission of the PingRequest CI-PDU is over, the CIASE waits for a DL-Data.indication service primitive containing a PingResponse CI-PDU from the physical device pinged, during the necessary time that depends on the initial credit of the request.

If the CIASE receives a DL-Data.indication service primitive containing a PingResponse CI-PDU before this delay is over, it sends to the initiator a confirmation with a positive result, containing the service parameter returned by the server system.

If no DL-DATA.indication service primitive is received, the CIASE sends to the initiator a confirmation with a negative result pointing out the absence of an answer (Ping-no-response).

Use – Server side

On the reception of a DL-Data.indication service primitive containing a PingRequest CI-PDU, the CIASE checks that the System_Title_Server service parameter is correct and that it is equal to its own system title.

If so, it invokes a Ping.response service primitive that includes its system title. The PingResponse CI-PDU is carried by a DL-Data.request service primitive.

If the service parameter of the Ping.indication service primitive is not correct, no response is sent.

Finally, if the System_Title_Server service parameter in the Ping.indication service primitive is correct, but not equal to the system title of the physical device, no response is sent.

10.5 The RepeaterCall service

Function

The purpose of the RepeaterCall service is to adapt the repeater status of server systems depending on the topology of the electrical network. It allows the automatic configuration of the repeater status on the whole network.

In the RepeaterCall mode, the client and the servers transmit short frames – two bytes long each – and measure the level of the signal to determine if a server system should be a repeater or not.

Semantics

The RepeaterCall service primitives shall provide parameters as shown in Table 4.

Table 4 – Service parameters of the RepeaterCall service primitives

	. request	**. indication**
Arguments		
Max_Adr_MAC	M	M (=)
Nb_Tslot_For_New	U	U (=)
Reception_Threshold	M	M (=)
Result (+)	S	S
Result (-)	S	S
Argument_Error(s)	M	M

The Max_Adr_MAC service parameter allows calculating the number of timeslots used in the RepeaterCall mode by the physical layer of the server systems registered to an initiator. It corresponds to the largest server system MAC address that is stored by the initiator.

NOTE 1 The largest allowable server MAC address is 3071 (BFF), as the range C00...DFF is reserved for the initiator, as specified in IEC 61334-5-1:2001, 4.3.7.7.1.

The value of Nb_Tslot is calculated from this information:

$$Nb\_Tslot = \lfloor MaxAdrMax / 21 \rfloor + 1$$

where $\lfloor x \rfloor$ means the floor of x, the nearest integer ≤ x.

EXAMPLE 1

Max_Adr_MAC = 20 (20 servers on the network)

$$Nb\_Tslot = \lfloor 20/21 \rfloor + 1 = 1$$

Nb_Tslot = 1, with 1 sub-timeslot for the NNAP (concentrator) and 20 sub-timeslots for the servers.

EXAMPLE 2

Max_Adr_MAC = 21 (21 servers on the network)

$$Nb\_Tslot = \lfloor 21/21 \rfloor + 1 = 2$$

Nb_Tslot = 2:

1 timeslot with 1 sub-timeslot for the NNAP (concentrator) and 20 sub-timeslots for 20 servers;

1 timeslot with 1 sub-timeslot for one server.

The Nb_Tslot_For_New service parameter defines the number of timeslots used in the RepeaterCall mode by the physical layer of the server systems in NEW state. If the value of this service parameter is 0 (or not present) then the systems in NEW state are not allowed to participate in the Repeater Call process.

The maximum number of timeslots used by the physical layer is equal to the sum of the number of timeslots for the server systems registered and the number of timeslots for the server systems in NEW state.

The Reception_Threshold service parameter defines the threshold of the signal level in dBµV, necessary to validate a physical pattern in a Sub_Tslot when the physical layer is in the RepeaterCall mode.

The Result (+) service parameter (positive result) indicates that the requested service has succeeded.

The Result (–) service parameter (negative result) indicates that the requested service has failed.

The “Arguments Error” indicates that at least one argument has a wrong value.

Use – Client side

The RepeaterCall service of the CIASE is invoked by the SMAP.

If any of the arguments is not valid, a confirmation is sent immediately with a negative result indicating the problem encountered.

Otherwise, the CIASE forms a DL-Data.request service primitive containing a RepeaterCall CI-PDU carrying the parameters requested. This request is sent to all server systems.

A positive confirmation is passed to the CIASE upon the reception of a DL-Data.cnf(+).

When this confirmation is received, the CIASE sends the Phy_AskForRepeaterCall.request primitive allowing the activation of the RepeaterCall mode of the physical layer. The parameters of this primitive are the following:

- Sub_Tslot position: on the client (initiator) side its value is 0;
- Reception_Threshold.

NOTE 2 On the client side, the Reception_Threshold parameter has no significance.

Use – server side

On the server side, on the reception of a DL-Data.indication service primitive containing a RepeaterCall CI-PDU, the CIASE verifies that the service parameters are correct.

If this is the case, it sends the Phy_AskForRepeaterCall.request primitive to activate the RepeaterCall mode of the physical layer. The parameters of this primitive are the following:

- Sub_Tslot position: A number expressed on two octets, between 0 and 65 535. The value 0 is reserved for the configuration of the NNAP (concentrator). The other values are available for the configuration of server systems.

  In the case of server systems registered by a NNAP (concentrator), Sub-Tslot takes the value of the local MAC address of the server system, between 1 and Max_Adr_MAC. For the server systems not registered, Sub_Tslot takes a random value between Max_Adr_MAC and Max_Adr_MAC + (Nb_Tslot_For_New * 21).

  NOTE 3 Max_Adr_MAC and Nb_Tslot_For_New are the service parameters of the RepeaterCall.request service.

- Reception_Threshold: This represents the signal level in dBµV. The default value is 104.

If the response to this request is negative, the command is cancelled. The following cases lead to a failure:

- the state of the physical layer is not correct (there is no physical synchronization);
- the repeater status is never_repeater;
- the parameters are incorrect.

The participation of the servers in the repeater call process and the effect of the process on their repeater status depend on the *repeater* management variable (attribute 10 of the S-FSK Phy&MAC setup object, see IEC 62056-6-2:—, 5.8.4) and the signal level heard:

- servers configured as never repeater do not participate: they do not transmit during their sub-timeslot and their repeater_status (attribute 11 of the S-FSK Phy&MAC setup object) is not affected;
- servers configured as always repeater participate: they transmit during their timeslot but their repeater_status is not affected;
- servers configured as dynamic repeater participate: they transmit during their sub-timeslot, if they have not heard a signal before from the client or from any servers, which is above the reception threshold. If during the whole repeater call process, a server does not hear a signal from the client or from other servers, which is above the reception threshold, then its repeater status will be TRUE: the server will repeat all frames. If a server hears a signal from the client or from other servers, which is above the reception threshold, then its repeater status will be FALSE: the server won't repeat any frames.

  NOTE 4 If each server configured as dynamic repeater hears a signal, which is above the reception threshold, this means that they are all close to a client, and no repetition is needed. So, none of them will become a repeater.

10.6 The ClearAlarm service

Function

The ClearAlarm service allows clearing the alarm state in (a) server system(s), in a point-to-point or in a broadcast mode.

Semantics

The ClearAlarm service primitives shall provide parameters as shown in Table 5.

Table 5 – Service parameters of the ClearAlarm service primitives

	.request	.indication
Arguments		
Alarm_Descriptor	S	S (=)
Alarm_Descriptor {Alarm_Descriptor}	S	S (=)
Alarm_Descriptor_List_And_Server_List	S	S (=)
System_Title {System_Title}	M	M (=)
Alarm_Descriptor {Alarm_Descriptor}	M	M (=)
Alarm_Descriptor_By_Server {Alarm_Descriptor_By_Server}	S	S (=)
System_Title	M	M (=)
Alarm_Descriptor	M	M (=)
Result (+)	S	S
Result (-)	S	S
Arguments Error	M	M

This service provides four different possibilities:

- the Alarm_Descriptor choice allows clearing a single alarm in all server systems. The value of the Alarm_Descriptor parameter identifies the alarm to be cleared;
- the Alarm_Descriptor {Alarm_Descriptor} choice allows clearing a list of alarms in all server systems. The value of the Alarm_Descriptor {Alarm_Descriptor} parameter identifies the list of alarms to be cleared;
- the Alarm_Descriptor_List_And_Server_List choice allows clearing a common list of alarms specified in the list of server systems specified. The System_Title {System_Title} parameter identifies the list of server systems in which the alarms have to be cleared. The value of the Alarm_Descriptor {Alarm_Descriptor} parameter identifies the list of alarms to be cleared;
- the Alarm_Descriptor_By_Server {Alarm_Descriptor_By_Server} choice allows clearing one alarm specified in each server specified. The System_Title parameter identifies the server system in which the alarm has to be cleared. The value of the Alarm_Descriptor parameter identifies the alarm to be cleared.

As specified in IEC 61334-4-511:2000, 6.2.2, alarm descriptors should be specified in companion specifications.

The Result (+) argument (positive result) indicates that the requested service has succeeded.

The Result (–) argument (negative result) indicates that the requested service has failed.

The “Argument Error(s)” indicates that at least one argument has a wrong value.

Use

The ClearAlarm CIASE service is invoked by the initiator.

If any of the service parameters is not valid, a confirmation is sent immediately with a negative result indicating the problem encountered.

Otherwise, the CIASE forms a DL-Data.request service primitive containing a ClearAlarm CI-PDU containing the parameters requested. This request is sent to the server system(s) concerned by the request. A positive confirmation is sent upon the reception of a DL-Data.cnf(+) service primitive.

On the server side, on the reception of a DL-Data.indication service primitive containing a ClearAlarm CI-PDU, the CIASE verifies that the arguments are correct. If this is the case, it clears the alarms corresponding to the list of alarms. Otherwise, the service is ignored.

10.7 The Intelligent Search Initiator process

10.7.1 General

The objective of the Intelligent Search Initiator process is to improve plug&play installation of server systems, by ensuring that each server system is registered by the correct initiator.

When a new server system is placed on the network, it will be discovered and registered by the first initiator it hears talking. It remains registered by that initiator as long as it keeps receiving correct frames (the time_out_not_addressed timer does not expire). If there is cross-talk on the network, the server system may be registered by the wrong initiator, i.e. one, which is “heard” by the server system due to cross-talk.

When the Intelligent Search Initiator process is implemented in the server system, it is capable to establish a list of all initiators it can “hear”, and to lock on the initiator with the best signal level.

10.7.2 Operation

10.7.2.1 Flow chart

The intelligent search initiator process comprises two phases:

- the *Search Initiator* phase;
- the *Check Initiator phase.*

The Intelligent Search Initiator process is shown in Figure 4. See also Figure 5 showing the complete discovery and registration process.

