

**NORME
INTERNATIONALE
INTERNATIONAL
STANDARD**

**CEI
IEC**

62056-31

Première édition
First edition
1999-11

**Comptage de l'électricité – Echange de données
pour la lecture des compteurs, le contrôle des
tarifs et de la charge**

**Partie 31:
Utilisation des réseaux locaux sur paire torsadée
avec signal de porteuse**

**Electricity metering – Data exchange for meter
reading, tariff and load control**

**Part 31:
Use of local area networks on twisted pair with
carrier signalling**



Numéro de référence
Reference number
CEI/IEC 62056-31:1999

Numéros des publications

Depuis le 1er janvier 1997, les publications de la CEI sont numérotées à partir de 60000.

Publications consolidées

Les versions consolidées de certaines publications de la CEI incorporant les amendements sont disponibles. Par exemple, les numéros d'édition 1.0, 1.1 et 1.2 indiquent respectivement la publication de base, la publication de base incorporant l'amendement 1, et la publication de base incorporant les amendements 1 et 2.

Validité de la présente publication

Le contenu technique des publications de la CEI est constamment revu par la CEI afin qu'il reflète l'état actuel de la technique.

Des renseignements relatifs à la date de reconfirmation de la publication sont disponibles dans le Catalogue de la CEI.

Les renseignements relatifs à des questions à l'étude et des travaux en cours entrepris par le comité technique qui a établi cette publication, ainsi que la liste des publications établies, se trouvent dans les documents ci-dessous:

- «Site web» de la CEI*
- **Catalogue des publications de la CEI**
Publié annuellement et mis à jour régulièrement
(Catalogue en ligne)*
- **Bulletin de la CEI**
Disponible à la fois au «site web» de la CEI* et comme périodique imprimé

Terminologie, symboles graphiques et littéraux

En ce qui concerne la terminologie générale, le lecteur se reportera à la CEI 60050: *Vocabulaire Electrotechnique International* (VEI).

Pour les symboles graphiques, les symboles littéraux et les signes d'usage général approuvés par la CEI, le lecteur consultera la CEI 60027: *Symboles littéraux à utiliser en électrotechnique*, la CEI 60417: *Symboles graphiques utilisables sur le matériel. Index, relevé et compilation des feuilles individuelles*, et la CEI 60617: *Symboles graphiques pour schémas*.

* Voir adresse «site web» sur la page de titre.

Numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series.

Consolidated publications

Consolidated versions of some IEC publications including amendments are available. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Validity of this publication

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology.

Information relating to the date of the reconfirmation of the publication is available in the IEC catalogue.

Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is to be found at the following IEC sources:

- **IEC web site***
- **Catalogue of IEC publications**
Published yearly with regular updates
(On-line catalogue)*
- **IEC Bulletin**
Available both at the IEC web site* and as a printed periodical

Terminology, graphical and letter symbols

For general terminology, readers are referred to IEC 60050: *International Electrotechnical Vocabulary* (IEV).

For graphical symbols, and letter symbols and signs approved by the IEC for general use, readers are referred to publications IEC 60027: *Letter symbols to be used in electrical technology*, IEC 60417: *Graphical symbols for use on equipment. Index, survey and compilation of the single sheets* and IEC 60617: *Graphical symbols for diagrams*.

* See web site address on title page.

**NORME
INTERNATIONALE
INTERNATIONAL
STANDARD**

**CEI
IEC**

62056-31

Première édition
First edition
1999-11

**Comptage de l'électricité – Echange de données
pour la lecture des compteurs, le contrôle des
tarifs et de la charge**

**Partie 31:
Utilisation des réseaux locaux sur paire torsadée
avec signal de porteuse**

**Electricity metering – Data exchange for meter
reading, tariff and load control**

**Part 31:
Use of local area networks on twisted pair with
carrier signalling**

© IEC 1999 Droits de reproduction réservés — Copyright - all rights reserved

Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photo-copie et les microfilms, sans l'accord écrit de l'éditeur.

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

International Electrotechnical Commission
Telefax: +41 22 919 0300

3, rue de Varembé Geneva, Switzerland
e-mail: inmail@iec.ch IEC web site <http://www.iec.ch>



Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

CODE PRIX
PRICE CODE

XC

Pour prix, voir catalogue en vigueur
For price, see current catalogue

SOMMAIRE

	Pages
AVANT-PROPOS	6
Articles	
1 Généralités	10
1.1 Domaine d'application.....	10
1.2 Références normatives	10
2 Présentation générale	10
2.1 Vocabulaire de base	10
2.2 Couches et protocoles	12
2.3 Langage de spécification	12
2.4 Services de communication pour l'échange de données en bus local sans DLMS .	14
2.4.1 Télérelève.....	14
2.4.2 Téléprogrammation	14
2.4.3 Télétransfert point à point	18
2.4.4 Télétransfert en diffusion	20
2.4.5 Initialisation du bus	20
2.4.6 Appel des stations oubliées.....	20
2.4.7 Champs de la trame.....	22
2.4.8 Principe de la télé-alimentation en énergie.....	24
2.4.9 Présélection d'une station télé-alimentée	26
2.4.10 Communication après la présélection.....	28
2.4.11 Fonction Alarme.....	28
2.5 Services de communication pour l'échange de données par bus local avec DLMS	30
2.6 Système d'administration.....	32
3 Echange de données par bus local sans DLMS.....	32
3.1 Couche Physique.....	32
3.1.1 Protocole Physique-62056-31	32
3.1.2 Paramètres de physique	34
3.1.3 Diagramme de temps	38
3.1.4 Services et primitives de service de physique	38
3.1.5 Transitions d'état	42
3.1.6 Répertoire et traitement des erreurs.....	60
3.2 Couche Liaison.....	62
3.2.1 Protocole Liaison-62056-31	62
3.2.2 Gestion des échanges.....	62
3.2.3 Services et primitives de service de liaison	62
3.2.4 Paramètres de liaison	64
3.2.5 Transitions d'état	66
3.2.6 Répertoire et traitement des erreurs.....	80
3.3 Couche Application.....	80
3.3.1 Protocole Application-62056-31	80
3.3.2 Services et primitives de service de Application	80
3.3.3 Paramètres de Application	82
3.3.4 Transitions d'état	84
3.3.5 Répertoire et traitement des erreurs.....	90

CONTENTS

	Page
FOREWORD	7
Clause	
1 General.....	11
1.1 Scope	11
1.2 Normative references.....	11
2 General description.....	11
2.1 Basic vocabulary.....	11
2.2 Layers and protocols.....	13
2.3 Specification language.....	13
2.4 Communication services for local bus data exchange without DLMS.....	15
2.4.1 Remote reading exchange	15
2.4.2 Remote programming exchange.....	15
2.4.3 Point to point remote transfer exchange.....	19
2.4.4 Broadcast remote transfer frame.....	21
2.4.5 Bus initialization frame.....	21
2.4.6 Forgotten station call exchange	21
2.4.7 Frame fields.....	23
2.4.8 Principle of the energy remote supply	25
2.4.9 Non-energized station preselection exchange	27
2.4.10 Communication exchange after preselection	29
2.4.11 Alarm function.....	29
2.5 Communication services for local bus data exchange with DLMS.....	31
2.6 Systems management.....	33
3 Local bus data exchange without DLMS.....	33
3.1 Physical layer	33
3.1.1 Physical-62056-31 protocol.....	33
3.1.2 Physical parameters.....	35
3.1.3 Timing diagrams	39
3.1.4 Physical services and service primitives.....	39
3.1.5 State transitions.....	43
3.1.6 List and processing of errors.....	61
3.2 Data Link layer.....	63
3.2.1 Link-62056-31 protocol	63
3.2.2 Management of exchanges	63
3.2.3 Data Link services and service primitives.....	63
3.2.4 Data Link parameters.....	65
3.2.5 State transitions.....	67
3.2.6 List and processing of errors.....	81
3.3 Application layer	81
3.3.1 Application-62056-31 protocol.....	81
3.3.2 Application services and service primitives	81
3.3.3 Application parameters	83
3.3.4 State transitions.....	85
3.3.5 List and processing of errors.....	91

Articles	Pages
4 Echange de données par bus local avec DLMS.....	90
4.1 Couche Physique.....	90
4.2 Couche Liaison.....	90
4.2.1 Protocole Liaison-E/D.....	90
4.2.2 Gestion des échanges.....	92
4.2.3 Services et primitives de service de liaison.....	92
4.2.4 Paramètres de liaison.....	94
4.2.5 Transitions d'état.....	96
4.2.6 Répertoire et traitement des erreurs.....	112
4.3 Couche Application.....	112
4.3.1 Sous-couche Transport.....	112
4.3.2 Sous-couche Application.....	112
5 Echange des données par bus en local – Spécifications matérielles.....	114
5.1 Généralités.....	114
5.2 Caractéristiques générales.....	114
5.2.1 Signal de transmission à 50 kHz.....	114
5.2.2 Signal pour l'alimentation en énergie.....	118
5.2.3 Station Secondaire simple et Station Secondaire multiple.....	122
5.3 Spécification du bus.....	124
5.3.1 Caractéristiques générales.....	124
5.3.2 Caractéristiques du câble.....	126
5.3.3 Raccordements.....	128
5.4 Couplage magnétique.....	128
5.4.1 Fonction.....	128
5.4.2 Caractéristiques mécaniques communes.....	130
5.4.3 Diagramme électrique avec couplage simple.....	132
5.4.4 Diagramme électrique avec couplage alimenté.....	134
5.5 Spécifications fonctionnelles de l'émetteur de la Station Primaire.....	134
5.6 Spécifications fonctionnelles du récepteur de la Station Primaire.....	136
5.7 Spécifications fonctionnelles de l'émetteur de la Station Secondaire.....	138
5.8 Spécifications fonctionnelles du récepteur de la Station Secondaire.....	138
Annexe A (normative) Langage de spécification.....	142
Annexe B (normative) Types et caractéristiques des temps.....	148
Annexe C (normative) Liste des erreurs fatales.....	152
Annexe D (normative) Codage du champ de commande des trames.....	154
Annexe E (normative) Principe du CRC.....	158
Annexe F (normative) Génération de nombres aléatoires pour la réponse des stations oubliées.....	160
Annexe G (normative) Génération de nombres aléatoires pour l'authentification (architecture sans DLMS).....	162
Annexe H (normative) Implémentation du service d'administration des systèmes.....	164
Annexe I (informative) Précision sur les échanges.....	166

Clause	Page
4 Local bus data exchange with DLMS.....	91
4.1 Physical layer	91
4.2 Data Link layer.....	91
4.2.1 Link-E/D protocol	91
4.2.2 Management of exchanges	93
4.2.3 Data Link services and service primitives	93
4.2.4 Data Link parameters	95
4.2.5 State transitions	97
4.2.6 List and processing of errors	113
4.3 Application layer	113
4.3.1 Transport sub-layer	113
4.3.2 Application sub-layer	113
5 Local bus data exchange – Hardware	115
5.1 General.....	115
5.2 General characteristics	115
5.2.1 Signal transmission at 50 kHz	115
5.2.2 Energy supply signal transmission	119
5.2.3 Simple Secondary Station and multiple Secondary Station	123
5.3 Bus specification.....	125
5.3.1 General characteristics	125
5.3.2 Cable characteristics	127
5.3.3 Wiring	129
5.4 Magnetic plug	129
5.4.1 Function	129
5.4.2 Common mechanical characteristics	131
5.4.3 Electrical Block diagram with simple plug	133
5.4.4 Electrical Block Diagram with energy supply plug	135
5.5 Functional specifications of Primary Station transmitter (for 50 kHz signal)	135
5.6 Functional specifications of Primary Station receiver (for 50 kHz signal)	137
5.7 Functional specification of Secondary Station transmitter (for 50 kHz signal)	139
5.8 Functional specifications of Secondary Station receiver (for 50 kHz signal)	139
Annex A (normative) Specification language.....	143
Annex B (normative) Timing types and characteristics	149
Annex C (normative) List of fatal errors	153
Annex D (normative) Coding the command code field of frames	155
Annex E (normative) Principle of the CRC	159
Annex F (normative) Random integer generation for response from forgotten stations.....	161
Annex G (normative) Random number generation for authentication (architecture without DLMS).....	163
Annex H (normative) Systems management implementation.....	165
Annex I (informative) Information about exchanges	167

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

COMPTAGE DE L'ÉLECTRICITÉ – ÉCHANGE DE DONNÉES POUR LA LECTURE DES COMPTEURS, LE CONTRÔLE DES TARIFS ET DE LA CHARGE –

Partie 31: Utilisation des réseaux locaux sur paire torsadée avec signal de porteuse

AVANT-PROPOS

- 1) La CEI (Commission Electrotechnique Internationale) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI, entre autres activités, publie des Normes Internationales. Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de la CEI en ce qui concerne les questions techniques, préparés par des Comités d'Etudes où sont représentés tous les Comités nationaux s'intéressant à ces questions, expriment dans la plus grande mesure possible un accord international sur les sujets examinés.
- 3) Ces décisions constituent des recommandations internationales publiées sous forme de normes, de rapports techniques ou de guides et sont agréées comme telles par les Comités nationaux.
- 4) Dans le but d'encourager l'unification internationale, les Comités nationaux de la CEI s'engagent à appliquer de façon transparente, dans toute la mesure possible, les Normes Internationales de la CEI dans leurs normes nationales et régionales. Toute divergence entre la recommandation de la CEI et la règle nationale correspondante doit être indiquée en termes clairs dans cette dernière.
- 5) La CEI n'a fixé aucune procédure concernant le marquage comme indication d'approbation et sa responsabilité n'est pas engagée quand un matériel déclaré conforme à l'une de ses normes.

La Norme internationale CEI 62056-31 a été établie par le comité d'études 13 de la CEI: Equipements de mesure de l'énergie électrique et de commande des charges.

Cette première édition de la CEI 62056-31 annule et remplace la première édition de la CEI 61142 parue en 1993, et constitue une révision technique.

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
13/1194/FDIS	13/1203/RVD

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

La Commission Electrotechnique Internationale (CEI) attire l'attention sur le fait qu'il est déclaré que la conformité aux dispositions de la présente Norme internationale peut impliquer l'utilisation d'un brevet concernant le protocole sur lequel est basée la présente Norme CEI 62056-31.

La CEI ne prend pas position quant à la preuve, la validité et la portée de ces droits de propriété.

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**ELECTRICITY METERING – DATA EXCHANGE FOR METER READING,
TARIFF AND LOAD CONTROL –****Part 31: Use of local area networks on twisted pair
with carrier signalling**

FOREWORD

- 1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, the IEC publishes International Standards. Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. The IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of the IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested National Committees.
- 3) The documents produced have the form of recommendations for international use and are published in the form of standards, technical reports or guides and they are accepted by the National Committees in that sense.
- 4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.
- 5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.

International Standard IEC 62056-31 has been prepared by IEC technical committee 13: Equipment for electrical energy measurement and load control.

This first edition of IEC 62056-31 cancels and replaces the first edition of IEC 61142, published in 1993, and constitutes a technical revision.

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

FDIS	Report on voting
13/1194/FDIS	13/1203/RVD

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

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this International Standard may involve the use of a patent concerning the stack of protocols on which the present standard IEC 62056-31 is based.

The IEC takes no position concerning the evidence, validity and scope of this patent right.

Le détenteur de ces droits a donné l'assurance à la CEI qu'il consent à négocier des licences avec des demandeurs du monde entier, en des termes et à des conditions raisonnables et non discriminatoires. A ce propos, la déclaration du détenteur des droits de propriété est enregistrée à la CEI. Des informations peuvent être obtenues auprès de:

Association EURIDIS,
Bureau P107, 1 Avenue du Général de GAULLE, 92141 Clamart Cedex, FRANCE

L'attention est par ailleurs attirée sur le fait que certains des éléments de la présente Norme internationale peuvent faire l'objet de droits de propriété autres que ceux mentionnés ci-dessus. La CEI ne saurait être tenue pour responsable de ne pas avoir dûment signalé tout ou partie de ces droits de propriété.

Cette publication a été rédigée selon les Directives ISO/CEI, Partie 3.

Le comité a décidé que cette publication reste valable jusqu'en 2004.

A cette date, selon décision préalable du comité, la publication sera

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

Les annexes A, B, C, D, E, F, G et H font partie intégrante de cette norme.

L'annexe I est donnée uniquement à titre d'information.

The holder of this patent right has assured the IEC that he is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with the IEC. Information may be obtained from:

EURIDIS Association
Bureau P107, 1 Avenue du Général de GAULLE, 92141 Clamart Cedex, FRANCE

Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights other than those identified above. IEC shall not be held responsible for identifying any or all such patent rights.

This publication has been drafted in accordance with ISO/IEC Directives, Part 3.

The committee has decided that this publication remains valid until 2004.

At this date, in accordance with the committee's decision, the publication will be

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

Annexes A, B, C, D, E, F, G and H form an integral part of this standard.

Annex I is for information only.

COMPTAGE DE L'ÉLECTRICITÉ – ÉCHANGE DE DONNÉES POUR LA LECTURE DES COMPTEURS, LE CONTRÔLE DES TARIFS ET DE LA CHARGE –

Partie 31: Utilisation des réseaux locaux sur paire torsadée avec signal de porteuse

1 Généralités

1.1 Domaine d'application

La présente partie de la CEI 62056 décrit deux nouvelles architectures d'échange de données par bus en local avec des stations alimentées ou non en énergie. Pour les stations télé-alimentées, c'est le bus qui fournit l'énergie pour l'échange des données.

La première architecture complète le protocole de base (CEI 61142) par des services de téléprogrammation simplifiée alors que la seconde permet de mettre en œuvre les services DLMS sur le même support physique et avec la même couche physique.

Cette totale compatibilité garantit qu'il est possible d'utiliser des équipements de type CEI 61142 et CEI 62056-31 sur le même bus.

1.2 Références normatives

Les documents normatifs suivants contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente Norme internationale. Pour les références datées, les amendements ultérieurs ou les révisions de ces publications ne s'appliquent pas. Toutefois, les parties prenantes aux accords fondés sur la présente Norme internationale sont invitées à rechercher la possibilité d'appliquer les éditions les plus récentes des documents normatifs indiqués ci-après. Pour les références non datées, la dernière édition du document normatif en référence s'applique. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur.

CEI 62056-51:1998, *Comptage de l'électricité – Echange de données pour la lecture des compteurs, le contrôle des tarifs et de la charge – Partie 51: Protocoles de couche application*

CEI 61334-4-41:1996, *Automatisation de la distribution à l'aide de systèmes de communication à courants porteurs – Partie 4: Protocoles de communication de données – Section 41: Protocoles d'application – Spécification des messages de ligne de distribution*

EIA 485:—, *Standard for Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems*

ISO/IEC 8482:1993, *Technologies de l'information – Télécommunications et échange d'informations entre systèmes – Interconnexions multipoints par paire torsadée (Publiée actuellement en anglais seulement)*

2 Présentation générale

2.1 Vocabulaire de base

Toute communication fait intervenir deux équipements représentés par les expressions Station Primaire et Station Secondaire. La Station Primaire est le système qui décide

ELECTRICITY METERING – DATA EXCHANGE FOR METER READING, TARIFF AND LOAD CONTROL –

Part 31: Use of local area networks on twisted pair with carrier signalling

1 General

1.1 Scope

This part of IEC 62056 describes two new architectures for local bus data exchange with stations either energized or not. For non-energized stations, the bus supplies energy for data exchange.

The first architecture completes the base protocol (IEC 61142) with remote transfer services while the second one allows operation of DLMS services using the same physical medium and the same physical layer.

This complete compatibility guarantees the possibility of using IEC 61142 and IEC 62056-31 equipment on the same bus.

1.2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this International Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of IEC and ISO maintain registers of currently valid International Standards.

IEC 62056-51:1998, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 51: Application layer protocols*

IEC 61334-4-41:1996, *Distribution automation using distribution line carrier systems – Part 4: Data communication protocols – Section 41: Application protocols – Distribution line message specification*

EIA 485: —, *Standard for Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems*

ISO/IEC 8482:1993, *Information technology – Telecommunications and information exchange between systems – Twisted pair multipoint interconnections*

2 General description

2.1 Basic vocabulary

All communication calls upon two systems called Primary Station and Secondary Station. The Primary Station is the system that decides to initialize a communication with a remote system called Secondary Station; these designations remain valid throughout the duration of the communication.

Une communication est décomposée en un certain nombre de transactions. Chaque transaction se traduit par une émission de l'Émetteur vers le Récepteur. Au gré de l'enchaînement des transactions, les systèmes Station Primaire et Station Secondaire jouent tour à tour le rôle d'Émetteur et de Récepteur.

Dans le cas de l'architecture d'échange de données par bus local avec DLMS, les termes Client et Serveur ont le même sens que dans le modèle DLMS (voir la CEI 61334-4-41). Le Serveur (qui est obligatoirement une Station Secondaire) est le système qui se comporte comme un VDE (voir la CEI 61334-4-41) pour toute soumission de requête de service particulière. Le Client (qui est obligatoirement une Station Primaire) est le système qui utilise le Serveur dans un but spécifique à l'aide d'une ou de plusieurs soumissions de requête de service.

2.2 Couches et protocoles

L'architecture d'échange de données par bus local adopte un découpage en trois couches réseau: *Physique*, *Liaison* et *Application*. Le protocole de la couche *Physique* est commun aux deux architectures d'échanges de données par bus local, avec ou sans DLMS, ce qui permet à tous les types de stations d'être installés sur le même bus.

Les couches *Liaison* et *Application* font l'objet de protocoles dont les noms sont donnés par le tableau 1.

Tableau 1 – Architectures

	Couches	Protocoles
Architecture	<i>Application</i>	<i>Application-62056-31</i>
Sans DLMS	<i>Liaison</i>	<i>Liaison-62056-31</i>
Architecture	<i>Application</i>	<i>DLMS+</i> <i>Application+</i> <i>Transport+</i>
Avec DLMS	<i>Liaison</i>	<i>Liaison-E/D</i>

Les protocoles *Transport+* et *Application+* des sous-couches *Transport* et *Application* de la couche *Application* sont décrits dans la CEI 62056-51.

Le protocole *DLMS+* de la sous-couche *DLMS* de la couche *Application* est décrit dans la CEI 61334-4-41.

2.3 Langage de spécification

Dans cette norme, le protocole de chaque couche est décrit par des transitions d'état représentées sous forme de tableaux. La syntaxe utilisée pour la constitution de ces tableaux est définie par un langage de spécification présenté à l'annexe A.

En cas de divergence d'interprétation entre une partie du texte et un tableau de transitions d'état, c'est toujours le tableau qui fait référence.

A communication is broken down into a certain number of transactions. Each transaction consists of a transmission from the Transmitter to the Receiver. During the sequence of transactions, the Primary Station and Secondary Station systems take turns to act as Transmitter and Receiver.

For the local bus data exchange architecture with DLMS, the terms Client and Server have the same meaning as for the DLMS model (refer to IEC 61334-4-41). The Server (which is a Secondary Station) acts as a VDE (refer to IEC 61334-4-41) for the submission of special service requests. The Client (which is a Primary Station) is the system that uses the Server for a specific purpose by means of one or more service requests.

2.2 Layers and protocols

The local bus data exchange architecture uses a breakdown into three network layers: *Physical*, *Data Link* and *Application*. The protocol corresponding to the *Physical* layer is the same for both local bus data exchange architecture, with and without DLMS, allowing all kinds of stations to be installed on the same bus.

The protocols corresponding to the *Data Link* and *Application* layers are defined in the table 1.

Table 1 – Architectures

	Layers	Protocols
Architecture Without DLMS	<i>Application</i>	<i>Application-62056-31</i>
	<i>Data Link</i>	<i>Link-62056-31</i>
Architecture With DLMS	<i>Application</i>	<i>DLMS+</i> <i>Application+</i> <i>Transport+</i>
	<i>Data Link</i>	<i>Link-E/D</i>

The *Transport+* and *Application+* protocols of the *Transport* and *Application* sub-layers of the *Application* layer are described in IEC 62056-51.

The *DLMS+* protocol of the *DLMS* sub-layer of the *Application* layer is described in IEC 61334-4-41.

2.3 Specification language

In this standard, the protocol of each layer is described by state transitions represented in the form of tables. The syntax used in making up these tables is defined by a specification language described in annex A.

In the event of a difference in interpretation between part of the text and a state transition table, the table is always taken as the reference.

2.4 Services de communication pour l'échange de données en bus local sans DLMS

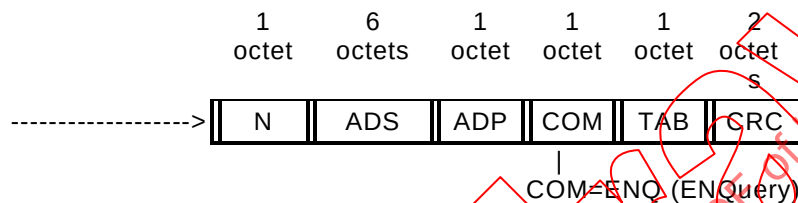
La liste des services disponibles est:

- a) télérelève de données;
- b) téléprogrammation de données;
- c) télétransfert point à point, qui est un service de téléprogrammation simplifié;
- d) télétransfert en diffusion;
- e) initialisation du bus;
- f) appel des stations oubliées.

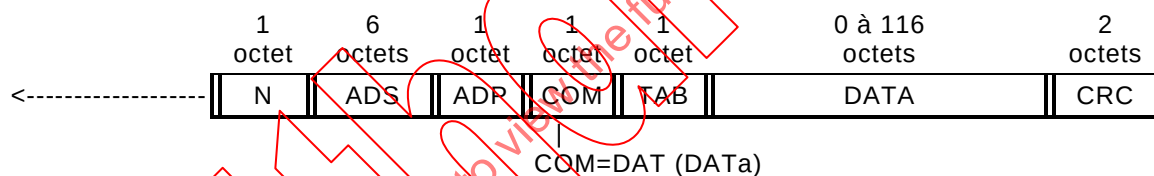
2.4.1 Télérelève

L'échange de télérelève est constitué de deux trames en une seule séquence:

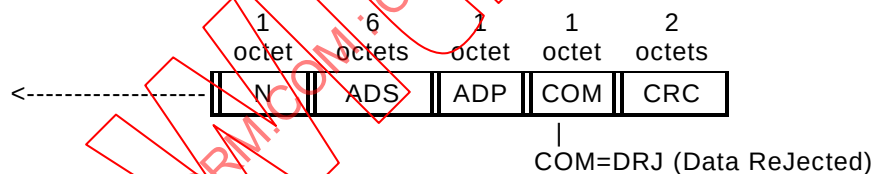
trame de télérelève contenant dans le champ TAB la référence des données à relever:



trame d'acquittement positif contenant les données relevées dans le champ DATA:



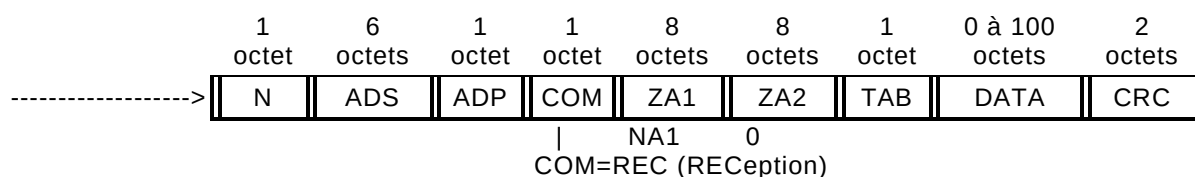
trame d'acquittement négatif (référence TAB inconnue)



2.4.2 Téléprogrammation

L'échange REC est constitué de quatre trames en deux séquences. Comme il comporte une séquence interne pour l'authentification, il apparaît du point de vue de l'application, comme un échange de deux trames en une seule séquence:

trame de téléprogrammation contenant les données à programmer dans le champ DATA et leur référence dans le champ TAB



2.4 Communication services for local bus data exchange without DLMS

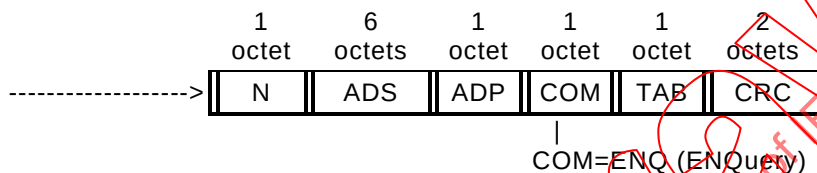
The list of available services is:

- a) remote reading of data;
- b) remote programming of data;
- c) point to point remote transfer, which is a simplified remote programming service;
- d) broadcast remote transfer;
- e) bus initialization;
- f) forgotten station call.

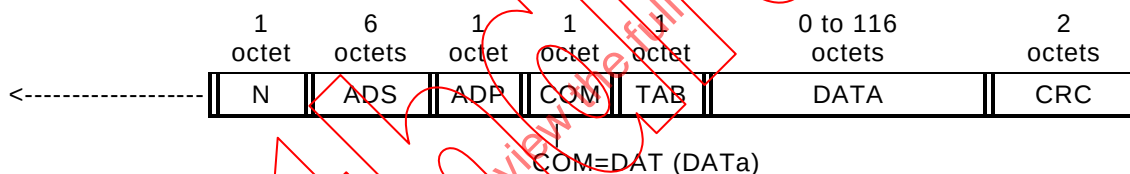
2.4.1 Remote reading exchange

The ENQ exchange consists of two frames arranged in one sequence:

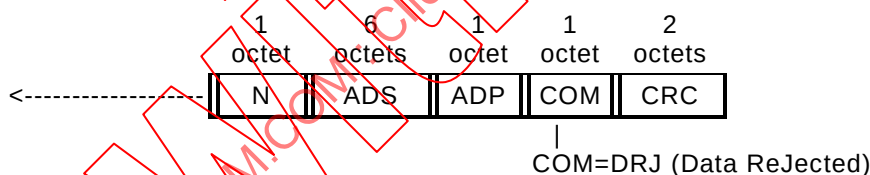
remote reading frame containing the type of data to select in the TAB field



positive acknowledgement frame with the selected data in the DATA field



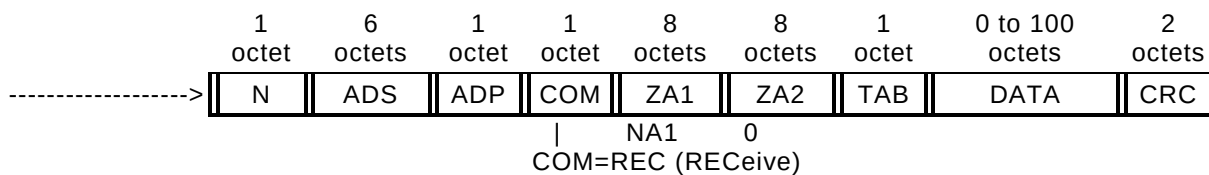
negative acknowledgement frame (TAB identifier unknown)



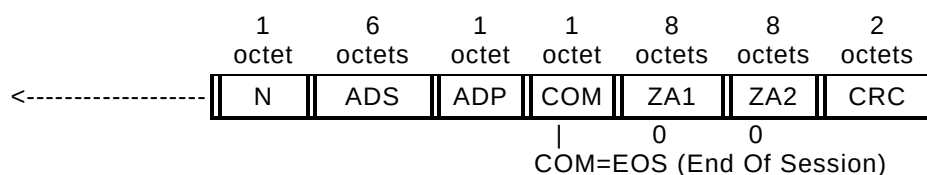
2.4.2 Remote programming exchange

The REC exchange consists of four frames arranged in two sequences. Since there is an internal sequence for authentication purpose, from the application point of view, it seems to be only one sequence with two frames:

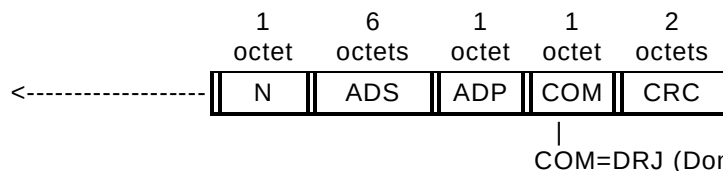
remote programming frame containing data in the DATA field and their type in the TAB field



trame d'acquittement positif (pas de problème d'authentification)



trame d'acquittement négatif (pas de problème d'authentification mais les données à programmer ne sont pas valables)



L'authentification est réalisée grâce à un échange de nombres aléatoires cryptés au moyen d'une clef secrète propre à chaque Station Secondaire. Les nombres aléatoires sont générés avec une taille de 8 octets puis cryptés grâce à l'algorithme DES, en utilisant une clef Ki sur 8 octets connue des deux stations Primaire et Secondaire.