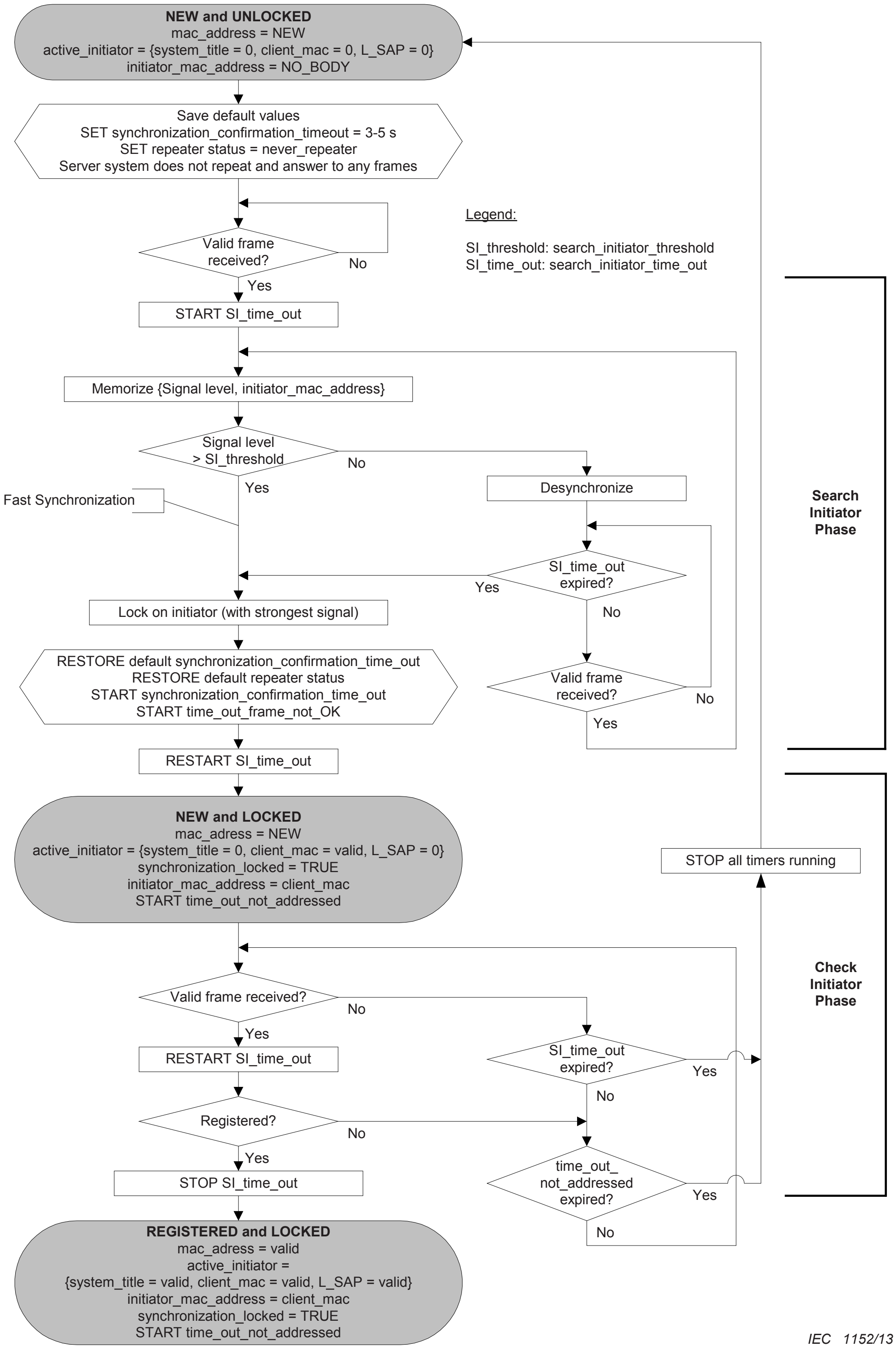
10.7.2.2 Process parameters

The Intelligent Search Initiator process is characterized by two parameters:

- The search_initiator_time_out, defining the duration of the Search Initiator Phase. This timeout is also used during the Check Initiator Phase, see 10.7.2.4;

  The Search Initiator Phase shall be long enough to allow the server systems to hear all the initiators around them and, when needed, to let sufficient time to start all the initiators by the Head End System. However, this time should not be too long either, so that the discovery phase could be executed correctly. The recommended value for this timeout is 10 minutes;
- The search_initiator_threshold, defining the minimum signal level allowing a fast synchronization.

The value of the search_initiator_threshold should be chosen so that the server systems next to an initiator lock on it immediately, but the server systems a bit further away (maybe on another network) do only lock on it after the search_initiator_time_out timer expires. The default value is 98 dBµV.



NOTE A valid frame is a frame in which either the source or the destination address is an initiator address, and otherwise correct.

Figure 4 – Intelligent Search Initiator process flow chart

10.7.2.3 Search Initiator Phase

Initially, the server system is in the NEW and UNLOCKED state waiting to receive a valid frame.

For the Search Initiator Phase, the synchronization_confirmation_time_out shall be reduced to 3-5 s. Otherwise, a server system would remain synchronized too long on a bad frame.

During this phase, a server system shall not repeat any frames. This is because if repetition was allowed, it would have to repeat all frames, not only the frames from the closest initiator, and so the other server systems next to it would listen to frames from a bad initiator with a strong signal level. Therefore, the Intelligent Search Initiator algorithm can only be efficient if all the server systems on a network have the algorithm implemented (otherwise, other server systems could repeat frames and foul the signal level).

For the same reasons, a server system shall not transmit any frames during the Search Initiator Phase:

- it should not answer to a Discover request (It should be desynchronised before, except if the signal level is good enough to allow fast synchronization. But in this case, the server system is not in the Search Initiator Phase anymore but in the Check Initiator Phase);
- it should not answer to a Register request either;
- and in particular, it should not answer any ACSE or xDLMS service requests (this is obvious because the server system is in the NEW state).

Notice that as soon as a server system becomes locked, it can repeat frames since it will only accept frames from the good initiator. (It is even advised that it repeats frames, since it will shorten the Search Initiator Phase for the other server systems).

Each time that the server system receives a frame, it checks the signal level and the MAC addresses in it:

- if none of the MAC addresses – source or destination – is an initiator MAC address, the frame is considered as invalid; the server system immediately desynchronises in order to listen to another frame;
- if one of the MAC addresses is an initiator MAC address, and the signal level is good enough (signal level > search_initiator_threshold – see 10.7.2.2) the server system locks on that initiator. This is called Fast Synchronization. This occurs when the server system is next to an initiator or to a server system that is already registered to that initiator. The server system enters the Check Initiator Phase;
- if one of the MAC addresses is an initiator MAC address but the signal level is not good enough (signal level < search_initiator_threshold), the server system memorizes the signal level and the MAC address, then desynchronizes immediately in order to listen to another frame;
- when the search_initiator_time_out expires, the server system locks to the initiator having provided the best signal level.

At this point:

- the default values of the synchronization_confirmation_time_out and the repeater status are restored;
- the synchronization_confirmation_time_out and the time_out_frame_not_OK timers are initialised;
- the search_initiator_time_out is restarted.

The server is in the NEW and LOCKED state and enters the Check Initiator Phase.

10.7.2.4 Check Initiator Phase

Once the server system is locked, it can be discovered and registered. The process is the following:

- the time_out_not_addressed timer is started;

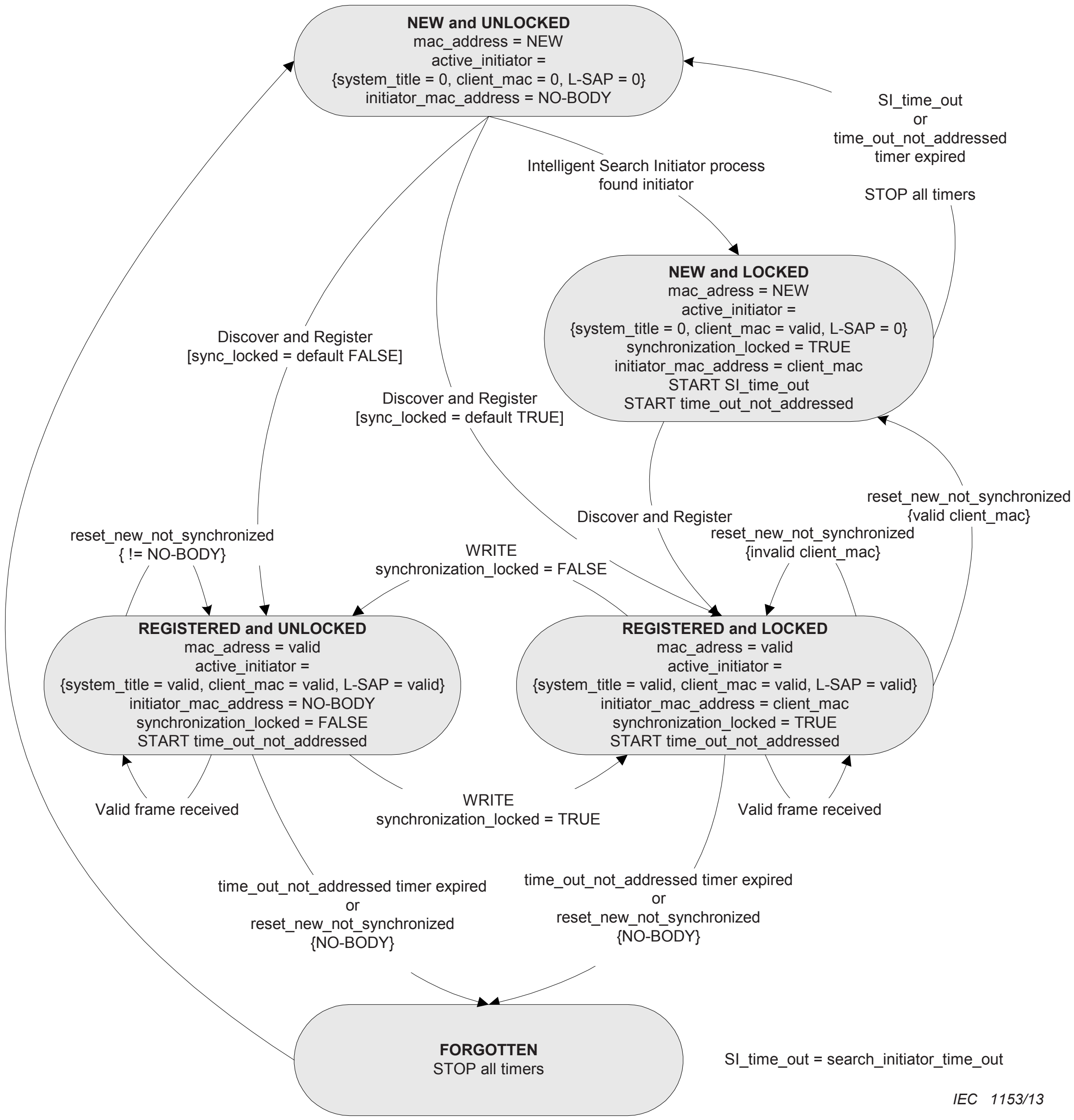
- the server waits then to be discovered and registered;
- the search_initiator_time_out timer is restarted each time a valid frame is received;
- when the server system is registered (valid frame carrying a Register CI-PDU received):
  - the search_initiator_time_out timer is stopped;
  - the active_initiator variable is set;
- if either the search_initiator_time_out or the time_out_not_addressed timer expires, it means that the server system did not receive a frame from or to its initiator for a long time: all timers are stopped then and the server system returns to the initial state: NEW and UNLOCKED.

10.7.2.5 Remarks

The Intelligent Search Initiator process has to be used cautiously. If a server system hears a frame with a wrong MAC address, it will lock on it and won't unlock until the search_initiator_time_out is over. It is advised not to change the initiator MAC address of an NNAP (concentrator) unless it can be made sure that all the server systems on the network are in the unconfigured state: NEW and UNLOCKED.

10.8 The Discovery and Registration process

The Discovery and Registration process, including the Intelligent Search Initiator process, is summarized in Figure 5.



NOTE The state transitions caused by writing the mac-address and initiator-mac-address variables are not shown.

Figure 5 – The Discovery and Registration process

The process is the following.

Initially, the server is in the NEW and UNLOCKED (UNCONFIGURED) state. In this state:

- mac_address = NEW;
- active_initiator: default value, with all three elements set to 0:
  - system_title = octet string of 0;
  - (client_)mac_address = NO-BODY;
  - L-SAP_selector = 0;
- initiator_mac_address = NO-BODY.

The default value of the synchronization_locked variable is as specified in the companion specification:

a) FALSE: the value of initiator_mac_address variable is always NO-BODY;
b) TRUE: the value of the initiator_mac_address variable follows the value of the (client_)mac_address element of the active_initiator variable;
c) The Intelligent Search Initiator process is used. In this case, the value of the search_initiator_time_out variable shall be different from 0. The server moves from the NEW and UNLOCKED state to the NEW and LOCKED state at the end of the Search Initiator Phase.