Un nombre aléatoire NA1 est d'abord généré par la Station Primaire puis transmis dans le champ ZA1 de la trame de téléprogrammation, tandis que le champ ZA2 est mis à zéro.

A l'arrivée sur la Station Secondaire, le champ ZA1 est crypté grâce à l'algorithme DES, en utilisant la clef Ki, opération qui génère le nombre aléatoire crypté NA1K. Puis les deux stations échangent les deux trames formant la séquence interne d'authentification.

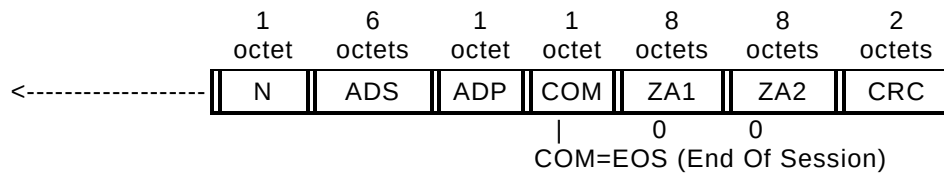
La première trame (dans le sens Station Secondaire vers Station Primaire) contient le nombre aléatoire crypté NA1K dans le champ ZA1 et un nombre aléatoire NA2 généré par la station Secondaire dans le champ ZA2.

A la réception de cette trame, la Station Primaire compare le contenu du champ ZA1 au nombre NA1' obtenu en cryptant le nombre transmis NA1 au moyen de l'algorithme DES et de sa clef Ki. Si NA1' = ZA1, la Station Primaire considère que la Station Secondaire est bien authentifiée. Sinon, elle suppose que la Station secondaire n'est pas authentifiée et coupe la session de communication.

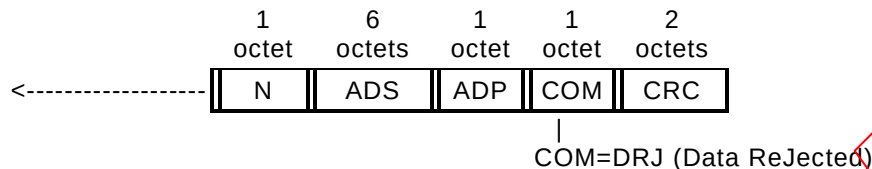
Dans le cas d'une authentification correcte de la Station Secondaire, la Station Primaire crypte le nombre aléatoire NA2 au moyen de l'algorithme DES et de sa clef Ki, obtenant le nombre aléatoire crypté NA2K qu'il transmet dans le champ ZA2, le champ ZA1 étant mis à zéro.

Sur réception de cette trame de réponse, la Station Secondaire compare le contenu du champ ZA2 au nombre NA2' obtenu en cryptant le nombre transmis NA2 au moyen de l'algorithme DES et de sa clef Ki. Si NA2' = ZA2, la Station Secondaire considère que la Station Primaire est bien authentifiée. Sinon, elle suppose que la Station Primaire n'est pas authentifiée et envoie une trame d'acquittement négatif.

positive acknowledgement frame (no authentication trouble)



negative acknowledgement frame (no authentication trouble but remote programming data not validated)



Authentication is carried out by an exchange of random numbers cyphered using a secret key specific to each Secondary Station. The random numbers are defined in 8 octets and they are cyphered with the DES algorithm using an 8-octet ciphering key Ki known both to the Primary and the Secondary station.

A random number NA1 is first generated by the Primary Station and transmitted into the ZA1 field of the remote programming frame while field ZA2 is set to zero.

On arrival at the Secondary Station, field ZA1 is cyphered by the DES algorithm with key Ki to get the cyphered random number NA1K. Then occurs the internal sequence for authentication which consists of two frames.

The first frame (from Secondary to Primary Station) contains this random number NA1K in field ZA1 and a random number NA2 generated by the Secondary station in field ZA2.

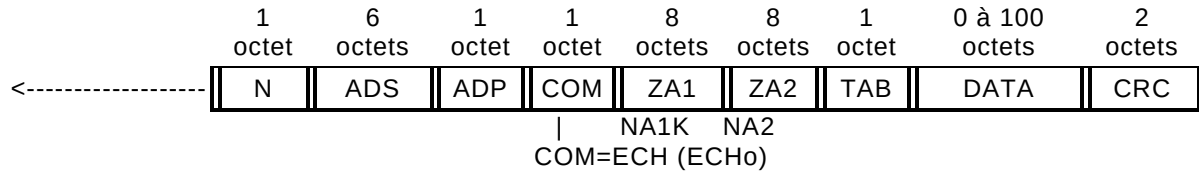
On reception of this frame, the Primary Station compares the ZA1 field to an NA1' number obtained by cyphering the transmitted NA1 number using the DES algorithm with key Ki. If $NA1' = ZA1$, then the Primary Station considers the called Secondary Station as authenticated. Otherwise, it considers the Secondary Station has not been authenticated and aborts the communication session.

After correct authentication of the Secondary Station, the Primary Station first cyphers the random number NA2 by the DES algorithm with key Ki to get the cyphered random number NA2K and then transmits it into field ZA2 while field ZA1 is set to zero.

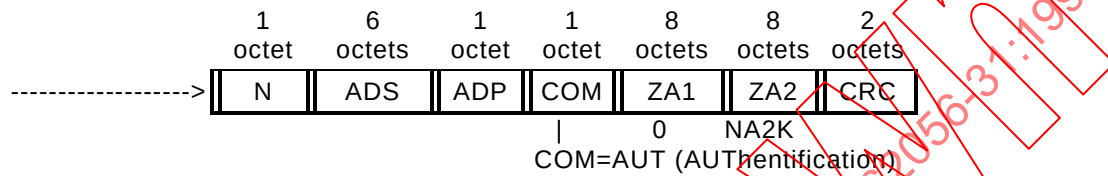
On reception of this response frame, the Secondary Station compares the ZA2 field to an NA2' number obtained by cyphering the transmitted NA2 number using the DES algorithm with key Ki. If $NA2' = ZA2$, then the Secondary Station considers the Primary Station as authenticated. Otherwise, it considers the Primary Station has not been authenticated and sends a negative acknowledgment frame.

L'échange interne d'authentification est le suivant:

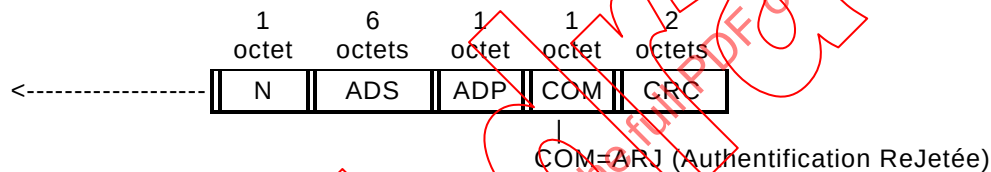
trame interne d'authentification contenant le nombre aléatoire crypté NA1K dans le champ ZA1 et le nombre aléatoire NA2 dans le champ ZA2



trame de réponse positive contenant le nombre aléatoire crypté NA2K dans le champ ZA2 (si la Station Secondaire est bien authentifiée)



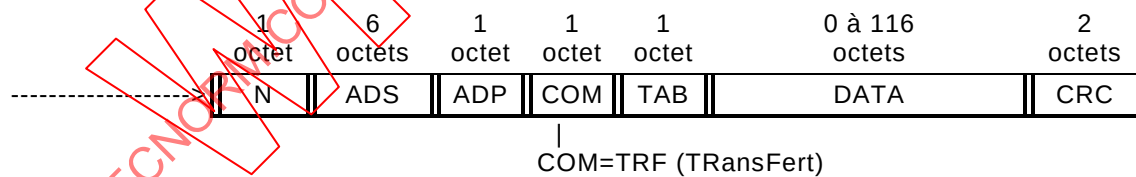
trame de rejet d'authentification, remplaçant les trames normales EOS ou DRJ quand la Station Primaire n'a pas été convenablement authentifiée



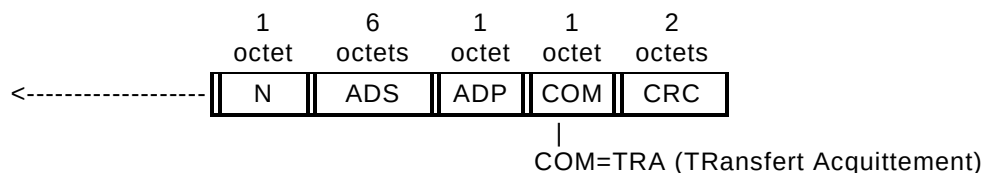
2.4.3 Télétransfert point à point

L'échange de télétransfert point à point est constitué de deux trames en une seule séquence. Du point de vue de l'application, il apparaît comme un échange de téléprogrammation en une seule séquence et sans authentification:

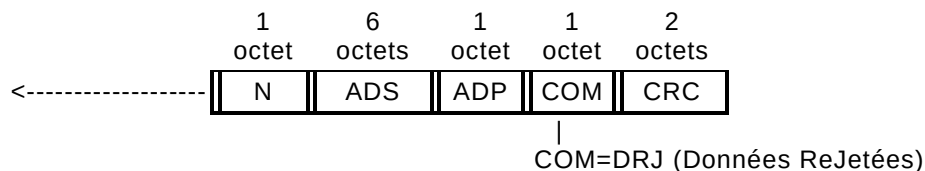
trame de télétransfert point à point contenant les données à programmer dans le champ DATA et leur référence dans le champ TAB:



trame d'acquiescement positif

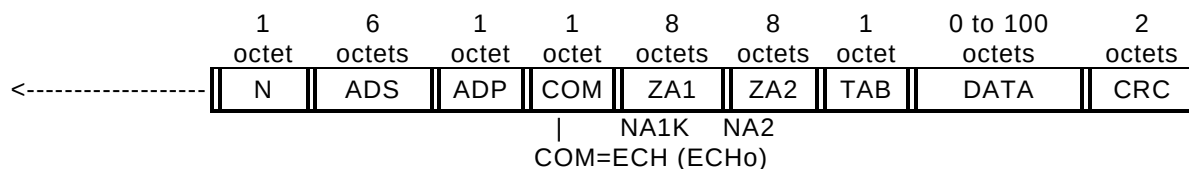


trame d'acquiescement négatif (les données transférées ne sont pas valables)

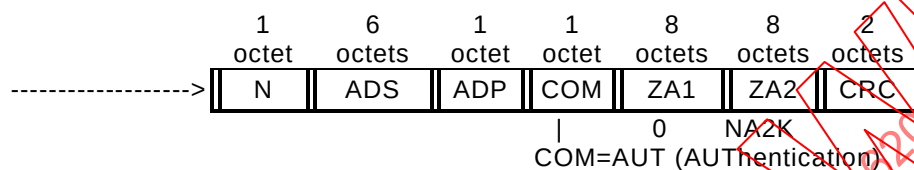


The internal authentication exchange is the following:

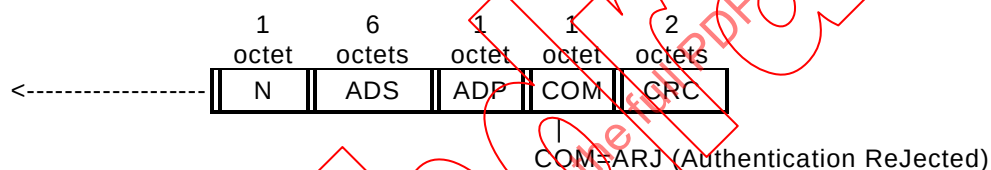
internal authentication frame containing the ciphered random number NA1K in field ZA1 and the random number NA2 in field ZA2



positive response frame containing the ciphered random number NA2K in field ZA2 (if the Secondary Station is considered as authenticated)



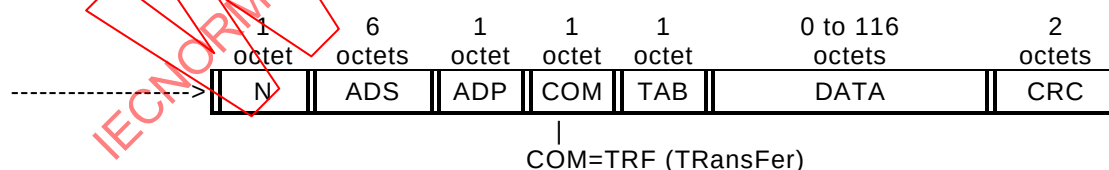
an authentication rejection frame replaces the normal EOS or DRJ frame when the Primary Station is not considered authenticated



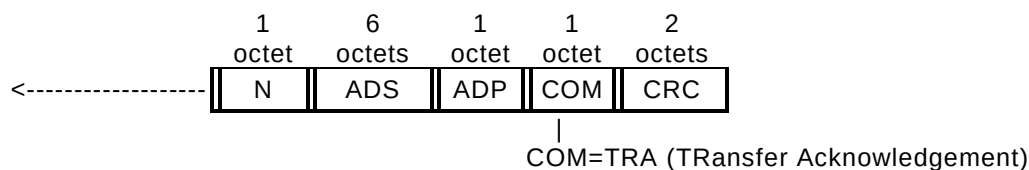
2.4.3 Point to point remote transfer exchange

This TRF exchange consists of two frames arranged in one sequence. From the application point of view, it seems to be a remote programming exchange in a single sequence with no authentication:

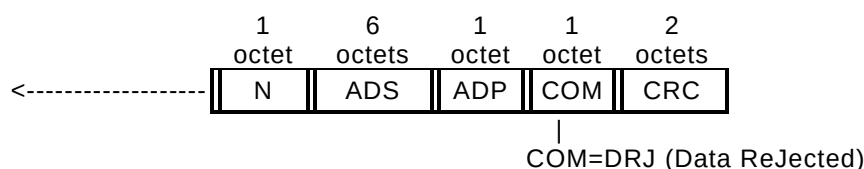
point to point remote transfer frame containing data in the DATA field and their type in the TAB field



positive acknowledgement frame



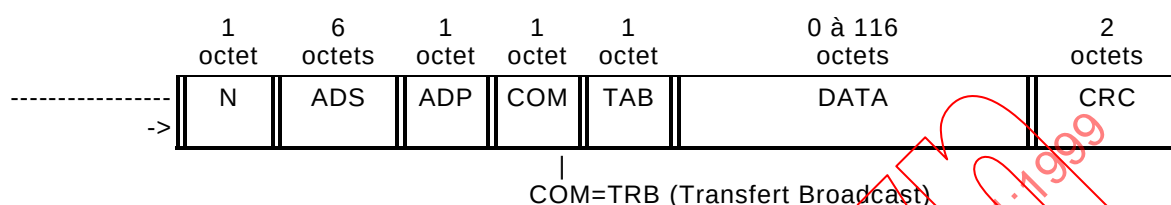
negative acknowledgement frame (remote transfer data not validated)



2.4.4 Télétransfert en diffusion

L'échange de télétransfert en diffusion ne comporte pas de trame de réponse. Du point de vue de l'application, il apparaît comme une séquence similaire à celle du télétransfert point à point, mais sans acquittement puisqu'il s'agit d'une diffusion.

trame de télétransfert en diffusion contenant les données à programmer dans le champ DATA et leur référence dans le champ TAB

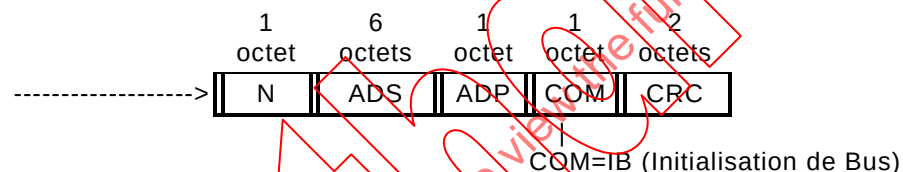


L'adresse secondaire (qui définit les Stations Secondaires destinataires) doit être une adresse de diffusion.

2.4.5 Initialisation du bus

La trame IB est sans acquittement puisqu'il s'agit d'une trame envoyée en diffusion. Elle ne véhicule pas de données, mais positionne à VRAI le drapeau de «station oubliée» pour toutes les Stations Secondaires sensibilisées à l'adresse primaire ADP véhiculée:

trame d'initialisation de bus



L'adresse secondaire (qui définit les Stations Secondaires destinataires) doit être une adresse de diffusion.