NOTE This third possibility is an extension to IEC 61334-4-511.

When a server is installed, it has to be discovered and registered.

In case a), upon the reception of a valid Register CIASE PDU the server moves to the REGISTERED and UNLOCKED state:

- the mac_address variable is set to the value allocated by the initiator;
- the elements of the active_initiator variable are set to the values contained in the Register CIASE PDU and the DL-Data.indication LLC PDU:
  - system_title: system title of the initiator;
  - (client_)MAC_address: MAC address of the initiator;
  - L-SAP_selector: the L-SAP used by the initiator;
- the initiator_MAC_address remains at NO-BODY;
- the synchronization_locked variable remains at FALSE;
- the time_out_not_addressed timer is started.

In case b), upon the reception of a valid Register CIASE PDU, the server moves to the REGISTERED and LOCKED state:

- the mac_address variable is set to the value allocated by the initiator;
- the elements of the active_initiator variable are set to the values contained in the Register CIASE PDU and the DL-Data.indication LLC PDU:
  - system_title: system title of the initiator;
  - (client_)mac_address: MAC address of the initiator;
  - L-SAP_selector: the L-SAP used by the initiator;
  - the initiator_MAC_address is updated to be equal to the (client_)MAC_address element of the active_initiator;
  - the synchronization_locked attribute remains at TRUE;
  - the time_out_not_addressed timer is started.

In case c), the server searches first for the initiator providing the strongest signal; see Figure 4. At the end of Search Initiator Phase, the search_initiator_time_out is re-started, the time_out_not_addressed timer is started, and the server locks on the initiator chosen:

- the (server_)mac_address variable is still at NO-BODY (server is in NEW state);
- the synchronization_locked variable is set to TRUE;
- the (client_)MAC_address element of the active_initiator variable takes the MAC_address of the initiator chosen. The other elements of the active_initiator variable remain at their default value;
- the initiator_MAC_address is updated to be equal to the (client_)MAC_address element of the active_initiator variable.

The server is in the NEW and LOCKED state and enters the Check Initiator Phase.

The server can only be registered by the initiator chosen. This takes place exactly as in case b). If the server is not registered before either the search_initiator_time_out or the time_out_not_addressed timer expires, it returns to the NEW and UNLOCKED state. All timers are stopped.

In the REGISTERED state, the server may receive frames addressed to it and respond to them. The time_out_not_addressed timer is restarted with each frame addressed to the server.

The server may also move from the REGISTERED and UNLOCKED state to the REGISTERED and LOCKED state and vice versa by writing the value of the synchronization_locked variable.

The server may leave the REGISTERED state either:

– by the expiration of the time_out_not_addressed timeout; or
– by writing the reset_new_not_synchronized variable.

In the first case, the server becomes “FORGOTTEN”: it loses its MAC address and its active initiator: the mac_address, the initiator_mac_address and the active_initiator variables are all reset. The server returns to the NEW and UNLOCKED state.

In the second case, the server checks first the value of the client_mac_address submitted as a parameter of the reset_new_not_synchronized request:

- if the value is not equal to a valid client address, or the pre-defined NO-BODY address, the writing is refused;
- if the value is NO-BODY, it has the same effect as the expiration of the time_out_not_addressed timer: the server returns to the NEW and UNLOCKED state;
- if the value is a valid client address, then the value of the synchronization_locked variable is checked:
  – if it is FALSE – the server is in the REGISTERED and UNLOCKED state – the writing is refused;
  – if it is TRUE – the server is in the REGISTERED and LOCKED state – the (client_)mac_address element of the active initiator variable is set to the value submitted, the system_title and the L-SAP_selector elements are reset to 0. The initiator_mac_address variable is also set to the value submitted. The time_out_not_addressed timer is stopped. The server returns to the NEW and LOCKED state, where it waits to be registered again by the initiator it has been reset to.

When the server leaves the REGISTERED state, all AAs are aborted.

If a power failure occurs, it is managed as follows:

- if the server is in the NEW and UNLOCKED state, it will stay there when the power returns;
- if the server is in the Search Initiator Phase, then it moves back to the NEW and UNLOCKED state when the power returns. The search_initiator_time_out timer is stopped and all data concerning the initiators heard so far (signal level and MAC address) are lost;
- if the server is in the NEW and LOCKED state, it will stay there when the power returns. The search_initiator_time_out and the time_out_not_addressed timers are restarted;
- if the server is in the REGISTERED (LOCKED or UNLOCKED) state, it will stay there when the power returns. The time_out_not_addressed timer is restarted. The Application Associations open before the power failure are locally re-established.

10.9 Abstract and transfer syntax

As specified in IEC 61334-4-511:2000, 7.3.1, the CIASE uses the ASN.1 abstract syntax. See also Clause 14. The transfer syntax is A-XDR.

11 Addressing

11.1 General

In the DLMS/COSEM S-FSK PLC profile, two levels of addresses are defined:

- at the MAC sublayer that processes MAC addresses to access an LLC entity;
- at the LLC sublayer that processes LLC addresses to access application entities.

11.2 IEC 61334-5-1 MAC addresses

A physical client or server – initiator or responder – system may be accessed using a MAC address specific to the system or by using one of the group MAC addresses (see Table 6).

Table 6 – MAC addresses

Address	Value	Reference
NO-BODY	000	IEC 61334-4-1:1996 4.3.2.6, IEC 61334-5-1:2001 4.3.7.5.1
Local MAC	001…FIMA-1	IEC 61334-4-1 :1996 4.3.2.5
Initiator	FIMA…LIMA	IEC 61334-4-1:1996 4.3.2.4
MAC group address	LIMA + 1…FFB	
All-configured	FFC	IEC 61334-4-1:1996 4.3.2.1, IEC 61334-5-1:2001 4.3.7.5.2
NEW	FFE	IEC 61334-4-1:1996 4.3.2.2, IEC 61334-5-1:2001 4.2.3.2
All Physical	FFF	IEC 61334-4-1:1996 4.3.2.3, IEC 61334-5-1:2001 4.3.7.5.3
NOTE MAC addresses are expressed on 12 bits. FIMA = First initiator MAC address; C00 LIMA = Last initiator MAC address; DFF		

11.3 Reserved special LLC addresses

NOTE Subclause 11.3 is based on IEC 61334-4-1:1996, 4.4.

11.3.1 General

Each application process within the physical device is bound to a data link layer address that consists of the doublet {MAC-address, L-SAP}. The following LLC addresses (L-SAPs) are specified:

- All-L-SAP: designates the group consisting of all L-SAPs actively serviced by the underlying MAC layer (with its specified MAC address);
- system management L-SAP (M-L-SAP): there is only one M-L-SAP in a physical system;
- initiator L-SAPs (I-L-SAP): These are defined in a specific range;
- individual LLC addresses: These are defined in a specific range;
- CIASE L-SAP (C-L-SAP).

11.3.2 Reserved addresses for the IEC 61334-4-32 LLC sublayer

The reserved LLC addresses for the IEC 61334-4-32 LLC sublayer on the client side and the server side are shown in Table 7 and

Table 8 respectively.

Table 7 – Reserved IEC 61334-4-32 LLC addresses on the client side

Address	L-SAP	Meaning
0x00		No-station
0x01	M-L-SAP I-L-SAP	Client management process. The CIASE is also bound to this address.
0x10		Public client (lowest security level)

Table 8 – Reserved IEC 61334-4-32 LLC addresses on the server side

Address	L-SAP	Meaning
0x00	I-L-SAP	CIASE
0x01	M-L-SAP	Management logical device
0x02…0x0F		Reserved for future use
0xFF	All-L-SAP	All-station (Broadcast)

11.3.3 Reserved addresses for the HDLC based LLC sublayer

The reserved LLC addresses for the HDLC based LLC sublayer on the client side and the server side are shown in Table 9 and

Table 10 respectively.

Table 9 – Reserved HDLC based LLC addresses on the client side

Address	L-SAP	Meaning
0x00		No-station
0x01	M-L-SAP I-L-SAP	Client management process. The CIASE is also bound to this address.
0x10		Public client

Table 10 – Reserved HDLC based LLC addresses on the server side

One byte address	Two bytes address	Meaning
0x00	0x0000	No-Station. The CIASE is also bound to this address.
0x01	0x0001	Management logical device
0x02…0x0F	0x0002…0x000F	Reserved for future use
0x7E	0x3FFE	Calling physical address, not used in the PLC S-FSK HDLC based profile.
0x7F	0x3FFF	All-station (Broadcast)

11.3.4 Source and destination APs and addresses of CI-PDUs

The Source and destination APs and addresses of CI-PDU requests are shown in Table 11.

Table 11 – Source and Destination APs and addresses of CI-PDUs

CI-PDU	Source AP	Destination AP	MAC SA	MAC DA	D-L-SAP	S-L-SAP
Discover	Initiator	AllSMAE	Initiator	FFF All-Physical	0x00 CIASE	0x01 CIASE [a]
DiscoverReport	Manager	AllSMAE	NEW FFE [b] or individual server MAC	FFF [c] All-Physical or Initiator	0xFD	0x00 CIASE
Register	Initiator	AllSMAE	Initiator	FFF All-Physical	0x00 CIASE	0x01 CIASE [d]
PingRequest	Initiator	Individual	Initiator	Individual	0x00 CIASE	0x01 CIASE [d]
PingResponse	Individual	Initiator	Individual	Initiator	0x01 CIASE [d]	0x00 CIASE
RepeaterCall	Initiator	AllSMAE	Initiator	FFF All-Physical	0x00 CIASE	0x01 CIASE [d]
ClearAlarm	Initiator	AllSMAE	Initiator	FFF All-Physical	0x00 CIASE	0x01 CIASE [d]

NOTE 1 In the MAC frame, the order of the addresses is Source Address – Destination Address.

NOTE 2 In the IEC 61334-4-32 LLC frame, the order of the addresses is Destination Address – Source Address.

NOTE 3 In the HDLC frame, the order of the addresses is Destination Address – Source Address.

[a] Could be a different value.

[b] FFE if the server is in the NEW state. Individual MAC address if the server is in ALARM state.

[c] If the reporting system list feature is used, then the Destination Address is All-Physical. Otherwise, it is the Initiator address.

[d] Could be a different value, but shall be the same as in the Discover CI-PDU.

12 Specific considerations / constraints for the IEC 61334-4-32 LLC sublayer based profile

12.1 Establishing application associations

AAs can only be established with server systems properly registered by the initiator. The MSC for the discovery and registration process is shown in Figure 6.

NOTE In the following examples, the NNAP acts as a COSEM client.

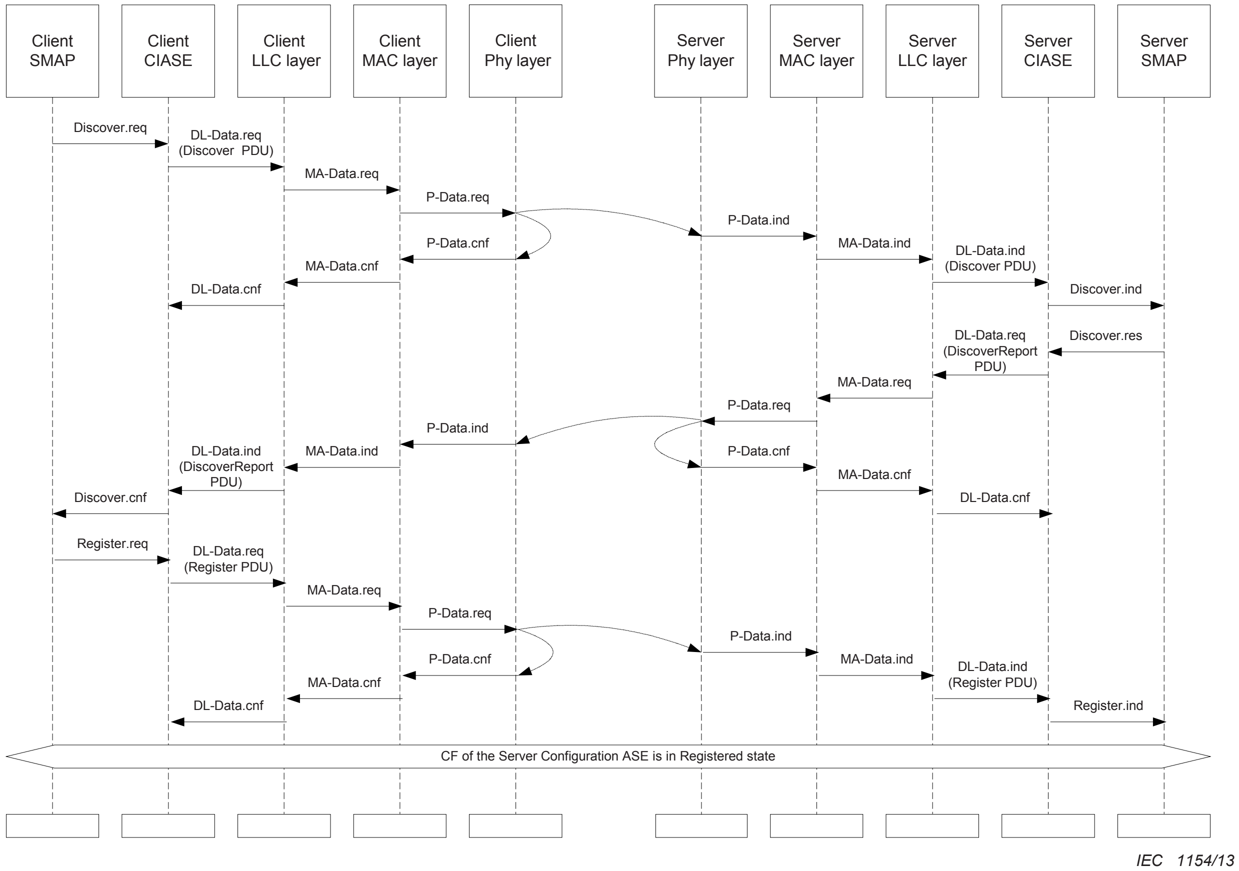


Figure 6 – MSC for the discovery and registration process

The MSC for the establishment of a confirmed AA establishment is shown in Figure 7.

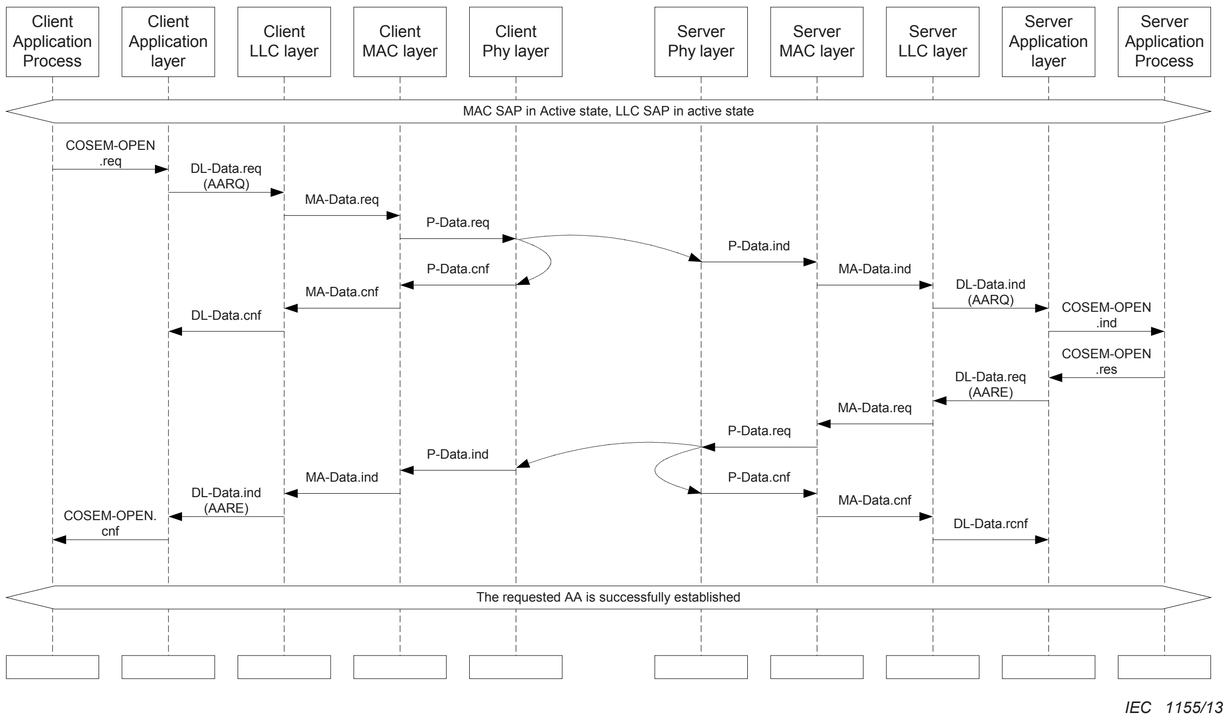


Figure 7 – MSC for successful confirmed AA establishment

12.2 Application association types, confirmed and unconfirmed xDLMS services

Table 12 shows the rules for establishing confirmed and unconfirmed AAs. In this table, grey areas represent cases, which are out of the normal operating conditions: either not allowed or have no useful purpose. According to this:

It is not allowed to request an xDLMS service in a confirmed way (Service_Class = Confirmed) within an unconfirmed AA. This is prevented by the Client AL. Servers receiving such APDUs shall simply discard them, or shall send back a ConfirmedServiceError APDU or – if the feature is implemented – send back the optional ExceptionResponse APDU.

In this profile, the Service_Class parameter of the COSEM-OPEN service is linked to the response-allowed parameter of the xDLMS InitiateRequest APDU. If the COSEM-OPEN service is invoked with Service_Class == Confirmed, the response-allowed parameter shall be set to TRUE. The server is expected to respond. If it is invoked with Service_Class == Unconfirmed, the response-allowed parameter shall be set to FALSE. The server shall not send back a response.

The Service_Class parameter of the GET, SET and ACTION services is linked to the service-class bit of the Invoke-Id-And-Priority byte. If the service is invoked with Service_Class = Confirmed, the service-class bit shall be set to 1, otherwise it shall be set to 0.

Table 12 – Application associations and data exchange in the S-FSK PLC profile using the connectionless LLC sublayer

<table>
<tr><th colspan="4">Application association establishment</th><th colspan="2">Data exchange</th></tr>
<tr><th>Protocol connection parameters</th><th>COSEM-OPEN service class</th><th>Use</th><th>Type of established AA</th><th>Service class</th><th>Use</th></tr>
<tr><td rowspan="4">Id:
LLC addresses, MAC addresses</td><td rowspan="2">Confirmed</td><td rowspan="2">Exchange AARQ/AARE APDU-s transported by DL-Data services</td><td rowspan="2">Confirmed</td><td>Confirmed</td><td>DL-Data services</td></tr>
<tr><td>Unconfirmed</td><td>DL-Data services</td></tr>
<tr><td rowspan="2">Unconfirmed</td><td rowspan="2">Send AARQ transported by DL-Data services</td><td rowspan="2">Unconfirmed</td><td>Confirmed (not allowed)</td><td>–</td></tr>
<tr><td>Unconfirmed</td><td>DL-Data services</td></tr>
</table>

12.3 xDLMS client/server type services

No specific features / constraints apply related to the use of client/server type services.

The MSC is essentially the same as for establishing confirmed AAs, except that instead of the COSEM-OPEN service primitives, the appropriate xDLMS service primitives are used.

12.4 Releasing application associations

As the LLC sublayer supporting the COSEM Application layer is connectionless, the COSEM-Release service may be invoked with the Use_RLRQ_RLRE option = TRUE to release an AA.

To secure the RLRQ APDU against denial-of-service attacks – executed by unauthorized releasing of the AA – the user-information field of the RLRQ APDU may contain the xDLMS InitiateRequest APDU, authenticated and encrypted using the AES-GCM-128 algorithm, the global unicast encryption key and the authentication key (the same as in the AARQ APDU).

The MSC for releasing an AA is shown in Figure 8.

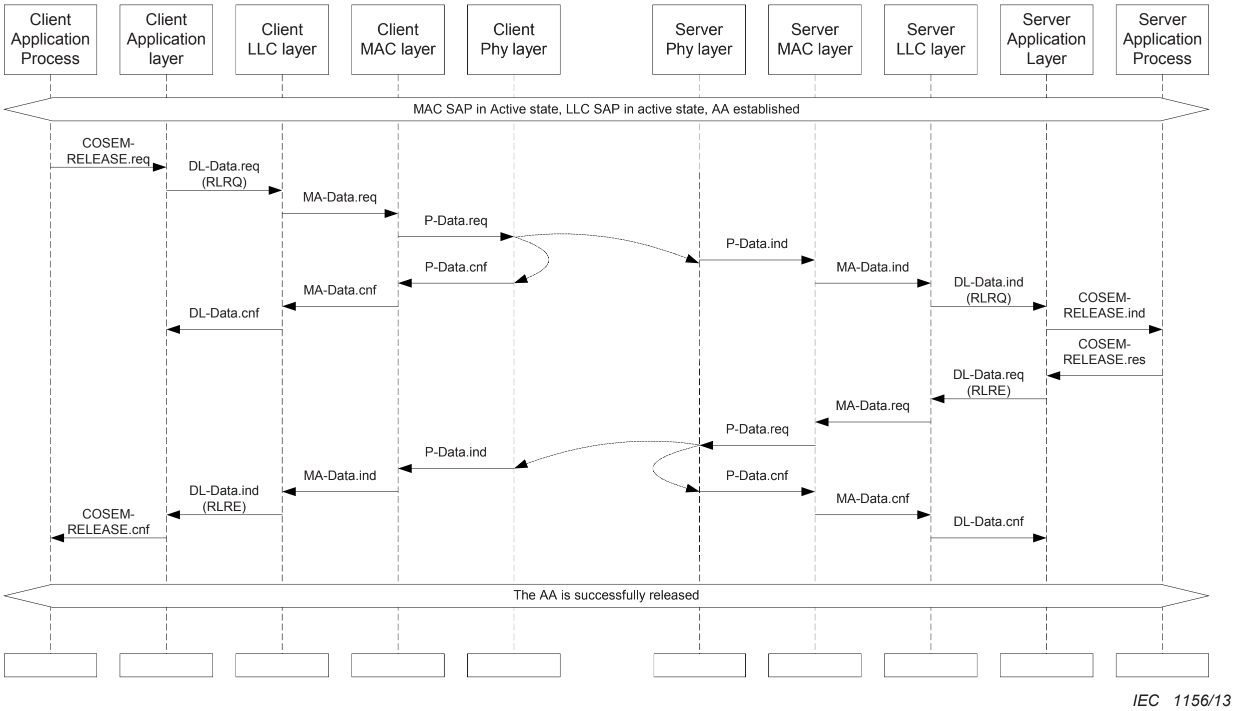


Figure 8 – MSC for releasing an Application Association

12.5 Service parameters of the COSEM-OPEN / -RELEASE / -ABORT services

The optional User_Information parameters of the COSEM-OPEN / -RELEASE services are not supported in this communication profile.