Après le passage d'une trame d'initialisation de bus, toute Station Secondaire recevant une trame ENQ correcte avec une référence TAB connue ne sera plus considérée comme une «station oubliée».

2.4.6 Appel des stations oubliées

L'échange d'appel de stations oubliées est constitué de deux trames en une seule séquence. A la fin d'une séquence de télérelève, la Station Primaire peut chercher les stations dont le drapeau de «station oubliée» est à VRAI (maximum 5 sur un total de 100).

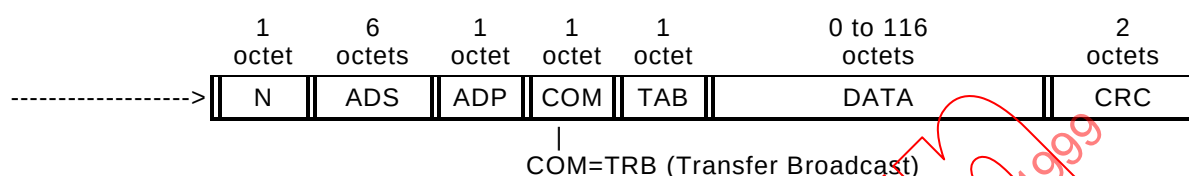
Comme un échange correct de télérelève positionne à FAUX le drapeau de «station oubliée» de la station concernée, ce service n'est demandé normalement qu'à la fin d'une séquence de télérelève, formée d'un ou de plusieurs échanges de télérelève précédés d'une trame d'initialisation de bus.

La Station Primaire gère plusieurs intervalles de temps. Quand elle détecte une collision en retour, elle doit retenter un appel des stations oubliées. Chaque fois qu'elle reçoit une réponse correcte d'une Station Secondaire, la Station Primaire doit l'éliminer de la liste des stations oubliées en effectuant une télérelève correcte de cette station.

2.4.4 Broadcast remote transfer frame

This TRB frame does not involve any frame answer. From the application point of view, it seems to be a point to point remote transfer, but without acknowledgement since it is a broadcasting:

broadcast remote transfer frame containing data in the DATA field and their type in the TAB field

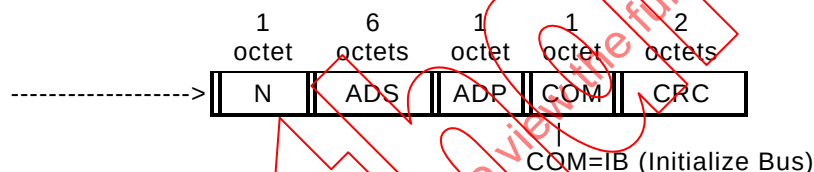


The secondary address (which defines the receiving Secondary Stations) shall be a broadcast address.

2.4.5 Bus initialization frame

This IB frame does not involve any frame answer. From the application point of view, it seems to be a broadcast remote transfer, but without data since its purpose is only to reset a special flag (called forgotten station flag) to TRUE for all Secondary Stations that have been programmed with the ADP address:

bus initialization frame



The secondary address (which defines the receiving Secondary Stations) shall be a broadcast address.

After the bus initialization frame, any Secondary Station receiving a correct ENQ frame containing a known TAB identifier will then no longer be considered as a "forgotten station".

2.4.6 Forgotten station call exchange

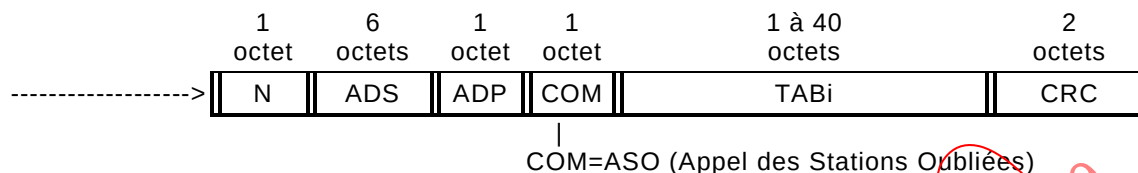
This ASO exchange consists of two frames arranged in one sequence. At the end of a remote reading sequence, the Primary Station can search for stations whose forgotten station flag is TRUE (maximum 5 in 100).

As a correct remote reading exchange sets the forgotten station flag of the corresponding station to FALSE, the ASO exchange normally occurs after the completion of a remote reading sequence that is one or several remote reading exchanges preceded by a bus initialization frame.

The Primary Station manages several time slots. When detecting a collision, it has to retry an ASO exchange. Nevertheless, each time a correct Secondary Station answer is received, the Primary Station shall eliminate it from the list of forgotten stations by operating a correct remote reading exchange with this station.

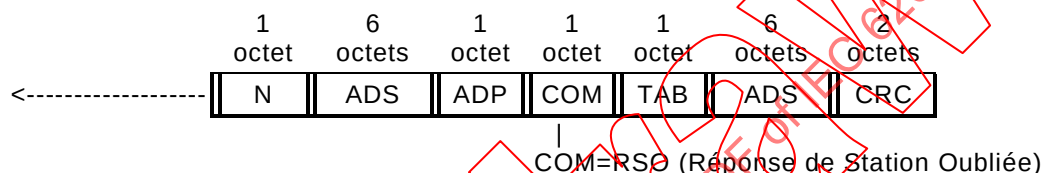
Afin de respecter les contraintes de présélection (décrites en 2.4.9), les stations télé-alimentées doivent répondre dans le premier intervalle de temps du premier échange ASO. Ensuite, seules sont sélectionnées les stations oubliées et le principe habituel peut être utilisé pour les échanges ASO suivants.

appel des stations oubliées contenant des critères de sélection dans le champ TABi (1 à 40 références TAB)



Il convient que l'adresse secondaire (qui définit les Stations Secondaires destinataires) soit une adresse de diffusion.

trame d'acquittement contenant la première référence TAB reconnue par l'unité, et l'ADS de la station



2.4.7 Champs de la trame

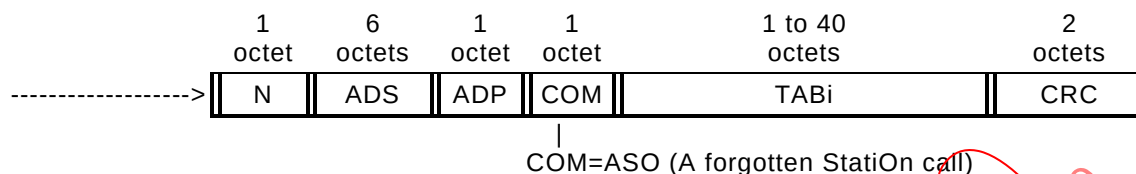
- N nombre total d'octets de la trame, y compris N.
- ADS adresse physique absolue de la Station Secondaire codée comme une bit string de taille 48. Il y a une seule adresse physique de diffusion, l'adresse générale ADG codée "000000000000" en hexadécimal ¹⁾. L'ADS correspond également au Titre-Système de la Station Secondaire.
- ADP adresse physique de la Station Primaire codée comme un bit string de taille 8. La valeur "00"H est réservée au codage de l'adresse primaire générale APG ²⁾. N'importe quelle Station Secondaire sollicitée par une Station Primaire dont l'adresse primaire est APG, répond avec la première adresse physique primaire avec laquelle elle a été programmée.
- COM code de la commande, dépendant de l'échange et de la direction de la trame (voir annexe D).
- ZA1, ZA2 champs réservés pour l'authentification réalisée au cours d'un échange de téléprogrammation.
- TAB référence des données sélectionnées, associée à certaines commandes (ENQ, DAT, REC, TRF, TRB ou RSO). La valeur "00"H est réservée aux systèmes d'administration, la valeur "FF"H à la gestion de la fonction alarme.
- DATA message de données provenant de l'application; ce champ peut être vide dans le cas de certaines commandes.
- CRC Champ de contrôle cyclique contenant les 16 bits redondants du CRC (Cyclic Redundancy Check) dont le principe est décrit à l'annexe E.

¹⁾ D'autres adresses de diffusion pourront être définies, en fonction des règles d'adressage adoptées dans les normes d'accompagnement pour la sémantique des Titres-Système, qui sont souvent basés sur un code de fabricant, une année de fabrication et un type d'équipement.

²⁾ D'autres adresses générales pourront être définies, en fonction des règles d'adressage adoptées dans les normes d'accompagnement pour la sémantique des identifications d'opérateurs, qui sont souvent basés sur un code utilitaire.

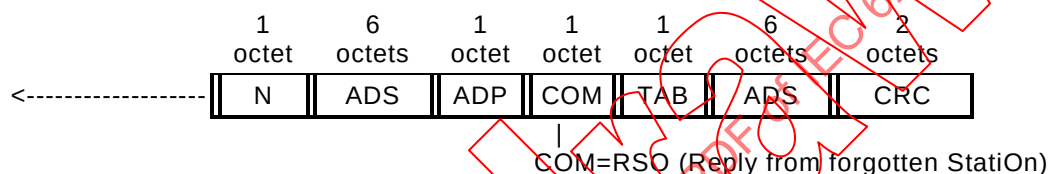
In order to assure the selection constraints (described in 2.4.9), the non-energized stations shall answer in the first time slot of the first ASO exchange. Then, only the forgotten stations are selected and the usual principle can be used for the following ASO exchanges.

forgotten station call frame containing selection criteria in the TABi field (1 to 40 TAB identifiers)



The secondary address (which defines the receiving Secondary Stations) should be a broadcast address.

acknowledgement frame containing the first TAB recognized by the unit and the ADS of the station



2.4.7 Frame fields

- N** total number of octets in the frame, including N.
- ADS** absolute physical address of the Secondary Station coded as a 48-bit string. There is only one broadcast physical address which is the general broadcast ADG coded as "000000000000" in hexadecimal ¹⁾. The ADS also corresponds exactly to the System Title of the Secondary Station.
- ADP** physical address of the Primary Station coded as an 8-bit string. The value "00"H is reserved for the coding of the physical address of the general primary APG ²⁾. Any Secondary Station solicited by a Primary Station whose physical address is APG, replies with the first primary physical address with which it has been programmed.
- COM** command code depending on the exchange and the frame direction (see annex D)
- ZA1, ZA2** fields reserved for authentication operated during the remote programming exchange
- TAB** type of data selected associated with some command codes (ENQ, DAT, REC, TRF, TRB or RSO). The value "00"H is reserved for systems management, the value "FF"H for alarm management.
- DATA** information packet from the host application. This field can be eventually empty depending on the command code.
- CRC** Cyclic Redundancy Check field corresponds to the 16 redundant bits of the CRC whose principle is described in annex E.

1) Other broadcast addresses could be defined depending on the naming rules adopted in companion standards for the semantics of the System Titles which are often based on a manufacturer code, a manufacture year and an equipment type.

2) Other general addresses could be defined depending on the naming rules adopted in companion standards for the semantics of operator identifiers which are often based on a utility code.

Les champs de la trame sont transmis dans l'ordre croissant (de N au CRC). Quand un champ contient des données sur plusieurs octets, l'émission commence par l'octet de poids faible, et se termine par l'octet de poids fort. Seul le champ DATA est considéré comme un octet string et est émis dans un ordre croissant.

2.4.8 Principe de la télé-alimentation en énergie

Le principe général qui régit les échanges de données est préservé pour les stations télé-alimentées. Il est seulement ajouté une notion d'alimentation d'énergie à distance pour les communications entre une Station Primaire et une ou plusieurs Stations Secondaires.

Pour démarrer une session, la Station Primaire doit envoyer un «Appel Général», qui alerte le système de communication de chaque Station Secondaire connectée au bus. Cet appel correspond à la présence de la porteuse durant une durée nominale qui dépend du mécanisme d'alimentation:

- la durée du signal d'«Appel Général» est AGT pour réveiller les stations télé-alimentées,
- la durée du signal d'«Appel Général» est AGN pour réveiller les stations alimentées.

Remarque: Une station secondaire peut être configurée en mode alarme. Elle est alors télé-alimentée en continu, et peut ainsi transmettre une alarme à la station primaire (voir 2.4.11).

De plus, quel que soit le type de la station distante sélectionnée (alimentée ou non), un «Appel Général» AGN intermédiaire doit être envoyé par la Station Primaire dans les circonstances suivantes:

- avant le premier échange de type ENQ ou TRF;
- avant le sixième échange consécutif et réussi de type ENQ ou TRF avec la même Station Secondaire;
- avant le premier échange de type ENQ ou TRF avec une autre Station Secondaire que celle sélectionnée auparavant, lors du précédent échange de type ENQ ou TRF;
- avant n'importe quel échange de type REC;
- avant n'importe quelle trame TRB;
- avant n'importe quelle trame IB;
- avant n'importe quel échange de type ASO.

Cela permet à la Station Primaire de ne pas réveiller toutes les stations distantes télé-alimentées si ce n'est pas nécessaire, et ainsi de sauvegarder l'énergie.

Une Station Primaire peut être munie d'un modem spécifique assurant à la fois la télé-alimentation, et les fonctions de modulation-démodulation. La durée de communication et le nombre de stations télé-alimentées doivent être optimisées pour sauvegarder les batteries de la Station Primaire.

Dans le cas où la Station Primaire ne réalise que les fonctions de modulation-démodulation, une station auxiliaire fournit en continu l'énergie sur le bus.

Une Station Secondaire, qu'elle soit alimentée ou non, ne contient généralement qu'une seule application logique, référencée par son ADS.

Il convient qu'une Station Secondaire multiple (contenant plusieurs applications logiques, correspondant à plusieurs ADSs) soit une station télé-alimentée. Ce type de station est décrit plus complètement à l'article 5.

The frame fields are transmitted in an ascending order (from N to CRC). When a field contains data over several octets, the transmission begins with the least significant octet and ends with the most significant one. However, the DATA field is considered as an octet string and transmitted in an ascending order.

2.4.8 Principle of the energy remote supply

The general principle of the data exchanges is preserved for the non-energized stations. The notion of energy remote supply is only added for communication between a Primary Station and one or more Secondary Stations.

To begin a communication session, the Primary Station should send a "Wakeup Call" designed to alert the communications system of every Secondary Station connected to the bus. This call is a continuous carrier for a nominal time depending on the energy remote supply mechanism:

- the "Wakeup Call" signal duration is AGT to wake up non-energized stations;
- the "Wakeup Call" signal duration is AGN to wake up energized stations.

Remark: A Secondary Station can be configured in Alarm mode. It is then remote supplied continuously and so can transmit the alarm to the Primary Station (see 2.4.11).

Then, whatever type of remote station is selected (energized or not), an intermediate AGN "Wakeup Call" signal shall also be required at the Primary Station side in the following circumstances:

- before the first ENQ or TRF exchange;
- before the sixth consecutive and successful ENQ or TRF exchange with the same Secondary Station;
- before the first ENQ or TRF exchange with a different Secondary Station to the one previously selected in the preceding ENQ or TRF exchange;
- before any REC exchange;
- before any TRB frame;
- before any IB frame;
- before any ASO exchange.

For non-energized stations, it means that the Primary Station can avoid to wake up all the remote stations when not necessary, and then save its energy.

A Primary Station can use a specific modem ensuring both the energy remote supply as well as the modulation and demodulation functions. The communication time and the number of non-energized stations shall be optimized in order to save the battery of the Primary Station.

As another possibility, the Primary Station might only focus on the modulation and demodulation functions. In this case, an auxiliary station continuously supplies the bus with energy.

A Secondary Station generally contains only one logical application referenced by its ADS. Such a station may or may not be energized.

A multiple Secondary Station (containing several logical applications corresponding to several ADSs) should be a non-energized station. This feature is described more fully in clause 5.

2.4.9 Présélection d'une station télé-alimentée

Pour optimiser la consommation d'énergie, un échange de présélection permet à la Station Primaire de sélectionner une Station Secondaire télé-alimentée.

L' échange de présélection est réalisé après un «Appel Général» AGT adressé à toutes les stations télé-alimentées présentes sur le bus. Pour limiter la consommation sur le bus, il convient que la première trame envoyée par la Station Primaire soit suffisamment courte, et il convient que la Station Secondaire concernée réponde avant l'expiration du délai TOPRE. Le modem de la Station Secondaire retourne dans un état de basse consommation s'il ne détecte pas de réponse dans le temps imparti.