12.6 The EventNotification service and the TriggerEventNotificationSending service

The EventNotification (LN referencing) / InformationReport (SN referencing) services are supported by the DL-Update-Reply and DL-Reply services of the LLC sublayer:

- in the case of LN referencing, the EventNotificationRequestAPDU shall be placed into the LLC sublayer using the DL-Update-Reply.request service;
- in the case of SN referencing, the InformationReportRequest APDU shall be placed into the LLC sublayer using the DL-Update-Reply.request service;
- these APDUs are available then for any client until they are cleared by placing an empty APDU.

The length of the APDUS shall not exceed the limitation imposed by the LLC / MAC sublayers.

The MSC for an EventNotification service is shown in Figure 9.

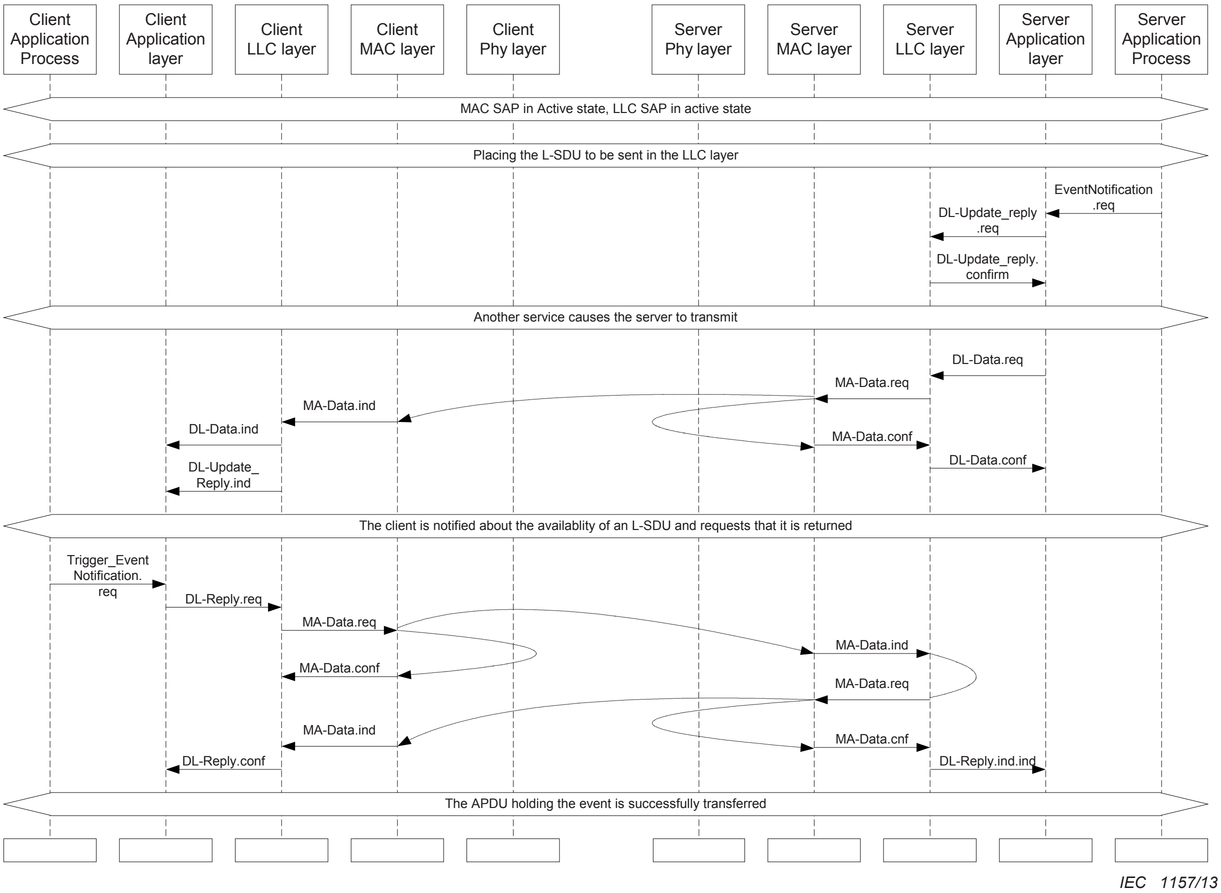


Figure 9 – MSC for an EventNotification service

12.7 Transporting long messages

In the S-FSK profile, the IEC 61334-4-32 LLC sublayer imposes a limitation on the length of the APDU that can be transported. For transporting long messages, application layer block transfer is available.

12.8 Broadcasting

Broadcast messages can be sent by the data NNAP (concentrator), acting as a client, to servers using broadcast addresses.

13 Specific considerations / constraints for the HDLC LLC sublayer based profile

13.1 Establishing Application Associations

AAs can only be established with server systems, which have been properly registered by the initiator.

The MSC for the discovery and registration process is shown in Figure 10.

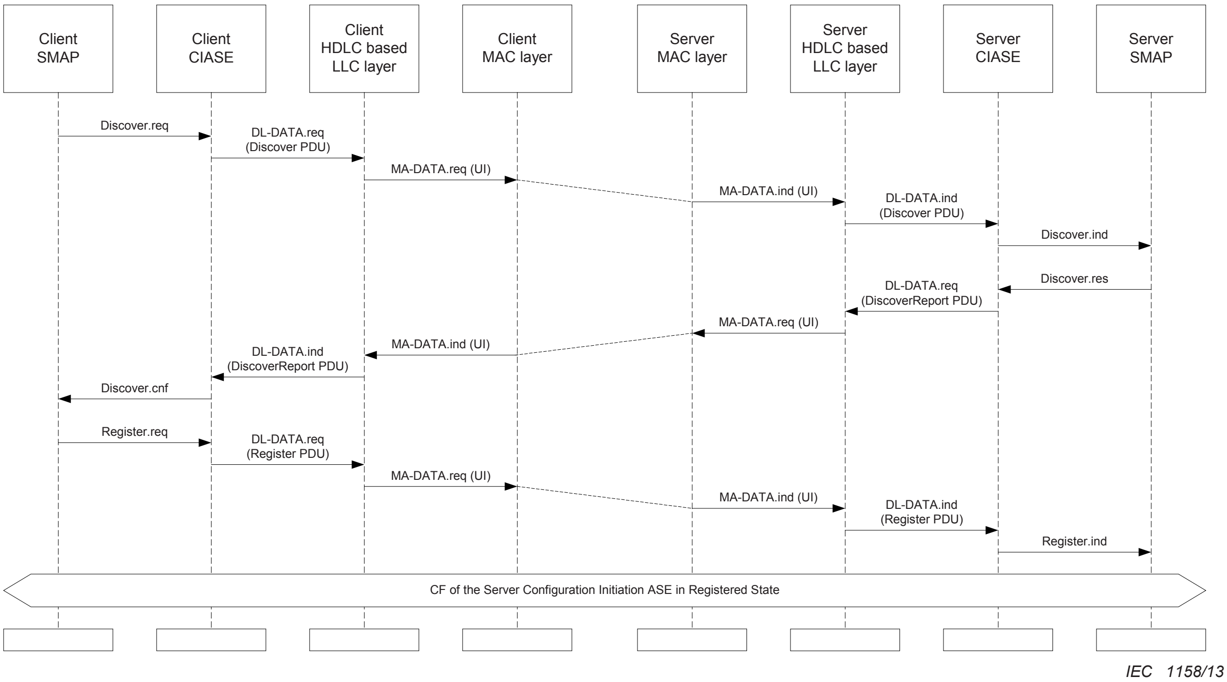


Figure 10 – MSC for the Discovery and Registration process

The MSC for the establishment of a confirmed AA establishment is shown in the upper part of Figure 11.

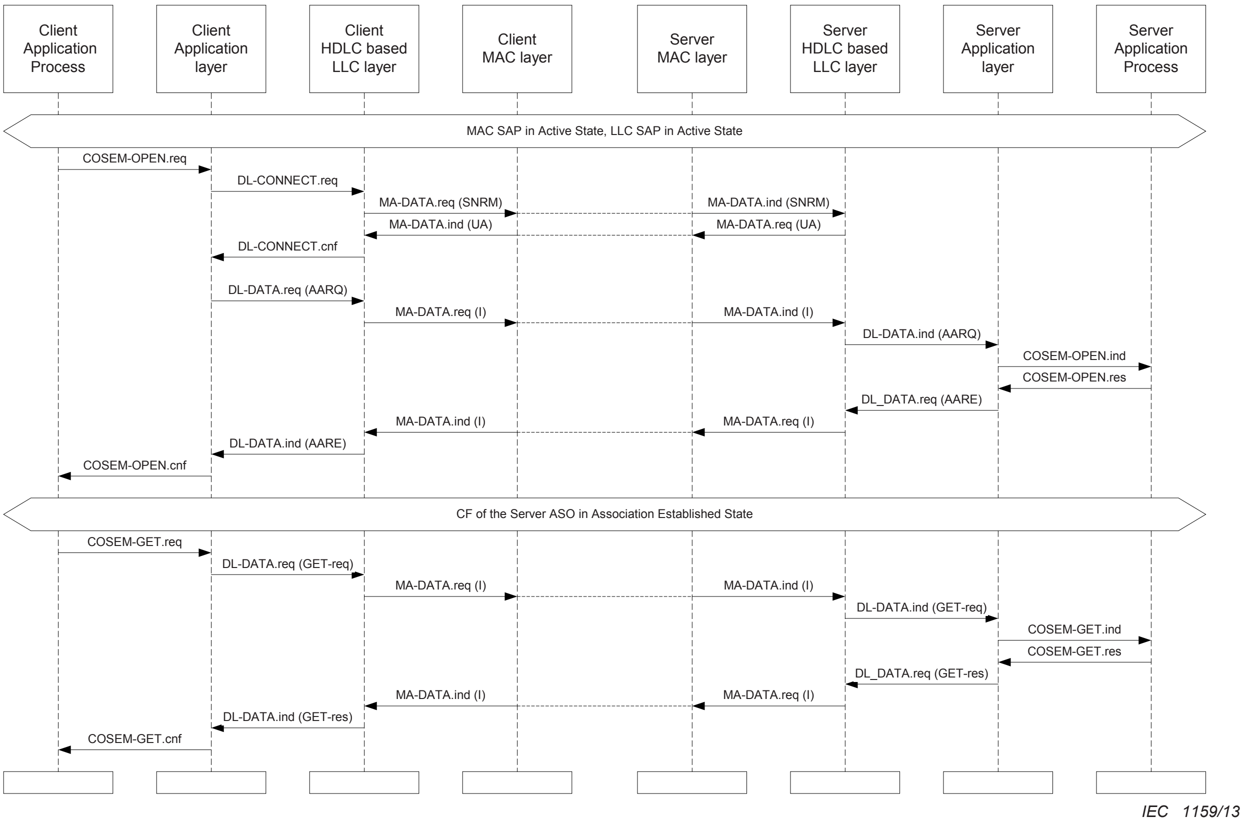


Figure 11 – MSC for successful confirmed AA establishment and the GET service

13.2 Application association types, confirmed and unconfirmed xDLMS services

The rules of the 3-layer, connection-oriented, HDLC based profile apply.

13.3 xDLMS client/server type services

The MSC for a COSEM GET.request service – preceded with the establishment of an AA – is shown in lower part of Figure 11.

13.4 Correspondence between AAs and data link layer connections, releasing AAs

The rules of the 3-layer, connection-oriented, HDLC based profile apply.

13.5 Service parameters of the COSEM-OPEN/ -RELEASE/ -ABORT services

The rules of the 3-layer, connection-oriented, HDLC based profile apply.

13.6 The EventNotification service and protocol

The rules of the 3-layer, connection-oriented, HDLC based profile apply, except that the EventNotificationRequestAPDU can be sent only when the server is registered and synchronized with the initiator.

13.7 Transporting long messages

The rules of the 3-layer, connection-oriented, HDLC based profile apply.

13.8 Broadcasting

Broadcast messages can be sent by the data NNAP (concentrator), acting as a client, to servers using broadcast addresses.