Durant l'échange de présélection, toutes les stations télé-alimentées consomment de l'énergie. La tension sur le bus et l'énergie stockée décroissent jusqu'à ce que les stations non sélectionnées retournent à un état de veille. Puis l'envoi continu d'énergie charge les éléments de stockage d'énergie et augmente la tension sur le bus.

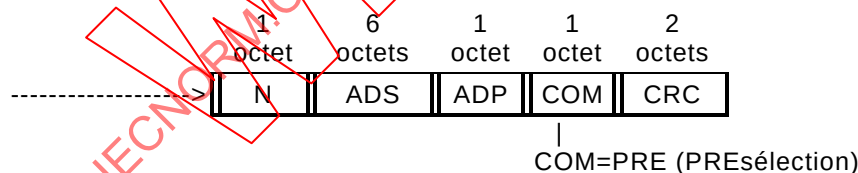
Il convient que le modem de la Station Primaire ait accumulé assez d'énergie avant la première présélection. Cette étape est garantie grâce à un temps d'attente contrôlé par le réveil TICB. A la fin de la présélection, les éléments de stockage d'énergie sont vides et la Station Primaire doit attendre la remontée de la tension sur le bus avant de réaliser une seconde présélection.

Il faut que la trame de présélection ait une taille maximale de 18 octets, elle peut correspondre à:

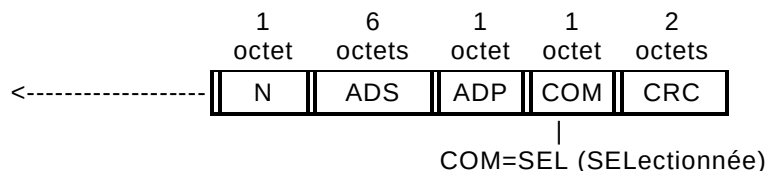
- une trame ENQ;
- une trame TRB ou TRF, si et seulement si le champ DATA a une taille inférieure ou égale à 6 octets;
- une trame IB;
- une trame ASQ, si et seulement si le nombre de champs TABi est inférieur ou égal à 7.

Comme la première trame d'un échange de type REC ou TRF peut être trop grande, un service supplémentaire est fourni pour la présélection. Cet échange complètement transparent est formé d'une séquence de deux trames.

trame de présélection d'une station télé-alimentée



trame d'acquittement



Pour préserver l'énergie de la Station Primaire, il n'y a pas de reprise lors d'un échange de présélection. Si une station télé-alimentée ne répond pas correctement, elle n'est pas sélectionnée, et la Station Primaire doit renvoyer un nouvel «Appel Général» AGT.

2.4.9 Non-energized station preselection exchange

To optimize the bus consumption, a preselection exchange enables the Primary Station to select a non-energized Secondary Station.

The preselection exchange occurs after an AGT “Wakeup Call” signal addressed to all non-energized stations of the bus. To limit the bus consumption, the first frame sent by the Primary Station should be short enough and the addressed Secondary Station should answer before the triggering of the TOPRE wakeup. Not seeing an answer in time, the modem of the Secondary Station goes back in a low consumption state.

During the preselection exchange, all the non-energized stations consume energy. The bus voltage and the energy storage capacitors decrease until the non selected stations goes back in a low consumption state. Then the continuously sent energy charges the energy storage capacitors and the bus voltage increase.

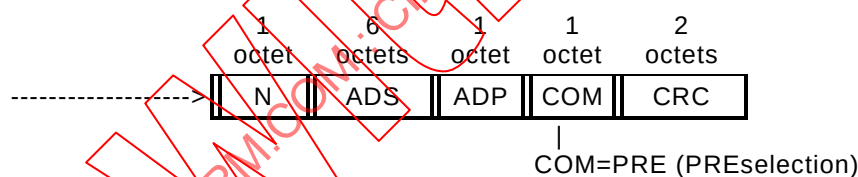
The modem of the Primary Station should store sufficient energy before the first preselection. This step is guaranteed by a wait-time controlled thanks to the TICB wakeup. At the end of a preselection, the energy storage capacitors are empty and the Primary Station shall wait for the bus voltage increase before a second preselection.

As the preselection frame shall not be more than 18 bytes long, it can be

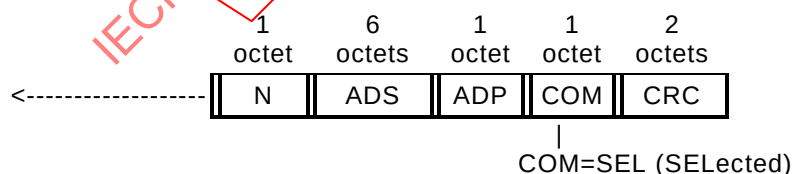
- an ENQ frame;
- a TRB or TRF frame, if and only if the data field is less than or equal to 6 octets long;
- an IB frame;
- an ASO frame, if and only if the number of TABi fields is less than or equal to 7.

As the first frame of REC and TRF exchanges may be too long, an additional service is provided for preselection. This fully transparent PRE exchange consists of two frames arranged in one sequence.

non-energized station preselection frame



acknowledgement frame



To save the energy of the Primary Station, there is no retry during a preselection exchange. If an addressed non-energized station does not answer correctly, it is not selected and the Primary Station shall send a new AGT “Wakeup Call” signal.

2.4.10 Communication après la présélection

Après la présélection, le modem de la station télé-alimentée peut rester éveillé tout le long de la communication; les délais ne sont plus critiques puisque le nombre d'équipements connectés est limité. La Station Primaire alimente la station sélectionnée et charge les accumulateurs des stations non sélectionnées.

La fin normale d'une session a lieu différemment suivant que la Station Secondaire est alimentée ou non:

- après une courte période d'inactivité au cours de la session, alors qu'un «Appel Général» AGN intermédiaire n'est pas requis, pour les stations alimentées. Cette période est contrôlée par le réveil TOL;
- après une période d'inactivité plus longue durant une session, pour une station télé-alimentée. Cette période est contrôlée par le réveil TOAG.

Noter que, pour une station télé-alimentée, un «Appel Général» intermédiaire AGN est suffisant pour poursuivre la session courante tant qu'il n'y a pas eu déclenchement du réveil TOAG.

2.4.11 Fonction Alarme

Un équipement intégré dans une station secondaire simple ou multiple (voir 5.2.3) peut transmettre des alarmes à une station primaire, à condition d'intégrer les fonctions d'interface définies ci dessous.

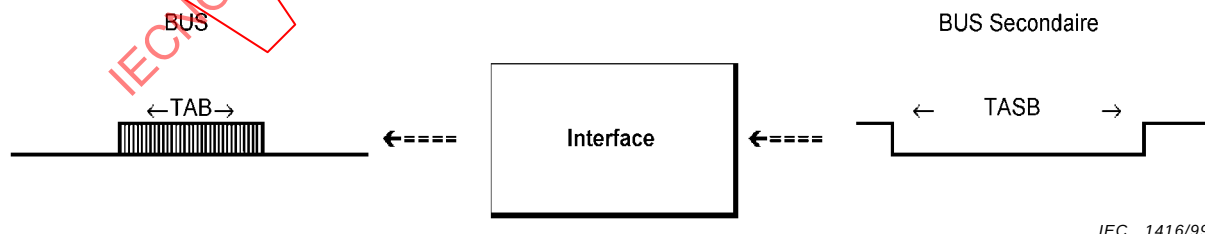
Une alarme doit être récupérée sur la station secondaire dans les 10 s au maximum.

Une configuration programmable, sur l'Interface et sur chaque équipement, permet de choisir le mode Alarme: Actif ou Inactif.

Quand le mode Alarme est Actif, un équipement à l'intérieur de la station secondaire peut générer une alarme. La fonction alarme n'est effective que si l'alimentation en énergie est présente et permanente sur le bus.

L'équipement envoie le signal d'alarme pendant un délai TASB. TASB est assez important pour forcer à un état «0» le bus secondaire et être détecté par l'Interface, même s'il y a une communication en cours.

Le mécanisme d'alarme est décrit à la figure 1.



IEC 1416/99

Figure 1 – Mécanisme d'alarme

L'alarme n'est pas directement transmise vers la station primaire. Elle est reçue par l'Interface qui la retransmet en envoyant des «0» (porteuse à 50 kHz) durant TAB sur le bus, dès que c'est possible:

- a) Aucune communication sur le bus.

Quand l'Interface reçoit l'alarme en provenance du bus secondaire, elle la retransmet sur le bus.

2.4.10 Communication exchange after preselection

After preselection, the modem of a non-energized station can stay awake for the continuing communication and delays are not critical since the number of connected devices is limited. The Primary Station supplies the selected station and charges the capacitive reservoirs of the non-selected stations.

The normal end of the communication session occurs differently depending on the energy remote supply mechanism:

- after a short period of inactivity during the communication session when no intermediate AGN “Wakeup Call” signal is required for energized stations. This period is checked by the wakeup TOL;
- after a longer period of inactivity during the communication session for non-energized stations. This period is checked by the wakeup TOAG.

Note that for a non-energized station, as far as there is no timeout of TOAG wakeup, an intermediate AGN “Wakeup Call” signal is enough to go on the current communication session.

2.4.11 Alarm function

A device integrated in a simple or multiple Secondary Station (see 5.2.3) can transmit alarms to the primary station, providing it can integrate functions of interface as described hereafter.

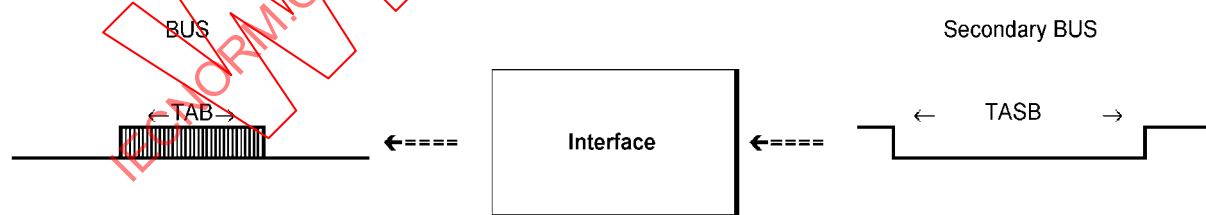
An alarm shall be fetched from the Secondary Station in 10 s maximum.

A programmable configuration on the interface and one on each device selects the status of Alarm mode: Active or Inactive.

When Alarm mode is active, the device can generate an alarm, inside the secondary station. The Alarm function is effective only if the supply is present and permanent on the bus.

The device sends the alarm during TASB. TASB is long enough to force an “0” state on the secondary bus and to be detected by the Interface, even if a communication is in progress.

Alarm mechanism is described in the figure 1.



IEC 1416/99

Figure 1 – Alarm mechanism

The alarm is not directly transmitted towards the primary station. The interface receives the alarm and transmits it by sending a “0” (50 kHz carrier) during TAB on the bus when it is possible:

- No communication on the bus

When the interface receives the alarm on the secondary bus, it transmits it on the bus.

b) En se synchronisant sur un AGN ou un AGT si une communication est en cours.

Quand une communication est en cours sur le bus, l'Interface mémorise l'alarme reçue.

Elle la retransmet sur le bus après l'un des événements suivants:

- TOALR après la fin de la réception d'un AGN ou d'un AGT;
- à la fin de la communication en cours.

Ainsi, l'Interface peut filtrer les alarmes pour éviter les conflits sur le bus.

Après la génération d'une alarme, une Station Secondaire sera considérée comme une «station oubliée» avec un critère de sélection égal à FF.

Une Station Primaire configurée en mode Alarme écoute le bus en dehors des communications et après l'émission d'un AGN ou d'un AGT pour détecter la présence d'une alarme. Après la réception d'une alarme, une Station Primaire effectue une procédure d'appel des stations oubliées, avec un critère de sélection du champ TABi égal à FF (voir 2.4.6).

La gestion des alarmes est illustrée en 3.1.3 par les diagrammes de temps.

2.5 Services de communication pour l'échange de données par bus local avec DLMS

DLMS n'offre pas de services permettant l'initialisation du bus et l'appel des stations oubliées. Néanmoins, la trame IB et l'échange ASO sont supportés et gérés comme pour l'échange de données par bus local sans DLMS. La seule différence est liée au fait que le drapeau de station oubliée est considéré comme une variable globale partagée avec l'interface de programmation de l'application.

La télérelève et le télétransfert en point à point sont réalisés directement par DLMS. Mais la téléprogrammation de données (redondante) n'est pas supportée puisque l'authentification est réservée à la couche *Application*.

Comme la sémantique des données est gérée par DLMS, le format des trames est très simple, et ne nécessite que des trames banalisées. Pour assurer la compatibilité avec l'architecture sans DLMS, le format des trames est défini à l'aide des neuf champs suivants:

1 octet	6 octets	1 octet	3 bits	1 bit	2 bits	2 bits	0 à 117 octets	2 octets
Size	ADS	ADP	DATA+	Priority	Send	Confirm	Text	CRC

Size nombre total d'octets de la trame, incluant Size. Si ce nombre diffère de 11, le récepteur sait que la trame contient des données dans le champ Text.

ADS même signification que pour l'architecture en bus local sans DLMS.

ADP même signification que pour l'architecture en bus local sans DLMS.

DATA+ toujours codé "111"B.

Priority niveau de priorité d'émission de la trame courante. La couche *Application* met la priorité en fonction du service demandé.

Send numéro de la dernière trame envoyée.

Confirm numéro de la dernière trame reçue sans erreur.

b) On synchronization of AGN or AGT when a communication is in progress. When a communication is in progress on the bus, the Interface memorises the alarm received. It transmits it to the bus after one of the following events:

- TOALR after the end of AGN or AGT reception,
- when the normal end of the communication session occurs.

In this way, the interface can filter the alarm to avoid conflict on the bus.

After the alarm generation, the Secondary Station will be considered as a “forgotten station” with a selection criteria equal to FF.

The primary station configured in Alarm mode listens to the bus when there is no communication on the bus and after transmission of an AGN or AGT in order to detect an alarm. When the primary station receives an alarm, it enters in Forgotten Station Call procedure with a selection criteria in the TABi field equal to FF (see 2.4.6).

Timing diagrams explain Alarm management in 3.1.3.

2.5 Communication services for local bus data exchange with DLMS

DLMS does not offer services to operate the bus initialization and forgotten station call mechanisms. Nevertheless, the IB frame and the ASO exchange are supported and managed as they are with the local bus data exchange architecture without DLMS except that the forgotten station flag is considered as a global variable shared with the Application Programming Interface.

Remote reading of data and point to point remote transfer are directly foreseen by DLMS. But the (redundant) remote programming of data is not supported since authentication is reserved for the *Application* layer.

As data semantics is managed by DLMS, the frame format is very simple and only unmarked frames are required. To ensure compatibility with the architecture without DLMS, this frame format is defined by the following nine fields:

1 octet	6 octets	1 octet	3 bits	1 bit	2 bits	2 bits	0 to 117 octets	2 octets
Size	ADS	ADP	DATA+	Priority	Send	Confirm	Text	CRC

Size total number of octets in the frame, including Size. If its value is not 11, the Receiver knows that the frame contains data in the Text field

ADS same rules as for local bus architecture without DLMS

ADP same rules as for local bus architecture without DLMS

DATA+ always coded "111"B

Priority transmission priority level of the current frame. The *Application* layer sets this priority according to the requested service

Send number of the last frame sent

Confirm number of the last frame received without error

- Text** DSDU (Data link Service Data Unit) du niveau supérieur. Une trame ne contient pas nécessairement du texte. Si des données en provenance de la couche *Application* sont disponibles au moment de l'envoi de la trame, elles seront mises dans le champ *Text*, autrement celui-ci sera vide. Cela permet l'échange symétrique et bidirectionnel des données. Afin de ne pas confondre les trames de type DATA+ avec les trames de l'architecture sans DLMS, les champs DATA+, Priority, Send et Confirm correspondent à un code de commande spécial, COM, dont les valeurs sont différentes de celles déjà réservées pour le champ COM (voir l'annexe D).
- CRC** même signification que pour l'architecture en bus local sans DLMS.

Les champs de la trame sont transmis dans l'ordre croissant (de Size au CRC). Quand un champ est codé sur plusieurs octets, l'émission commence par l'octet de poids faible, et se termine par l'octet de poids fort. Seul, le champ Text est considéré comme un octet string et est émis dans un ordre croissant.

2.6 Système d'administration

L'objet du système d'administration est de permettre un enrôlement qui comporte l'identification des stations secondaires sur un bus. Il offre pour ce faire le service Discover.

L'enrôlement consiste en une séquence de demande Discover effectuée par l'initiateur actif situé dans la Station Primaire. Chaque service Discover sert à informer les nouvelles stations qu'elles auront une possibilité de répondre durant les prochains intervalles de temps.

La demande Discover contient un paramètre correspondant à une probabilité de réponse spécifique. Ce paramètre est un entier compris dans l'intervalle [0, 100]. Il donne la probabilité de réponse d'une nouvelle station en pourcentage. Quand il vaut 100, toutes les nouvelles stations sur le bus doivent répondre.

Sur réception d'un Discover Indication, chaque Station Secondaire teste la valeur de son drapeau Discovered. Si il est égal à TRUE, l'Indication est détruite; sinon la station tire un nombre aléatoire entre 1 et 100. Si ce nombre est plus petit ou égal au paramètre de probabilité de réponse, la nouvelle station répond au Discover et met le drapeau associé à TRUE.

La réception d'une trame IB remet la valeur du drapeau à FAUX.

Pour assurer une compatibilité maximale (aussi bien pour les stations DLMS que pour les autres), il est proposé d'implémenter le service d'administration conformément à ce qui est indiqué à l'annexe H.

3 Echange de données par bus local sans DLMS

3.1 Couche Physique

3.1.1 Protocole Physique-62056-31

Le protocole *Physique-62056-31* de la couche *Physique* de l'architecture d'échange de données par bus local sans DLMS adopte un comportement asymétrique. La machine d'état de la Station Primaire est donc différente de celle de la Station Secondaire.

Le protocole *Physique-62056-31* supporte les Stations Secondaires télé-alimentées ou non. Ainsi qu'il est indiqué dans la description générale, les stations distantes sont réveillées par un signal «Appel Général», de type AGN ou AGT, et une session se termine à l'expiration du réveil TOL ou TOAG.

Après un signal «Appel Général», les communications se déroulent en asynchrone et en semi-duplex à 1 200 bits/s sur le bus.

- Text** DSDU (Data link Service Data Unit) from the higher level. A frame does not necessarily contain text. If data from the *Application* layer is available when the frame is sent, then the *Text* field will contain data, otherwise it will be empty. This mechanism provides the conditions for balanced bi-directional data transmission. In order not to confuse DATA+ frame with frames from the architecture without DLMS, the DATA+, Priority, Send and Confirm fields make up a special command code COM whose values shall be different from the already reserved COM values (see annex D)
- CRC** same rules as for local bus architecture without DLMS

The frame fields are transmitted in ascending order (from Size to CRC). When a field is coded on several octets, the transmission begins with the least significant octet and ends with the most significant one. However, the Text field is considered as an octet string and transmitted in ascending order.

2.6 Systems management

The purpose of Systems management is to make an enrolment which includes an identification of Secondary Stations on a bus. For that aim, it provides the Discover service.

The enrolment consists of a sequence of Discover requests issued by the active initiator located inside the Primary Station. Each Discover service is provided to inform the remaining new stations that they will have a chance to respond in the next time slots.

A Discover request conveys a specific response-probability argument as an integer in the range [0, 100]. It expresses the probability, in per cent, that a new station responds. When it is set to 100, all the new stations on the bus shall respond.

On reception of a Discover indication, each Secondary Station tests the value of its flag Discovered. If it is set to TRUE, the indication is discarded else it draws a random number between 1 and 100. If this number is smaller than or equal to the response-probability argument, the new station will issue a Discover response and set its flag Discovered to TRUE.

The flag Discovered is always reset by an IB frame.

To ensure a maximum compatibility (for stations including DLMS or not), it is proposed to implement the systems management as indicated in annex H.

3 Local bus data exchange without DLMS

3.1 Physical layer

3.1.1 Physical-62056-31 protocol

The *Physical-62056-31* protocol of the *Physical* layer of the local bus data exchange architecture without DLMS behaves asymmetrically. The state machine of the Primary Station is therefore different from that of the Secondary Station.

The *Physical-62056-31* protocol supports the Secondary Stations whether or not they are energized. As already stated in the general description, the remote stations shall be woken up either by an AGN or an AGT "Wakeup Call" signal and a communication session ends after expiry of TOL or TOAG wakeup.

After a "Wakeup Call" signal, a communication session is then made asynchronously and by half-duplex at 1 200 bits/s on the bus.

3.1.2 Paramètres de physique

La valeur de la taille maximale MaxIndex d'une trame en réception est fixée à 128.

La valeur du nombre maximal MaxRSO d'intervalles de temps RSO pour le traitement d'une trame «Appel des Stations Oubliées» est fixée à 3.

La durée du signal «Appel Général» AGN doit être dans l'intervalle [50, 150[ms, tandis que celle du signal «Appel Général» AGT doit être dans l'intervalle [200, 300[ms.

Les caractéristiques et les types des réveils et des temporisations sont données à l'annexe B.

Les valeurs données dans le tableau 2 sont définies pour une Station Primaire.

Tableau 2 – Temps d'une station primaire

	Min. ms	Nominal ms	Max. ms	Type	Définition
TA1O	–	–	120	TSL1	Temps d'attente maximal du premier octet d'une trame en réception
TAB		100		TC	Durée d'un signal d'alarme sur le bus
TAGN	–	100	–	TPDF	Durée d'un signal " Appel Général " AGN
TAO	–	–	40	TC	Temps d'attente maximal d'un octet en réception dont l'expiration indique implicitement la fin d'une trame
TARSO	–	500	–	TC	Durée d'un intervalle de temps pour le RSO
TASB		1 200		TC	Période d'attente après le début d'un signal d'alarme
TEMPO	–	40	–	TC	Délai de temporisation en fin d'émission d'un «Appel Général» ou d'une trame
TOE	–	–	1 100	TL	Délai de garde en émission pour se protéger contre un défaut matériel
TOL	–	–	100	TSL2	Temps d'attente maximal d'une requête issue de la couche supérieure
Spécifique d'une station télé-alimentée (Alimentation)					
TAGT	–	250	–	TPDF	Durée d'un signal «Appel Général» AGT
TICB	8 000	–	–	Ta	Délai initial de chargement du bus
TOAG	–	–	3 000	TPFD	Temps d'attente maximal pour une station télé-alimentée, une fois sélectionnée, pour reconnaître un signal «Appel Général» AGN

3.1.2 Physical parameters

The value of the maximum size of a frame being received, MaxIndex, is set to 128.

The value of the maximum number of RSO time slots for the processing of a “Forgotten Stations Call”, MaxRSO, is set to 3.

The AGN duration of an AGN “Wakeup Call” signal shall be in the range [50, 150[ms, while the AGT duration of an AGT “Wakeup Call” signal shall be in the range [200, 300[ms.

Timing type and characteristics are described in annex B.

The values of the table 2 are defined for a Primary Station.

Table 2 – Primary station timing

	Min. ms	Nominal ms	Max. ms	Type	Definition
TA1O	–	–	120	TSL1	Maximum waiting period of the first octet of a frame being received
TAB		100		Tc	Duration of an alarm signal on the bus
TAGN	–	100	–	TPDF	Duration of an AGN “Wakeup Call” signal
TAO	–	–	40	Tc	Maximum waiting period of one octet of a frame being received whose expiry indicates the end of a frame
TARSO	–	500	–	Tc	Length of an RSO time slot
TASB		1 200		Tc	Waiting period after the beginning of an alarm signal
TEMPO	–	40	–	Tc	Safety delay at the end of transmission of a Wake up signal or a frame
TOE	–	–	1 100	TL	Safety delay for transmitting to protect against defective hardware
TOL	–	–	100	TSL2	Maximum waiting time for a request coming from the upper layer
Non-energized station Specific (Supply)					
TAGT	–	250	–	TPDF	Duration of an AGT “Wakeup Call” signal
TICB	8 000	–	–	Ta	Initial bus charge period
TOAG	–	–	3 000	TPFD	Maximum delay for a selected non-energized station to recognize an AGN “Wakeup Call” signal

Les valeurs données dans le tableau 3 sont définies pour une Station Secondaire.

Tableau 3 – Temps d'une station secondaire

	Min. ms	Nomi nal ms	Max. ms	Type	Définition
TA1O	30 ¹⁾	–	160	TSL1	Temps d'attente maximal du premier octet d'une trame en réception
TAB	–	100	–	TC	Durée d'un signal d'alarme sur le bus
TAGN	50	100	150	TPDF	Durée d'un signal «Appel Général» AGN
TAO	–	–	40	TC	Temps d'attente maximal d'un octet d'une trame en réception dont l'expiration indique implicitement la fin d'une trame
TARSO	–	–	500	Tc	Durée d'un intervalle de temps pour le RSO
TOALR	20	–	–	TL	Attente avant l'envoi d'un AGN après la réception d'un AGN ou d'un AGT
TOE	–	–	1 100	TL	Délai de garde en émission pour se protéger contre un défaut matériel
TOL	–	–	100	TSL2	Temps d'attente maximal d'une requête issue de la couche supérieure
Spécifique d'une station télé-alimentée (Alimentation)					
TAGT	200	250	300	TPDF	Durée d'un signal " Appel Général " AGT
TASB	–	1200	–	TL	Durée d'un signal d'alarme sur le bus secondaire
TICB	8 000	–	–	Ta	Délai initial de chargement du bus
TOAG	–	–	3000	TPDF	Temps d'attente maximal pour une station télé-alimentée, une fois sélectionnée, pour reconnaître un signal " Appel Général " AGN
TOAGN	–	–	300	Tc	Temps d'inactivité maximal pour reconnaître la fin d'une session de communication avec une station télé-alimentée
TOAPPEL	–	–	180	TPFD	Temps d'attente maximal de réception du premier octet d'une trame de présélection
TOBAVARD	–	–	260	TPDF	Délai de garde pour se protéger contre les trames de présélection défectueuses
TOPRE	–	–	130	TPFD	Temps d'attente maximal de la réponse à une présélection
TOSEUIL	–	150	–	TC	Durée d'un signal «Appel Général» qui réveille une station télé-alimentée
TVASB	40	–	–	TL	Durée minimale d'un signal d'alarme sur le bus secondaire

1) Après un «Appel Général», une durée minimale de 30 ms est nécessaire.

The values of table 3 are defined for a Secondary Station.

Table 3 – Secondary station timing

	Min. ms	Nominal ms	Max. ms	Type	Definition
TA1O	30 ¹⁾	–	160	TSL1	Maximum waiting period of the first octet of a frame being received
TAB		100	–	TC	Duration of an alarm signal on the bus
TAGN	50	100	150	TPDF	Duration of an AGN "Wakeup Call" signal
TAO	–	–	40	TC	Maximum waiting period of one octet of a frame being received whose expiry indicates the end of a frame
TARSO	–	–	500	TC	Length of an RSO time slot
TOALR	20	–	–	TL	Waiting to send an AGN after an AGN or AGT reception
TOE	–	–	1 100	TL	Safety delay for transmitting to protect against defective hardware
TOL	–	–	100	TSL2	Maximum waiting time for a request coming from the upper layer
Non-energized station Specific (Supply)					
TAGT	200	250	300	TPDF	Duration of an AGT "Wakeup Call" signal
TASB	–	1200	–	TL	Duration of a secondary-bus alarm signal
TICB	8 000	–	–	Ta	Initial bus charge period
TOAG	–	–	3000	TPFD	Maximum delay for a selected non-energized station to recognize an AGN "Wakeup Call" signal
TOAGN	–	–	300	Tc	Maximum delay of inactivity to recognize the end of a communication session with an energized station
TOAPPEL	–	–	180	TPFD	Maximum waiting period of the first octet of a preselection frame being received
TOBAVAR D	–	–	260	TPDF	Safety delay to protect against defective length of preselection frame
TOPRE	–	–	130	TPFD	Maximum waiting for a preselection answer
TOSEUIL	–	150	–	TC	Duration of a "Wakeup Call" signal which wakes up a non-energized station
TVASB	40	–	–	TL	Minimum duration of an alarm signal on the secondary bus

¹⁾ After a "Wakeup Call", a minimum duration of 30 ms is necessary.

3.1.3 Diagrammes de temps

Les figures 2, 3 et 4 illustrent différents types d'échanges pour les stations secondaires télé-alimentées.

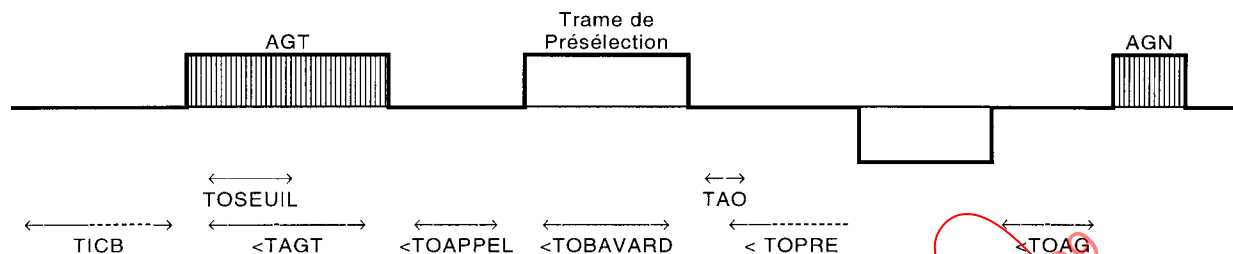
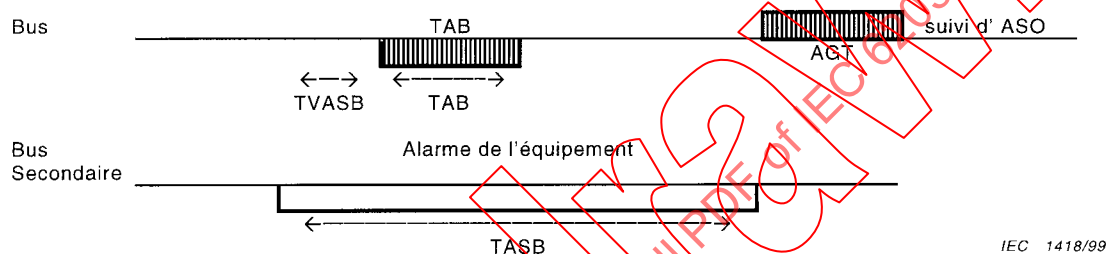


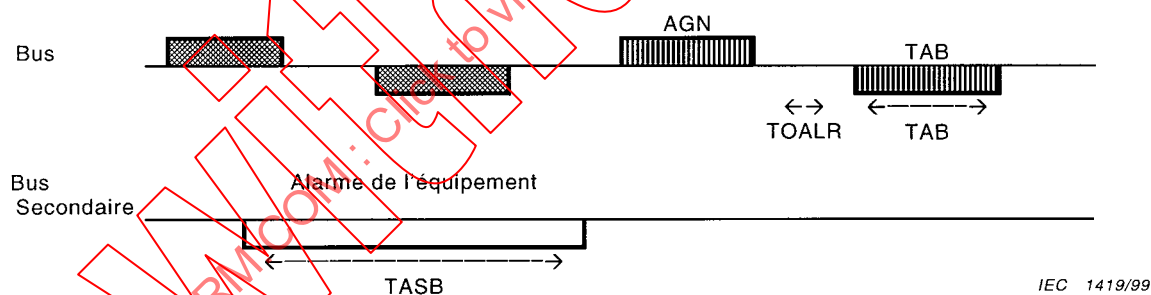
Figure 2 – Echanges sans interruption

IEC 1417/99



IEC 1418/99

Figure 3 – Alarme alors qu'il n'y a pas de communication sur le bus



IEC 1419/99

Figure 4 – Alarme alors qu'il y a une communication en cours sur le bus

3.1.4 Services et primitives de service de physique

L'utilisateur du protocole *Physique-62056-31* dispose des services et primitives de service donnés dans le tableau 4.

3.1.3 Timing diagrams

The figures 2, 3 and 4 can be used to show different types of session of the protocol for non-energized secondary stations.