14 Abstract syntax of CIASE APDUs

```
CIASEpdu Definitions::= BEGIN

PingRequestPDU    ::= SEQUENCE
{
    system-title-server                System-Title
}

PingResponsePDU   ::= SEQUENCE
{
    system-title-server                System-Title
}

RegisterPDU       ::= SEQUENCE
{
    active-initiator-system-title     System-Title,
    list-of-correspondence            Correspondence-List
}

DiscoverPDU       ::= SEQUENCE
{
     response-probability              INTEGER(0..100),
     allowed-time-slots                INTEGER(0..32767),
     -- see IEC 61334-5-1 for the value of MAX_INITIAL_CREDIT
     DiscoverReport-initial-credit     INTEGER(0..MAX_INITIAL_CREDIT),
     ICEqualCredit                     INTEGER(0..1)
}

DiscoverReportPDU ::= SEQUENCE
{
      -- the first one of this list is the system-title of the reporting system
      system-title-list     System-Title-List,

      -- alarm-descriptor of the reporting system
      alarm-descriptor      Alarm-Descriptor OPTIONAL
}

RepeaterCallPDU   ::= SEQUENCE
{
```

```
      max-adr-mac              INTEGER (0..4095),
      nb-Tslot-for-new         INTEGER (0..255),
      reception-threshold      INTEGER (0..255) DEFAULT 104
}

ClearAlarmPDU::= CHOICE
{
     -- clears a single alarm in all servers
     alarm-descriptor                       [0] Alarm-Descriptor,

     -- clears a list of alarms in all servers
     alarm-descriptor-list                  [1] Alarm-Descriptor-List,

     -- clears a common list of alarms specified in the list of servers specified
     alarm-descriptor-list-and-server-list  [2] SEQUENCE
     {
          server-id-list              System-Title-List,
          alarm-descriptor-list       Alarm-Descriptor-List
     }

     -- clears one alarm specified in each server specified
     alarm-descriptor-by-server-list        [3] Alarm-Descriptor-By-Server-List
}

-- Useful types used with the S-FSK PLC profile

-- SYSTEM-TITLE-SIZE shall be specified by the naming authority
System-Title       ::= OCTET STRING (SIZE(SYSTEM-TITLE-SIZE))

System-Title-List              ::= SEQUENCE OF System-Title

MAC-address                    ::= INTEGER(0..4095)




Correspondence            ::= SEQUENCE
{
    new-system-title    System-Title,
    mac-address         MAC-address
}

Correspondence-List            ::= SEQUENCE OF Correspondence

Alarm-Descriptor               ::= INTEGER (0..255)

Alarm-Descriptor-List          ::= SEQUENCE OF Alarm-Descriptor

Alarm-Descriptor-By-Server     ::= SEQUENCE
{
      server-id            System-Title,
      alarm-descriptor     Alarm-Descriptor
}

Alarm-Descriptor-By-Server-List::= SEQUENCE OF Alarm-Descriptor-By-Server

-- The following has only local significance on the client side.

CIASELocalError::= ENUMERATED
{
     Other                                  (0),
     Discover-probability-out-of-range      (1),
     Discover-initial-credit-out-of-range   (2),
     DiscoverReport-list-too-long           (3),
     Register-list-too-long                 (4),
     ICEqualCredit-out-of-range             (5),
     Ping-no-response                       (6),
     Ping-system-title-nok                  (7)
}
END
```

Annex A
(informative)

S-FSK PLC encoding examples

A.1 CI-PDUs, ACSE APDUs and xDLMS APDUs carried by MAC frames using the IEC 61334-4-32 LLC sublayer

In these examples, the following communication sequence is shown, when the DLMS/COSEM PLC S-FSK profile is used with the IEC 61334-4-32 LLC sublayer:

- the initiator Discovers, then Registers a new server system;
- the initiator establishes an AA;
- it reads the time attribute of the Clock object (once and 13 times, to show block transfer);
- the initiator Pings a server;
- the initiator sends a RepeaterCall service.

In these examples: SYSTEM-TITLE-SIZE = 6.

The traces have been taken from a protocol analyser. The contents of the MAC frame are explained. The MAC frame is shown between the brackets () following the "02 xx 50" header and followed by 00 00 (final field, normally a frame check). The Pad fields are not shown.

Server in the NEW state with one alarm (Meter New): MAC frame carrying a Discover CI-PDU

```
17:15:35:645  ===>  Discover.Request(MAC:C00/FFF  Ic:7  Dc:0  LLC:0/1)
(Prob:100 NbTslot:10 CreditReponse:0 ICequalCredit:0)
Hex: 02 11 50 ( FC C0 0F FF 11 90 00 01 1D 64 00 0A 00 00 ) 00 00

-- Explanation:

FC // Credit fields: 1111 1100 IC = 7, CC = 7, DC = 0
C0 0F FF // MAC addresses: SA = C00 (Initiator), DA = FFF (All-
            Physical)
11 // Pad length
90 // Control byte 1001 0000, DL-Data.request
00 01 // L-SAPs: DA = 00 (CIASE server), SA SAP = 01 (CIASE client)
1D // DiscoverRequest PDU
     64 // response-probability = 100
     00 0A // allowed-time-slots = 10
     00 // DiscoverReport-initial_credit = 00
     00 // ICEqualCredit = 00
```

MAC frame carrying a DiscoverReport CI-PDU

```
17:15:40:441  <==  Alarm.Report(MAC:FFE/FFF  Ic:0  Dc:0  LLC:FD/0)    SN:
040890000001
Hex: 02 15 50 ( 00 FF EF FF 0D 90 FD 00 1E 01 04 08 90 00 00 01 01
01 ) 00 00

-- Explanation:

00 // Credit fields
FF EF FF // MAC addresses: SA = FFE (NEW), DA address = FFF
0D // Pad length
90 // DL-Data.request
```

```
FD 00 // L-SAPs: DA = FD, SA = 00
1E // DiscoverReport CI-PDU
     01 // SEQUENCE OF 1
          04 08 90 00 00 01 // System-Title
     01 // Alarm-Descriptor presence flag
     01 // Alarm-Descriptor
```

Register service: MAC frame carrying a Register CI-PDU

```
17:15:41:129 ===> Register(MAC:C00/FFF Ic:7 Dc:0 LLC:0/1) (AddrMAC:
0x3 SN: 040890000001 )
Hex: 02 1B 50 ( FC C0 0F FF 07 90 00 01 1C 04 08 99 00 00 01 01 04 08
90 00 00 01 00 03 ) 00 00

-- Explanation:

FC // Credit fields: 1111 1100 IC = 7, CC = 7, DC = 0
C0 0F FF // MAC addressees: SA = C00, DA = FFF
07 // Pad length
90 // DL-Data.request
00 01 // L-SAPs: DA = 00, SA = 01
1C // RegisterRequest CI-PDU
     04 08 99 00 00 01 // active-initiator-system-title
     01 // SEQUENCE OF 1
          04 08 90 00 00 01 // new-system-title
          00 03 // mac-address 0x03
```

Server in registered state with an alarm: MAC frame carrying a DiscoverRequest CI-PDU

```
17:17:02:973 ===> Discover.Request(MAC:C00/FFF Ic:7 Dc:0 LLC:0/1)
(Prob:100 NbTslot:10 CreditReponse:0 ICequalCredit:0)
Hex: 02 11 50 ( FC C0 0F FF 11 90 00 01 1D 64 00 0A 00 00 ) 00 00

-- Explanation:

FC // Credit fields: 1111 1100 IC = 7, CC = 7, DC = 0
C0 0F FF // MAC addresses: SA = C00, DA = FFF
11 // Pad length
90 // DL-Data.request
00 01 // L-SAPs: DA = 00, SA = 01
1D // DiscoverRequest CI-PDU
     64 // response-probability = 100
     00 0A // allowed-time-slots 10
     00 // DiscoverReport-Initial-Credit 00
     00 // ICEqualCredit 0
```

Response: MAC frame carrying a DiscoverResponse CI-PDU

```
 17:17:07:316 <== Alarm.Report(MAC:003/FFF Ic:0 Dc:0 LLC:FD/0)   SN:
040890000001
Hex: 02 15 50 ( 00 00 3F FF 0D 90 FD 00 1E 01 04 08 90 00 00 01 01
82 ) 00 00

-- Explanation:

00 // Credit fields
00 3F FF // MAC addresses: SA = 003, DA = FFF
0D // Pad length
90 // DL-Data.request
FD 00 // L-SAPs: DA = FD, SA = 00
1E // DiscoverReport CI-PDU
     01 // SEQUENCE OF 1
```

```
          04 08 90 00 00 01 // System-Title
     01 // alarm-descriptor presence flag
     82 // alarm-descriptor
```

Open association on the Logical device LsapDest=0x01 and Client R/W LsapSrc=0x02: MAC frame carrying an AARQ APDU

```
17:28:52:691 ===> AARQ.Request(MAC:C00/003 Ic:4 Dc:0 LLC:1/2)
 Hex: 02 43 50 ( 90 C0 00 03 03 90 01 02 60 36 A1 09 06 07 60 85 74 05
08 01 02 8A 02 07 80 8B 07 60 85 74 05 08 02 01 AC 0A 80 08 31 32 33
34 35 36 37 38 BE 10 04 0E 01 00 00 00 06 5F 1F 04 00 1C 1A 20 00 EF )
00 00

--Explanation:

90 // Credit fields
C0 00 03 // MAC addresses: SA = C00, DA = 003
03 // Pad length
90 // DL-Data.request
01 02 // L-SAPs: DA = 0x01, SA = 0x02
60 36 // AARQ APDU
     A1 09 06 07 60 85 74 05 08 01 02 // application-context-name
     8A 02 07 80 // acse-requirements
     8B 07 60 85 74 05 08 02 01 // mechanism-name
     AC 0A 80 08 31 32 33 34 35 36 37 38 // calling-authentication-
                                            value
     BE 10 04 0E 01 00 00 00 06 5F 1F 04 00 1C 1A 20 00 EF
     // user-information xDLMS InititateRequest
```

Response: MAC frame carrying an AARE APDU

```
17:28:54:191 <=== AARE.Response(MAC:003/C00 Ic:4 Dc:0 LLC:2/1)
Hex: 02 37 50 ( 90 00 3C 00 0F 90 02 01 61 29 A1 09 06 07 60 85 74 05
08 01 02 A2 03 02 01 00 A3 05 A1 03 02 01 00 BE 10 04 0E 08 00 06 5F
1F 04 00 1C 1A 20 00 EF FA 00 00 ) 00 00

-- Explanation:

90 // Credit fields
00 3C 00 // MAC addresses: SA = 003, DA = C00
0F // Pad length
90 // DL-Data.request
02 01 L-SAPs: DA = 0x02, SA = 0x01
61 29 // AARE APDU
     A1 09 06 07 60 85 74 05 08 01 02 // application-context-name
     A2 03 02 01 00 // result
     A3 05 A1 03 02 01 00 // result-source-diagnostic
     BE 10 04 0E 08 00 06 5F 1F 04 00 1C 1A 20 00 EF FA 00 00
      // user-information xDLMS-InititateResponse
```

Read date and time current (1 short name)

```
17:35:16:082    ===>  Read.Request[1](7304)  (MAC:C00/003  Ic:3  Dc:0
LLC:1/2)
Hex: 02 10 50 ( 6C C0 00 03 12 90 01 02 05 01 02 1C 88 ) 00 00

-- Explanation:
6C // Credit fields
C0 00 03 // MAC addresses
12 // Pad length
90 // DL-Data.request
01 02 // L-SAPs
05 01 // ReadRequest
```

```
   02 1C 88 // Variable-Name 1C88

17:35:16:832  <== Read.Response[1]     (MAC:003/C00  Ic:3  Dc:0  LLC:2/1)
ObjACMM: 0x1C88 (7304) | {2009/06/22 FF 17:35:15:FF 8000 FF}
Hex: 02 1C 50 ( 6C 00 3C 00 06 90 02 01 0C 01 00 09 0C 07 D9 06 16 FF
11 23 0F FF 80 00 FF ) 00 00

-- Explanation:

6C // Credit fields
00 3C 00 // MAC addresses
06 // Pad length
90 // DL-Data.request
02 01 // L-SAPs
0C 01 // ReadResponse
   00 // Success
   09 0C 07 D9 06 16 FF 11 23 0F FF 80 00 FF // value of the attribute
```

<u>Read date and time current (13 short name, to provoke block transfer):</u>

```
17:36:38:406    ===>  Read.Request[13](7304)  (MAC:C00/003  Ic:0  Dc:0
LLC:1/2)  | ObjACMM: 0x1C88 (7304) | 0x1C88 (7304) | 0x1C88 (7304) |
0x1C88 (7304) | 0x1C88 (7304) | 0x1C88 (7304) | 0x1C88 (7304) | 0x1C88
(7304) | 0x1C88 (7304) | 0x1C88 (7304) | 0x1C88 (7304) | 0x1C88 (7304)
| 0x1C88 (7304) |
 Hex: 02 34 50 ( 00 C0 00 03 12 90 01 02 05 0D 02 1C 88 02 1C 88 02 1C
88 02 1C 88 02 1C 88 02 1C 88 02 1C 88 02 1C 88 02 1C 88 02 1C 88 02
1C 88 02 1C 88 02 1C 88 ) 00 00

--Explanation:

00 // Credit fields
C0 00 03 // MAC addresses
12 // Pad length
90 // DL-Data.request
01 02 // L-SAPs
05 0D // Read 13 Variable-Access-Specification
     02 1C 88 // variable-name 1C 88
     02 1C 88 02 1C 88 02 1C 88 02 1C 88 02 1C 88 02 1C 88
     02 1C 88 02 1C 88 02 1C 88 02 1C 88 02 1C 88 02 1C 88

17:36:39:609  <==  Read.Response  DataBlock_DLMS(MAC:003/C00  Ic:0  Dc:0
LLC:2/1) LastBlock:0 Block:1
Hex: 02 91 50 ( 00 00 3C 00 21 90 02 01 0C 01 02 00 00 01 81 7E 0D 00
09 0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF 00 09 0C 07 D9 06 16 FF 11
24 25 FF 80 00 FF 00 09 0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF 00 09
0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF 00 09 0C 07 D9 06 16 FF 11 24
25 FF 80 00 FF 00 09 0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF 00 09 0C
07 D9 06 16 FF 11 24 25 FF 80 00 FF 00 09 0C 07 D9 06 16 FF 11 24 25
FF 80 00 FF 00 09 0C 07 D9 ) 00 00

--Explanation:

00 // Credit fields
00 3C 00 // MAC addresses
21 // Pad length
90 // DL-Data.request
02 01 // L-SAPs
0C 01 02 // ReadResponse data-block-result
     00 // last-block = FALSE
     00 01 // block-number 00 01
     81 7E //raw-data octet-string of 126 bytes
     0D // 13 results
```

```
     00 09 0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF
     // first result success, attribute value
     00 09 0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF
     // second result success, attribute value
     00 09 0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF
     00 09 0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF
     00 09 0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF
     00 09 0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF
     00 09 0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF
     00 09 0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF
     00 09 0C 07 D9 // end of the first block

17:36:40:047   ===>  ReadNextBlock.Request[1]  (MAC:C00/003  Ic:0  Dc:0
LLC:1/2)  Block:1
Hex: 02 10 50 ( 00 C0 00 03 12 90 01 02 05 01 05 00 01 ) 00 00

00 // Credit fields
C0 00 03 // MAC addresses
12 // pad length
90 // DL-Data.request
01 02 // L-SAPs
05 01 05 // ReadRequest, variable-access-specification, block-number-
         // access
     00 01 // block-number 00 01

17:36:40:797  <==  Read.Response  DataBlock_DLMS(MAC:003/C00  Ic:0  Dc:0
LLC:2/1) LastBlock:1 Block:2
 Hex: 02 58 50 ( 00 00 3C 00 12 90 02 01 0C 01 02 01 00 02 46 06 16 FF
11 24 25 FF 80 00 FF 00 09 0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF 00
09 0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF 00 09 0C 07 D9 06 16 FF 11
24 25 FF 80 00 FF 00 09 0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF ) 00 00

-- Explanation:

00 // Credit fields
00 3C 00 // MAC addresses
12 // Pad length
90 // DL-Data.request
02 01 // L-SAPs
0C 01 02 // ReadResponse data-block-result
     01 // last-block = TRUE
     00 02 // block-number = 00 02
     46 // octet-string of 70 bytes
              06 16 FF 11 24 25 FF 80 00 FF
00 09 0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF
00 09 0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF
00 09 0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF
00 09 0C 07 D9 06 16 FF 11 24 25 FF 80 00 FF
```

Ping service: MAC frame carrying a PingRequest CI-PDU

```
17:38:43:633  ===>  Ping.Request(MAC:C00/003  Ic:0  Dc:0  LLC:0/1  |SN: 04
08 90 00 00 01)
Hex: 02 12 50 ( 00 C0 00 03 10 90 00 01 19 04 08 90 00 00 01 ) 00 00

-- Explanation:

00 // Credit fields
C0 00 03 // MAC addresses
10 // Pad length
90 //DL-Data.request
00 01 // L-SAPs
19 // PingRequest CI-PDU
```

```
     04 08 90 00 00 01 // System-Title
```

Response: MAC frame carrying a PingResponse CI-PDU

```
17:38:44:383 <=== Ping.Response(MAC:003/C00 Ic:0 Dc:0 LLC:1/0 |SN: 04
08 90 00 00 01)
Hex: 02 12 50 ( 00 00 3C 00 10 90 01 00 1A 04 08 90 00 00 01 ) 00 00

-- Explanation:

00 // Credit fields
00 3C 00 // MAC addresses
10 // Pad length
90 // DL-Data.request
01 00 // L-SAPs
1A // PingResponse CI-PDU
     04 08 90 00 00 01 // System-Title
```

RepeaterCall service

```
17:38:54:727  ===>    RepeaterCall(MAC:C00/FFF  Ic:7  Dc:0  LLC:0/1)
Max_Adr_MAC: 0x63 Nb_Tslot_For_NEW: 0
 Hex: 02 10 50 ( FC C0 0F FF 12 90 00 01 1F 00 63 00 00 ) 00 00

-- Explanation:

FC // Credit fields
C0 0F FF // MAC addresses: SA = C00, DA = FFF
12 // Pad length
90 // DL-Data.request
00 01 // L-SAPs: DA = 00, SA = 01
1F // RepeaterCall CI-PDU
     00 63 // MaxAdrMac 0x63
     00 // Nb_Tslot_For_New = 0
     00 // Reception-Threshold default value
```

A.2 CI-PDUs, ACSE APDUs and xDLMS APDUs carried by MAC frames using the HDLC based LLC sublayer

In these examples, the following communication sequence is shown, when the DLMS/COSEM S-FSK PLC profile is used with the HDLC based LLC sublayer:

- the initiator Discovers, then Registers a new server system;
- it connects the HDLC based LLC sublayer and establishes an AA;
- it reads the time attribute of the Clock object;
- it releases the AA by disconnecting the HDLC based LLC sublayer.

In these examples: SYSTEM-TITLE-SIZE = 8.

```
-- The following trace is a spy frame of a chip implementing
IEC 61334-5-1.

2009-05-14 16:04:53.922686 IEC61334-5-1-SPY [SPY-SUBFRAME] LEN=55
 S0/N0=7928/164 S1/N1=4654/374 THR=27 MET=4 SYN=0 RGAIN=2 P_SDU_LEN=38

-- Spy frame carrying a Phy frame. The Spy frame is not part of this
companion specification.

0000   02 35 B0 F8 1E A4 00 2E 12 76 01 1B 00 04 02 00
0010   00 6C 6C 00 C0 1F FF 05 7E A0 13 CE FF CD 13 61
0020   D5 E6 E6 00 1D 64 00 14 00 00 2C 66 7E 00 00 00
```

```
0030   00 00 32 9B EA 6E 10

-- Explanation:

02 // STX
35 // length
B0 // Spy subframe
F8 1E A4 00 2E 12 76 01 1B 00 04 02  // Spy data
-- here follows the 38 bytes Phy frame, carrying the MAC frame
00 00 6C 6C 00 C0 1F FF 05 7E A0 13 CE FF CD 13
61 D5 E6 E6 00 1D 64 00 14 00 00 2C 66 7E 00 00
00 00 00 32 9B EA
-- end of Phy frame
6E 10 // spy frame check field
```

<u>Discover service: MAC frame carrying a Discover CI-PDU</u>

```
-- For the MAC frame format, see IEC 61334-5-1:2001, 4.2.2

0000   6C 6C 00 C0 1F FF 05 7E A0 13 CE FF CD 13 61 D5
0010   E6 E6 00 1D 64 00 14 00 00 2C 66 7E 00 00 00 00
0020   00 32 9B EA

-- Explanation:

6C 6C // NS field, number of MAC subframes is 1
00 // Credit fields, IC = 0, CC = 0, DC = 0
C0 1F FF // MAC addresses: SA = C01, Initiator, DA = FFF
// All-Physical
05 // Pad length
7E // HDLC frame flag
A0 13 // Frame type and length
CE FF CD // MAC addresses: DA = 0x677F, upper HDLC address 0x67, lower
HDLC address = All-station, SA = 0x66
13 // UI frame
61 D5 // HDLC HCS
E6 E6 00 // DLMS/COSEM LLC addresses
1D // Discover CI-PDU
     64 // response-probability = 100
     00 14 // allowed-time-slots = 20
     00 // DiscoverReport-initial-credit = 0
     00 // ICEqualCredit = 0
2C 66 // HDLC FCS
7E // HDLC frame flag
00 00 00 00 00 // padding
32 9B EA // MAC FCS

-- From here on, only the MAC frames are shown and explained
```

<u>Response: MAC frame carrying a DiscoverReport CI-PDU</u>

```
0000   6C 6C 00 FF EC 01 00 7E A0 18 CD CE 23 13 BB 18
0010   E6 E7 00 1E 01 49 53 4B 05 00 00 00 01 00 B3 01
0020   7E 38 CD 0F

-- Explanation:

6C 6C // NS field, number of MAC subframes is 1
00 // Credit fields, IC = 0, CC = 0, DC = 0
FF EC 01 // MAC addresses: SA = FFE (NEW), DA = C01, Initiator
00 // Pad length
7E // HDLC frame flag
A0 18 // Frame type and length
```

```
CD CE 23 // DA = 0x66, SA = 0x6711, 0x11 is the lower HDLC address of
            the system sending the DiscoverReport
13 // UI frame
BB 18 // HDLC HCS
E6 E7 00 // DLMS/COSEM LLC addresses
-- DiscoverReport CI-PDU
     1E // DiscoverReport CI-PDU tag [30]
     01 // Sequence of 1
          49 53 4B 05 00 00 00 01 // system-title-server
     00 // Presence flag of the alarm-descriptor, not present
B3 01 // HDLC FCS
7E // HDLC frame flag
38 CD 0F // MAC FCS
```

Register service: MAC frame carrying a Register CI-PDU

```
0000   3A 3A 00 C0 1F FF 1B 7E A0 21 CE FF CD 13 38 17
0010   E6 E6 00 1C FE FE FE FE FE FE FE FE 01 49 53 4B
0020   05 00 00 00 01 00 10 0C E6 7E 00 00 00 00 00 00
0030   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040   00 00 00 00 00 54 F2 23

-- Explanation:

3A 3A // NS field, number of MAC subframes is 2
00 // Credit fields, IC = 0, CC = 0, DC = 0
C0 1F FF // MAC addresses: SA = C01, Initiator, DA = FFF, All-Physical
1B // Pad length, 27 bytes
7E // HDLC frame flag
A0 21 // Frame type and length
CE FF CD // DA = 0x677F, upper HDLC address All-station, SA = 66
13 // UI frame
38 17 // HDLC HCS
E6 E6 00 // DLMS/COSEM LLC addresses
1C // Register CI-PDU tag
     FE FE FE FE FE FE FE FE // active-initiator-system-title
     01 // sequence of 1
          49 53 4B 05 00 00 00 01 // system-title-server
     00 10 // MAC-address
0C E6 // HDLC FCS
7E // HDLC frame flag
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 // padding
54 F2 23 // MAC FCS
```