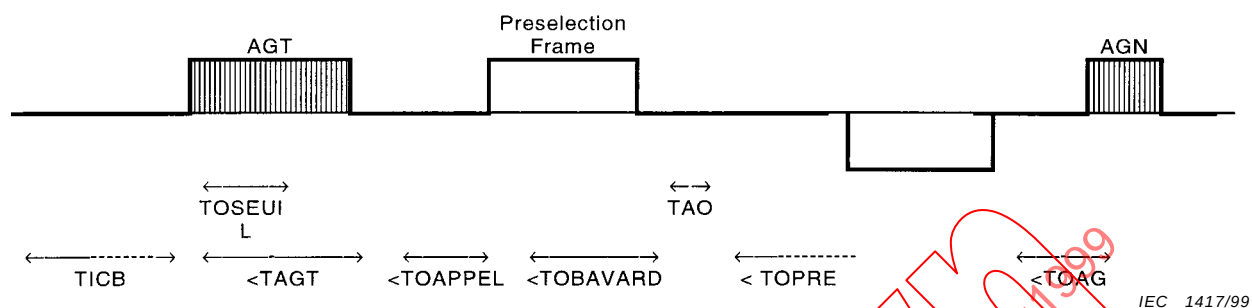


Figure 2 – Exchanges in continuous operation

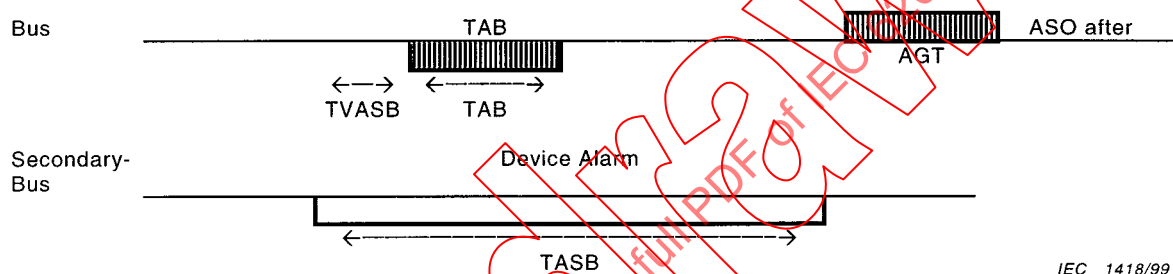


Figure 3 – Alarm event without any communication in progress

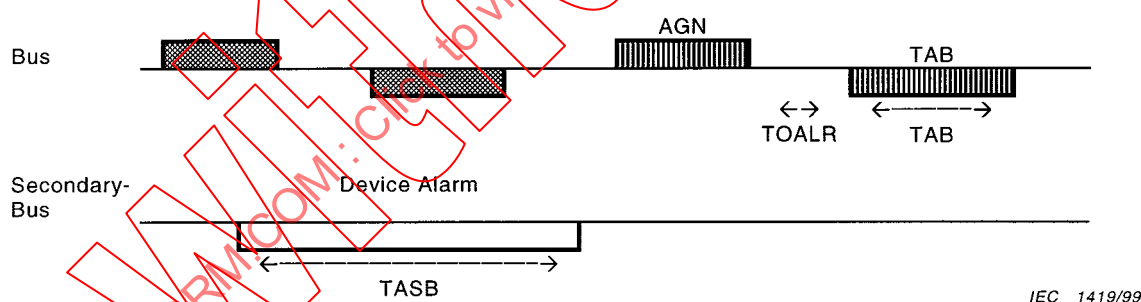


Figure 4 – Alarm event with a communication in progress

3.1.4 Physical services and service primitives

The user of the *Physical-62056-31* protocol can use the services and service primitives given in the table 4.

Tableau 4 – Services et primitives de services de physique

Service	Primitive de service
Phy_DATA	Phy_DATA.req(Frame) Phy_DATA.ind(Frame)
Phy_UNACK	Phy_UNACK.req(Frame)
Phy_APPG	Phy_APPG.req(TypeAG) Phy_APPG.ind()
Phy_ASO	Phy_ASO.req(Frame) Phy_ASO.ind(Frame)
Phy_RSO	Phy_RSO.req(Frame, Window)
Phy_COLL	Phy_COLL.ind()
Phy_ALARM	Phy_ALARM.req() Phy_ALARM.ind()
Phy_ABORT	Phy_ABORT.req() Phy_ABORT.ind(ErrorNb)

Le rôle attribué à chaque primitive est le suivant:

- Phy_DATA.req(Frame) permet à la couche *Liaison* de demander à la couche *Physique* l'émission d'une trame Frame;
- Phy_DATA.ind(Frame) permet à la couche *Physique* d'informer la couche *Liaison* qu'une trame Frame est disponible;
- Phy_UNACK.req(Frame) permet à la couche *Liaison* de demander à la couche *Physique* l'émission d'une trame Frame sans attendre d'acquittement;
- Phy_APPG.req(TypeAG) permet à la couche *Liaison* de demander à la couche *Physique* l'émission d'un signal «Appel Général». La durée TypeAG du signal est soit AGN soit AGT;
- Phy_APPG.ind() permet à la couche *Physique* d'informer la couche *Liaison* de la fin d'émission d'un signal «Appel Général»;
- Phy_ASO.req(Frame) permet à la couche *Liaison* de demander à la couche *Physique* l'émission d'une trame «Appel des Stations Oubliées»;
- Phy_ASO.ind(Frame) permet à la couche *Physique* d'informer la couche *Liaison* qu'une trame Frame a été reçue dans l'un des intervalles de temps réservés pour la réponse des stations oubliées;
- Phy_RSO.req(Frame, Window) permet à la couche *Liaison* de demander à la couche *Physique* l'émission d'une trame Frame de Réponse de Stations Oubliées dans l'intervalle de temps de numéro Window;
- Phy_COLL.ind() permet à la couche *Physique* d'informer la couche *Liaison* qu'une collision a été détectée dans l'un des intervalles de temps de réponse des stations oubliées;
- Phy_ALARM.req() permet à la couche *Liaison* de demander à la couche *Physique* l'émission d'un signal «Alarme»;
- Phy_ALARM.ind() permet à la couche *Physique* d'informer la couche *Liaison* qu'une Alarme a été détectée;
- Phy_ABORT.req() permet à la couche *Liaison* de demander à la couche *Physique* de mettre fin à son activité;
- Phy_ABORT.ind(ErrorNb) permet à la couche *Physique* d'informer la couche *Liaison* de l'occurrence d'une erreur fatale repérée par le numéro ErrorNb.

Table 4 – Physical services and service primitives

Service	Service primitive
Phy_DATA	Phy_DATA.req(Frame) Phy_DATA.ind(Frame)
Phy_UNACK	Phy_UNACK.req(Frame)
Phy_APPG	Phy_APPG.req(TypeAG) Phy_APPG.ind()
Phy_ASO	Phy_ASO.req(Frame) Phy_ASO.ind(Frame)
Phy_RSO	Phy_RSO.req(Frame, Window)
Phy_COLL	Phy_COLL.ind()
Phy_ALARM	Phy_ALARM.req() Phy_ALARM.ind()
Phy_ABORT	Phy_ABORT.req() Phy_ABORT.ind(ErrorNb)

The role assigned to each primitive is as follows:

- Phy_DATA.req(Frame) enables the *Data Link* layer to request the *Physical* layer to transmit a frame Frame;
- Phy_DATA.ind(Frame) enables the *Physical* layer to inform the *Data Link* layer that a frame Frame is available;
- Phy_UNACK.req(Frame) enables the *Data Link* layer to request the *Physical* layer to transmit a frame Frame without waiting for acknowledgement;
- Phy_APPG.req(TypeAG) enables the *Data Link* layer to request the *Physical* layer to transmit a “Wakeup Call” signal. The duration TypeAG of this signal is either AGN or AGT;
- Phy_APPG.ind() enables the *Physical* layer to inform the *Data Link* layer of the end of the transmission of a “Wakeup Call” signal;
- Phy_ASO.req(Frame) enables the *Data Link* layer to request the *Physical* layer to transmit a “Forgotten Stations Call” frame;
- Phy_ASO.ind(Frame) enables the *Physical* layer to inform the *Data Link* layer that a frame Frame has been received in one of the time slots of the forgotten stations;
- Phy_RSO.req(Frame, Window) enables the *Data Link* layer to request the *Physical* layer to transmit a Forgotten Stations Call frame Frame in the time slot number Window;
- Phy_COLL.ind() enables the *Physical* layer to inform the *Data Link* layer that a collision has been detected in one of the time slots of the forgotten stations;
- Phy_ALARM.req() enables the *Data Link* layer to request the *Physical* layer to transmit an Alarm;
- Phy_ALARM.ind() enables the *Physical* layer to inform the *Data Link* layer of the arrival of an alarm;
- Phy_ABORT.req() enables the *Data Link* layer to request the *Physical* layer to end its activity;
- Phy_ABORT.ind(ErrorNb) enables the *Physical* layer to inform the *Data Link* layer of the occurrence of a fatal error identified by the number ErrorNb.

3.1.5 Transitions d'état

Tableau 5 – Transitions d'état de *Physique-62056-31*: Station Primaire

Etat initial	Condition de déclenchement	Ensemble d'actions	Etat final
<i>Initial</i>	\$true()	MaxRSO=3 MaxIndex=128 Collision=FALSE SessionAGT=FALSE wait_time(TICB)	Stopped
Stopped	Phy_APPG.req(AG) & AG=AGN	stop_timer(TOAG) FlagAbort=FALSE TypeAG=AGN send_AG(TypeAG)	W.AG
Stopped	Phy_APPG.req(AG) & AG=AGT	SessionAGT=TRUE FlagAbort=FALSE TypeAG=AGT send_AG(TypeAG)	W.AG
Stopped	time_out(TOAG)	Phy_ABORT.ind(EP-2) SessionAGT=FALSE	Stopped
Stopped	Phy_ABORT.req()	\$none()	Stopped
Stopped	data-carrier_on	init_timer(TAB) init_timer (TASB)	W.ETABS
W.ETABS	data_carrier_off	stop_timer(TASB) stop_timer(TAB)	Stopped
W.ETABS	time_out(TAB)	Phy_ABORT.ind(EP-3) Phy_ALARM.ind()	W.TASB
W.AG	AG_sent_event	Phy_APPG.ind() init_timer(TEMPO)	W.TAB
W.AG	Phy_ABORT.req()	FlagAbort=TRUE	W.AG
W.TAB	data-carrier_on	Carrier = TRUE init_timer(TAB) init_timer (TASB)	W.TAB
W.TAB	data-carrier_off	Carrier = FALSE stop_timer(TAB) stop_timer (TASB)	W.TAB
W.TAB	time_out(TEMPO) & not(FlagAbort) & not(Carrier)	init_timer(TOL)	M.Send
W.TAB	time_out(TEMPO) & FlagAbort & not(Carrier)	wait_time (TOL)	<i>T.Session</i>
W.TAB	time_out(TEMPO) & Carrier	init_timer(TOL)	W.ETAB
W.TAB	Phy_ABORT.req()	FlagAbort=TRUE	W.TAB

3.1.5 State transitions

Table 5 – Physical-62056-31 state transitions: Primary Station

Initial state	Triggering condition	Set of actions	Final state
<i>Initial</i>	\$true()	MaxRSO=3 MaxIndex=128 Collision=FALSE SessionAGT=FALSE wait_time(TICB)	Stopped
Stopped	Phy_APPG.req(AG) & AG=AGN	stop_timer(TOAG) FlagAbort=FALSE TypeAG=AGN send_AG(TypeAG)	W.AG
Stopped	Phy_APPG.req(AG) & AG=AGT	SessionAGT=TRUE FlagAbort=FALSE TypeAG=AGT send_AG(TypeAG)	W.AG
Stopped	time_out(TOAG)	Phy_ABORT.ind(EP-2) SessionAGT=FALSE	Stopped
Stopped	Phy_ABORT.req()	\$none()	Stopped
Stopped	data-carrier_on	init_timer(TAB) init_timer(TASB)	W.ETABS
W.ETABS	data_carrier_off	stop_timer(TASB) stop_timer(TAB)	Stopped
W.ETABS	time_out(TAB)	Phy_ABORT.ind(EP-3) Phy_ALARM.ind()	W.TASB
W.AG	AG_sent_event	Phy_APPG.ind() init_timer(TEMPO)	W.TAB
W.AG	Phy_ABORT.req()	FlagAbort=TRUE	W.AG
W.TAB	data-carrier_on	Carrier = TRUE init_timer(TAB) init_timer(TASB)	W.TAB
W.TAB	data-carrier_off	Carrier = FALSE stop_timer(TAB) stop_timer(TASB)	W.TAB
W.TAB	time_out(TEMPO) & not(FlagAbort) & not(Carrier)	init_timer(TOL)	M.Send
W.TAB	time_out(TEMPO) & FlagAbort & not(Carrier)	wait_time(TOL)	<i>T.Session</i>
W.TAB	time_out(TEMPO) & Carrier	init_timer(TOL)	W.ETAB
W.TAB	Phy_ABORT.req()	FlagAbort=TRUE	W.TAB

W.ETAB	time_out(TAB)	Phy_ABORT.ind(EP-3) Phy_ALARM.ind() stop_timer(TOL)	W.TASB
W.ETAB	data_carrier_off & not(FlagAbort)	stop_timer(TAB) stop_timer (TASB)	M.Send
W.ETAB	data_carrier_off & FlagAbort	stop_timer(TAB) stop_timer (TASB)	W.TOL
W.ETAB	Phy_ABORT.req()	FlagAbort=TRUE	W.ETAB
W.TASB	time_out(TASB)	\$none()	Stopped
W.TOL	time_out(TOL)	\$none()	T.Session
M.Send	Phy_DATA.req(Frame)	Service=NORMAL	SendFirst
M.Send	Phy_UNACK.req(Frame)	Service=UNACKNOWLEDGED	SendFirst
M.Send	Phy_ASO.req(Frame)	Service=ASO	SendFirst
M.Send	Phy_ABORT.req()	\$none()	M.Send
M.Send	time_out(TOL)	\$none()	T.Session
T.Session	SessionAGT=TRUE	init_timer(TOAG) Phy_ABORT.ind(EP-1) wait_time(TEMPO)	Stopped
T.Session	SessionAGT=FALSE	Phy_ABORT.ind(EP-1) wait_time(TEMPO)	Stopped
SendFirst	\$true()	stop_timer(TOL) Size=Size(Frame) Index=1 send_octet(Frame, Index) Size=Size-1 init_timer(TOE)	Sending
Sending	octet_sent_event & Size>0	Index=Index+1 send_octet(Frame, Index) Size=Size-1	Sending
Sending	octet_sent_event & Size=0	stop_timer(TOE) wait_time(TAO) Index=1 Frame=""	Answer
Sending	Phy_ABORT.req()	stop_timer(TOE) wait_time(TAO) init_timer(TA1O) FlagAbort=TRUE	M.Rec
Sending	time_out(TOE)	Phy_ABORT.ind(EP-3F) wait_time(TAO) init_timer(TA1O) FlagAbort=TRUE	M.Rec
Answer	Service=NORMAL I Service=UNACKNOWLEDGED	init_timer(TA1O)	M.Rec
Answer	Service=ASO	WinRSO=1 init_timer(TARSO) init_timer(TA1O)	M.Rec

W.ETAB	time_out(TAB)	Phy_ABORT.ind(EP-3) Phy_ALARM.ind() stop_timer(TOL)	W.TASB
W.ETAB	data_carrier_off & not(FlagAbort)	stop_timer(TAB) stop_timer (TASB)	M.Send
W.ETAB	data_carrier_off & FlagAbort	stop_timer(TAB) stop_timer (TASB)	W.TOL
W.ETAB	Phy_ABORT.req()	FlagAbort=TRUE	W.ETAB
W.TASB	time_out(TASB)	\$none()	Stopped
W.TOL	time_out(TOL)	\$none()	T.Session
M.Send	Phy_DATA.req(Frame)	Service=NORMAL	SendFirst
M.Send	Phy_UNACK.req(Frame)	Service=UNACKNOWLEDGED	SendFirst
M.Send	Phy_ASO.req(Frame)	Service=ASO	SendFirst
M.Send	Phy_ABORT.req()	\$none()	M.Send
M.Send	time_out(TOL)	\$none()	T.Session
T.Session	SessionAGT=TRUE	init_timer(TOAG) Phy_ABORT.ind(EP-1) wait_time(TEMPO)	Stopped
T.Session	SessionAGT=FALSE	Phy_ABORT.ind(EP-1) wait_time(TEMPO)	Stopped
SendFirst	\$true()	stop_timer(TOL) Size=size(Frame) Index=1 send_octet(Frame, Index) Size=Size-1 init_timer(TOE)	Sending
Sending	octet_sent_event & Size>0	Index=Index+1 send_octet(Frame, Index) Size=Size-1	Sending
Sending	octet_sent_event & Size=0	stop_timer(TOE) wait_time(TAO) Index=1 Frame=""	Answer
Sending	Phy_ABORT.req()	stop_timer(TOE) wait_time(TAO) init_timer(TA1O) FlagAbort=TRUE	M.Rec
Sending	time_out(TOE)	Phy_ABORT.ind(EP-3F) wait_time(TAO) init_timer(TA1O) FlagAbort=TRUE	M.Rec
Answer	Service=NORMAL I Service=UNACKNOWLEDGED	init_timer(TA1O)	M.Rec
Answer	Service=ASO	WinRSO=1 init_timer(TARSO) init_timer(TA1O)	M.Rec