Establishment of a data link layer connection: MAC frame carrying an SNRM HDLC frame

```
0000   6C 6C 00 C0 10 10 10 7E A0 08 02 23 C9 93 E4 43
0010   7E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0020   00 3F 96 F1

-- Explanation:

6C 6C // NS field, number of MAC subframes is 1
00 // Credit fields, IC = 0, CC = 0, DC = 0
C0 10 10 // MAC addresses: SA = C01, Initiator, DA = 010, Individual
10 // Pad length
7E // HDLC frame flag
A0 08 // Frame type and length
02 23 C9 // DA = 0x0111, SA = 0x64
93 //SNRM frame
E4 43 // HDLC FCS
```

```
7E // // HDLC frame flag
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 // padding
3F 96 F1 // MAC FCS
```

Response: MAC frame carrying a HDLC UA frame

```
0000   3A 3A 00 01 0C 01 1D 7E A0 1F C9 02 23 73 B4 96
0010   81 80 12 05 01 7E 06 01 7E 07 04 00 00 00 01 08
0020   04 00 00 00 01 5F 75 7E 00 00 00 00 00 00 00 00
0030   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040   00 00 00 00 00 72 3D 01

-- Explanation:

3A 3A // NS field, number of MAC subframes is 2
00 // Credit fields, IC = 0, CC = 0, DC = 0
01 0C 01 // MAC addresses: SA = 010, Individual, DA = C01, Initiator
1D // pad length
7E // HDLC frame flag
A0 1F // Frame type and length
C9 02 23 // SA = 0x64, DA = 0x0111
73 // UA frame
B4 96 // HDLC HCS
81 80 12 05 01 7E 06 01 7E 07 04 00 00 00 01 08
04 00 00 00 01 // information field
5F 75 // HDLC FCS
7E // HDLC frame flag
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 // padding
72 3D 01 //MAC FCS
```

Establishment of an AA: MAC frame carrying an AARQ APDU

```
0000   56 56 00 C0 10 10 1B 7E A0 45 02 23 C9 10 21 48
0010   E6 E6 00 60 36 A1 09 06 07 60 85 74 05 08 01 01
0020   8A 02 07 80 8B 07 60 85 74 05 08 02 01 AC 0A 80
0030   08 31 32 33 34 35 36 37 38 BE 10 04 0E 01 00 00
0040   00 06 5F 1F 04 00 00 7E 1F FF FF 83 D7 7E 00 00
0050   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060   00 00 00 00 00 00 00 00 00 9B FF 67

-- Explanation:

56 56 // NS field, number of MAC subframes is 3
00 // Credit fields, IC = 0, CC = 0, DC = 0
C0 10 10 // MAC addresses SA = C01, Initiator, DA = 010, Individual
1B // Pad length
7E // HDLC frame flag
A0 45 // Frame type and length
02 23 C9 // DA = 0x0111, SA = 0x64
10 // I frame
21 48 // HDLC HCS
E6 E6 00 // LLC addresses
60 36 // AARQ APDU
     A1 09 06 07 60 85 74 05 08 01 01 // application-context-name
     8A 02 07 80 // acse-requirements
     8B 07 60 85 74 05 08 02 01 // mechanism-name
     AC 0A 80 08 31 32 33 34 35 36 37 38 // calling-authentication-
                                            value
     BE 10 04 0E 01 00 00 00 06 5F 1F 04 00 00 7E 1F FF FF
     // user-information xDLMS-Inititate.request
83 D7 // HDLC FCS
7E // HDLC frame flag
```

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00
9B FF 67
```

Response: MAC frame carrying an AARE APDU

```
0000   3A 3A 00 01 0C 01 03 7E A0 39 C9 02 23 30 22 BD
0010   E6 E7 00 61 2A A1 09 06 07 60 85 74 05 08 01 01
0020   A2 03 02 01 00 A3 05 A1 03 02 01 00 BE 11 04 0F
0030   08 01 00 06 5F 1F 04 00 00 7C 1F 04 00 00 07 19
0040   4A 7E 00 00 00 10 E9 9A
```

```
-- Explanation:

3A 3A // NS field, number of MAC subframes is 2
00 // Credit fields, IC = 0, CC = 0, DC = 0
01 0C 01 // MAC addresses: SA = 010 Individual, DA = 010, Initiator
03 // pad length
7E // HDLC frame flag
A0 39 // Frame type and length
C9 02 23 // SA = 0x64, DA = 0x0111
30 // HDLC I frame
22 BD // HDLC HCS
E6 E7 00 // LLC addresses
61 2A // AARE APDU
     A1 09 06 07 60 85 74 05 08 01 01 // application-context-name
     A2 03 02 01 00 // result
     A3 05 A1 03 02 01 00 // result-source-diagnostic
     BE 11 04 0F 08 01 00 06 5F 1F 04 00 00 7C 1F 04 00 00 07
     // user-information xDLMS-Inititate.response
19 4A // HDLC FCS
7E // HDLC frame flag
00 00 00 // pad
10 E9 9A // MAC FCS
```

Get-request-normal APDU

```
0000   3A 3A 00 C0 10 10 22 7E A0 1A 02 23 C9 32 AF 55
0010   E6 E6 00 C0 01 40 00 08 00 00 01 00 00 FF 02 00
0020   EA DD 7E 00 00 00 00 00 00 00 00 00 00 00 00 00
0030   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040   00 00 00 00 00 C2 2B 4A
```

```
-- Explanation:

3A 3A // NS field, number of MAC subframes is 2
00 // Credit fields, IC = 0, CC = 0, DC = 0
C0 10 10 // MAC addresses: SA = C01, Initiator, DA = 010, Individual
22 // pad length
7E // HDLC frame flag
A0 1A // Frame type and length
02 23 C9 // DA = 0x0111, SA = 0x64
32 // HDLC I frame
AF 55 // HDLC HCS
E6 E6 00 // LLC addresses
C0 01 40 00 08 00 00 01 00 00 FF 02 00 // Get-request-normal APDU
EA DD // HDLC FCS
7E // HDLC frame flag
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 // PAD
C2 2B 4A // MAC FCS
```

Get-response-normal APDU

```
0000   3A 3A 00 01 0C 01 1D 7E A0 1F C9 02 23 52 3F A6
0010   E6 E7 00 C4 01 40 00 09 0C 07 D2 01 07 01 01 23
0020   1A 00 FF C4 00 80 EC 7E 00 00 00 00 00 00 00 00
0030   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040   00 00 00 00 00 C1 62 A6

-- Explanation:

3A 3A // NS field, number of MAC subframes is 2
00 // Credit fields, IC = 0, CC = 0, DC = 0
01 0C 01 // MAC addresses: SA = 010 Individual, DA = 010, Initiator
1D / Pad length
7E // HDLC frame flag
A0 1F // Frame type and length
C9 02 23 // SA = 0x64, DA = 0x0111
52 // HDLC I frame
3F A6 // HDLC HCS
E6 E7 00 // LLC addresses
-- Get-response-normal APDU
C4 01 40 00 09 0C 07 D2 01 07 01 01 23 1A 00 FF C4 00
80 EC // HDLC FCS
7E // HDLC frame flag
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00
C1 62 A6 // MAC FCS
```

Releasing the AA: MAC frame carrying a HDLC DISC frame

```
0000   6C 6C 00 C0 10 10 10 7E A0 08 02 23 C9 53 E8 85
0010   7E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0020   00 B9 A4 CD

6C 6C // NS field, number of MAC subframes is 1
00 // Credit fields, IC = 0, CC = 0, DC = 0
C0 10 10 // MAC addresses: SA = C01, Initiator, DA = 010, Individual
10 // pad length
7E // HDLC frame flag
A0 08 // Frame type and length
02 23 C9 // DA = 0x0111, SA = 0x64
53 // HDLC DISC frame
E8 85 // HDLC FCS
7E // HDLC frame flag
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 // pad
B9 A4 CD // MAC FCS
```

Response: MAC frame carrying a HDLC UA frame

```
0000   3A 3A 00 01 0C 01 1D 7E A0 1F C9 02 23 73 B4 96
0010   81 80 12 05 01 7E 06 01 7E 07 04 00 00 00 01 08
0020   04 00 00 00 01 5F 75 7E 00 00 00 00 00 00 00 00
0030   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040   00 00 00 00 00 72 3D 01
3A 3A // NS field, number of MAC subframes is 2
00
01 0C 01 // MAC addresses: SA = 010 Individual, DA = 010, Initiator
1D // Pad length
7E // HDLC frame flag
A0 1F // Frame type and length
C9 02 23 // SA = 0x64, DA = 0x0111
73 // HDLC UA frame
B4 96 // HDLC HCS
```

```
// Information field
81 80 12 05 01 7E 06 01 7E 07 04 00 00 00 01 08 04 00 00 00 01
5F 75 // HDLC FCS
7E // HDLC frame flag
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 // pad
72 3D 01 // MAC FCS
```

A.3 Clear Alarm examples

In these examples, SYSTEM-TITLE-SIZE = 6.

Example 1: Clearing a single alarm in all servers

```
39 // tag for ClearAlarm, [57]
00 // Choice 0, Alarm-descriptor
00 // Alarm-Descriptor, fixed length unsigned integer
```

Example 2: Clearing a list of alarms in all servers

```
39 // tag for ClearAlarm, [57]
01 // CHOICE 1, alarm-descriptor-list, SEQUENCE OF Alarm-Descriptor
01 // Number of elements in the SEQUENCE OF
00 // Contents field: Alarm-Descriptor, fixed length unsigned integer
```

Example 3: Clearing a list of alarms in some servers

```
39 / tag for ClearAlarm, [57]
02 // CHOICE 2, SEQUENCE Alarm-Descriptor-List-And-Server-List
01 // server-id-list, number of elements in the SEQUENCE OF System-Title
040967000001 // System-title, fixed length octet-string
01 // alarm-descriptor-list, number of elements in the SEQUENCE OF Alarm-Descriptor
00 // Alarm-Descriptor, fixed length unsigned integer
```

Example 4: Clearing a different alarm in each different server

```
39 // tag for ClearAlarm, [57]
03 // CHOICE 3, alarm-descriptor-by-server-list
01 // SEQUENCE OF Alarm-Descriptor-By-Server
040967000001 // First element of the SEQUENCE: System-Title
   00 / Second element of the SEQUENCE: Alarm-Descriptor
```

Bibliography

DLMS UA 1000-1:2010, *COSEM Identification System and Interface Classes, the “Blue Book”*

DLMS UA 1000-2:2009, *DLMS/COSEM Architecture and Protocols, the "Green Book"*

DLMS UA 1001-1:2010, *DLMS/COSEM Conformance Test and certification process, the "Yellow Book"*

IEC 61334-4-512:2001, *Distribution automation using distribution line carrier systems – Part 4-512: Data communication protocols – System management using profile 61334-5-1 – Management Information Base (MIB)*

Index

This page deliberately left blank

British Standards Institution (BSI)

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

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

About us

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

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

Information on standards

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

Buying standards

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

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

Subscriptions

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

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

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

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

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

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

Revisions

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

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

Copyright

Useful Contacts:

Customer Services
Tel: +44 845 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 845 086 9001
Email: subscriptions@bsigroup.com

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

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

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

bsi.

...making excellence a habit.™