M.Rec	octet_received_event	stop_timer(TA1O) Index=Index+1 read_data(RecB) concat(Frame, RecB) init_timer(TAO)	Receiving
M.Rec	collision_detected_event	stop_timer(TA1O) Collision=TRUE init_timer(TAO)	Receiving
M.Rec	time_out(TA1O)	\$none()	Received
M.Rec	Phy_ABORT.req()	FlagAbort=TRUE	M.Rec
Receiving	octet_received_event & Index<=MaxIndex	stop_timer(TAO) Index=Index+1 read_data(RecB) concat(Frame, RecB) init_timer(TAO)	Receiving
Receiving	octet_received_event & Index>MaxIndex	Phy_ABORT.ind(EP-4F) wait_time(TAO) FlagAbort=TRUE	Received
Receiving	collision_detected_event	stop_timer(TAO) Collision=TRUE init_timer(TAO)	Receiving
Receiving	time_out(TAO)	\$none()	Received
Receiving	time_out(TARSO)	Phy_ABORT.ind(EP-5F) wait_time(TAO) FlagAbort=TRUE	Received
Receiving	Phy_ABORT.req()	Flagabort = TRUE	Receiving
Received	Service=NORMAL & not(Flagabort)	Phy_DATA.ind(Frame) init_timer(TOL)	M.Send
Received	(Service=NORMAL & Flagabort) Service=UNACKNOWLEDGED	wait_time(TOL)	T.Session
Received	Service=ASO & Collision & not(Flagabort)	Phy_COLL.ind() Collision=FALSE	T.RSO
Received	Service=ASO & not(Collision) & not(Flagabort)	Phy_ASO.ind(Frame)	T.RSO
Received	Service=ASO & Flagabort	\$none()	T.RSO
T.RSO	(TypeAG=AGT) (WinRSO>=MaxRSO) & (TypeAG=AGN)	stop_timer(TARSO)	T.Session
T.RSO	(WinRSO<MaxRSO) & (TypeAG=AGN)	Index=1 Frame=""	W.RSO
W.RSO	time_out(TARSO)	WinRSO=WinRSO+1 init_timer(TARSO) init_timer(TA1O)	M.Rec
W.RSO	Phy_ABORT.req()	Flagabort=TRUE	W.RSO

M.Rec	octet_received_event	stop_timer(TA1O) Index=Index+1 read_data(RecB) concat(Frame, RecB) init_timer(TAO)	Receiving
M.Rec	collision_detected_event	stop_timer(TA1O) Collision=TRUE init_timer(TAO)	Receiving
M.Rec	time_out(TA1O)	\$none()	Received
M.Rec	Phy_ABORT.req()	FlagAbort=TRUE	M.Rec
Receiving	octet_received_event & Index<=MaxIndex	stop_timer(TAO) Index=Index+1 read_data(RecB) concat(Frame, RecB) init_timer(TAO)	Receiving
Receiving	octet_received_event & Index>MaxIndex	Phy_ABORT.ind(EP-4F) wait_time(TAO) FlagAbort=TRUE	Received
Receiving	collision_detected_event	stop_timer(TAO) Collision=TRUE init_timer(TAO)	Receiving
Receiving	time_out(TAO)	\$none()	Received
Receiving	time_out(TARSO)	Phy_ABORT.ind(EP-5F) wait_time(TAO) FlagAbort=TRUE	Received
Receiving	Phy_ABORT.req()	Flagabort = TRUE	Receiving
Received	Service=NORMAL & not(Flagabort)	Phy_DATA.ind(Frame) init_timer(TOL)	M.Send
Received	(Service=NORMAL & Flagabort) Service=UNACKNOWLEDGED	wait_time(TOL)	T.Session
Received	Service=ASO & Collision & not(Flagabort)	Phy_COLL.ind() Collision=FALSE	T.RSO
Received	Service=ASO & not(Collision) & not(Flagabort)	Phy_ASO.ind(Frame)	T.RSO
Received	Service=ASO & Flagabort	\$none()	T.RSO
T.RSO	(TypeAG=AGT) (WinRSO>=MaxRSO) & (TypeAG=AGN)	stop_timer(TARSO)	T.Session
T.RSO	(WinRSO<MaxRSO) & (TypeAG=AGN)	Index=1 Frame=""	W.RSO
W.RSO	time_out(TARSO)	WinRSO=WinRSO+1 init_timer(TARSO) init_timer(TA1O)	M.Rec
W.RSO	Phy_ABORT.req()	Flagabort=TRUE	W.RSO

**Tableau 6 – Transitions d'état de la gestion d'alimentation en énergie
(Station Secondaire télé-alimentée seulement)**

Etat initial	Condition de déclenchement	Ensemble d'actions	Etat final
<i>Initial</i>	alarm_detection()	Flagalarm=TRUE FlagSendAlarm =FALSE station_power(ON)	Stopped
<i>Initial</i>	not(alarm_detection())	Flagalarm=FALSE	Stopped
Stopped	occur(cpt_carrier_on) & Flagalarm	init_timer(TVASB)	W.TVASB2
Stopped	occur(data_carrier_on)	init_timer(TOSEUIL) init_timer(TAGT)	W.TOSEUIL
W.TOSEUIL	time_out(TOSEUIL)& not(Flagalarm)	station_power(ON)	W.AGT
W.TOSEUIL	occur(data_carrier_off) & not(Flagalarm)	stop_timer(TOSEUIL) stop_timer(TAGT)	Stopped
W.TOSEUIL	time_out(TOSEUIL) & Flagalarm	station_signal(ON) Tend = TOAG	W.AGT
W.TOSEUIL	occur(data_carrier_off) & Flagalarm	stop_timer(TOSEUIL) stop_timer(TAGT) Tend = TOAGN init_timer(Tend)	Hide
W.TOSEUIL	occur(cpt_carrier_on) & Flagalarm	init_timer(TVASB)	W.TVASB1
W.AGT	occur(data_carrier_off)	stop_timer(TAGT) init_timer(TOAPPEL)	W.Sel
W.AGT	time_out(TAGT) & not(Flagalarm)	station_power(OFF)	Stopped
W.AGT	time_out(TAGT) & Flagalarm	init_timer(Tend)	Hide
W.Sel	occur(octet_received_event)	stop_timer(TOAPPEL) init_timer(TOBAVARD) init_timer(TAO)	Select
W.Sel	time_out(TOAPPEL) & not(Flagalarm)	station_power(OFF)	Stopped
W.Sel	time_out(TOAPPEL) & Flagalarm	station_signal(OFF)	Stopped
W.Sel	occur(cpt_carrier_on) & Flagalarm & not(FlagSendalarm)	init_timer(TVASB)	W.TVASB1
Select	occur(octet_received_event)	stop_timer(TAO) init_timer(TAO)	Select
Select	time_out(TAO)	stop_timer(TOBAVARD) init_timer(TOPRE)	W.Answer

**Table 6 – Power supply management state transitions
(only for non-energized Secondary Station)**

Initial state	Triggering condition	Set of actions	Final state
<i>Initial</i>	alarm_detection()	Flagalarm = TRUE FlagSendAlarm = FALSE station_power(ON)	Stopped
<i>Initial</i>	not(alarm_detection())	Flagalarm = FALSE	Stopped
Stopped	occur(cpt_carrier_on) & Flagalarm	init_timer(TVASB)	W.TVASB2
Stopped	occur(data_carrier_on)	init_timer(TOSEUIL) init_timer(TAGT)	W.TOSEUIL
W.TOSEUIL	time_out(TOSEUIL)& not(Flagalarm)	station_power(ON)	W.AGT
W.TOSEUIL	occur(data_carrier_off) & not(Flagalarm)	stop_timer(TOSEUIL) stop_timer(TAGT)	Stopped
W.TOSEUIL	time_out(TOSEUIL) & Flagalarm	station_signal(ON) Tend = TOAG	W.AGT
W.TOSEUIL	occur(data_carrier_off) & Flagalarm	stop_timer(TOSEUIL) stop_timer(TAGT) Tend = TOAGN init_timer(Tend)	Hide
W.TOSEUIL	occur(cpt_carrier_on) & Flagalarm	init_timer(TVASB)	W.TVASB1
W.AGT	occur(data_carrier_off)	stop_timer(TAGT) init_timer(TOAPPEL)	W.Sel
W.AGT	time_out(TAGT) & not(Flagalarm)	station_power(OFF)	Stopped
W.AGT	time_out(TAGT) & Flagalarm	init_timer(Tend)	Hide
W.Sel	occur(octet_received_event)	stop_timer(TOAPPEL) init_timer(TOBAVARD) init_timer(TAO)	Select
W.Sel	time_out(TOAPPEL) & not(Flagalarm)	station_power(OFF)	Stopped
W.Sel	time_out(TOAPPEL) & Flagalarm	station_signal(OFF)	Stopped
W.Sel	occur(cpt_carrier_on) & Flagalarm & not(FlagSendalarm)	init_timer(TVASB)	W.TVASB1
Select	occur(octet_received_event)	stop_timer(TAO) init_timer(TAO)	Select
Select	time_out(TAO)	stop_timer(TOBAVARD) init_timer(TOPRE)	W.Answer

Select	time_out(TOBAVARD) & not(Flagalarm)	stop_timer(TAO) station_power(OFF)	Stopped
Select	time_out(TOBAVARD) & Flagalarm	stop_timer(TAO) init_timer(Tend)	Hide
W.Answer	occur(octet_sent_event)	stop_timer(TOPRE) init_timer(Tend)	Hide
W.Answer	time_out(TOPRE) & not(Flagalarm)	station_power(OFF)	Stopped
W.Answer	time_out(TOPRE) & Flagalarm	init_timer(Tend)	Hide
W.Answer	occur(cpt_carrier_on) & Flagalarm & not(FlagSendalarm)	init_timer(TVASB)	W.TVASB1
Hide	occur(octet_received_event) occur(octet_sent_event) (occur(data_carrier_on) & not(FlagSendAlarm))	stop_timer(Tend) init_timer(Tend)	Hide
Hide	occur(data_carrier_on) & FlagSendAlarm	stop_timer(Tend)	W.AGend
Hide	time_out(Tend) & not(Flagalarm)	station_power(OFF)	Stopped
Hide	time_out(Tend) & Flagalarm & not(FlagSendAlarm)	station_signal(OFF)	Stopped
Hide	time_out(Tend) & Flagalarm & FlagSendAlarm	Send_AG(AGN)	W.AB
Hide	occur(cpt_carrier_on) & Flagalarm & not(FlagSendalarm)	init_timer(TVASB)	W.TVASB1
W.AGend	occur(data_carrier_off)	wait_time(TOALR) Send_AG(AGN)	W.AB
W.TVASB1	occur(cpt_carrier_off)	stop_timer(TVASB) init_timer(Tend)	Hide
W.TVASB1	time_out(TVASB)	FlagSendAlarm = TRUE init_timer(Tend)	Hide
W.TVASB1	time_out(Tend)	\$none()	W.TVASB2
W.TVASB2	occur(cpt_carrier_off)	stop_timer(TVASB) station_signal(OFF)	Stopped
W.TVASB2	occur(data_carrier_on)	\$none()	W.TVASB1
W.TVASB2	time_out(TVASB)	Send_AG(AGN)	W.AB
W.AB	AG_sent_event	FlagSendAlarm = FALSE station_signal(OFF)	Stopped

Select	time_out(TOBAVARD) & not(Flagalarm)	stop_timer(TAO) station_power(OFF)	Stopped
Select	time_out(TOBAVARD) & Flagalarm	stop_timer(TAO) init_timer(Tend)	Hide
W.Answer	occur(octet_sent_event)	stop_timer(TOPRE) init_timer(Tend)	Hide
W.Answer	time_out(TOPRE) & not(Flagalarm)	station_power(OFF)	Stopped
W.Answer	time_out(TOPRE) & Flagalarm	init_timer(Tend)	Hide
W.Answer	occur(cpt_carrier_on) & Flagalarm & not(FlagSendalarm)	init_timer(TVASB)	W.TVASB1
Hide	occur(octet_received_event) occur(octet_sent_event) (occur(data_carrier_on) & not(FlagSendAlarm))	stop_timer(Tend) init_timer(Tend)	Hide
Hide	occur(data_carrier_on) & FlagSendAlarm	stop_timer(Tend)	W.AGend
Hide	time_out(Tend) & not(Flagalarm)	station_power(OFF)	Stopped
Hide	time_out(Tend) & Flagalarm & not(FlagSendAlarm)	station_signal(OFF)	Stopped
Hide	time_out(Tend) & Flagalarm & FlagSendAlarm	Send_AG(AGN)	W.AB
Hide	occur(cpt_carrier_on) & Flagalarm & not(FlagSendalarm)	init_timer(TVASB)	W.TVASB1
W.AGend	occur(data_carrier_off)	wait_time(TOALR) Send_AG(AGN)	W.AB
W.TVASB1	occur(cpt_carrier_off)	stop_timer(TVASB) init_timer(Tend)	Hide
W.TVASB1	time_out(TVASB)	FlagSendAlarm = TRUE init_timer(Tend)	Hide
W.TVASB1	time_out(Tend)	\$none()	W.TVASB2
W.TVASB2	occur(cpt_carrier_off)	stop_timer(TVASB) station_signal(OFF)	Stopped
W.TVASB2	occur(data_carrier_on)	\$none()	W.TVASB1
W.TVASB2	time_out(TVASB)	Send_AG(AGN)	W.AB
W.AB	AG_sent_event	FlagSendAlarm = FALSE station_signal(OFF)	Stopped

Tableau 7 – Transitions d'état de Physique-62056-31: Station Secondaire

Etat initial	Condition de déclenchement	Ensemble d'actions	Etat final
<i>Initial</i>	energized()	MaxIndex=128 FlagRSO=FALSE FirstWinRSO=FALSE	Stopped
<i>Initial</i>	not(energized())	MaxIndex=18 FlagRSO=FALSE FirstWinRSO=TRUE	Stopped
Stopped	AG_received_event	Stop_timer(TOAG) init_timer(TA1O)	M.Rec
Stopped	Phy_ALARM.req()	TypeAG=ASB Send_AG(TypeAG)	W.ASB
Stopped	time_out(TOAG)	MaxIndex=18 FirstWinRSO=TRUE	Stopped
M.Rec	octet_received_event	Stop_timer(TA1O) Index=2 Frame="" Read_data(RecB) Concat(Frame, RecB) Init_timer(TAO)	Receiving
M.Rec	time_out(TA1O)	Phy_ABORT(EP-1)	WTOAG
M.Rec	Phy_ABORT.req()	Stop_timer(TA1O)	WTOAG
Receiving	octet_received_event & Index<=MaxIndex	Stop_timer(TAO) Index=Index+1 Read_data(RecB) Concat(Frame, RecB) Init_timer(TAO)	Receiving
Receiving	octet_received_event & Index>MaxIndex	Stop_timer(TAO) Phy_ABORT.ind(EP-4F)	WTOAG
Receiving	time_out(TAO)	Phy_DATA.ind(Frame) Init_timer(TOL)	M.Send
Receiving	Phy_ABORT.req()	Stop_timer(TAO)	WTOAG
M.Send	Phy_DATA.req(Frame)	Stop_timer(TOL) MaxIndex=128 Size=size(Frame) Index=1 Send_octet(Frame, Index) Size=Size-1 Init_timer(TOE)	Sending
M.Send	Phy_RSO.req(Frame, Window)	Stop_timer(TOL) MaxIndex=128 Wait_window(FirstWinRSO, Window) FirstWinRSO=FALSE Size=size(Frame) Index=1 Send_octet(Frame, Index) Size=Size-1 FlagRSO=TRUE Init_timer(TOE)	Sending
M.Send	time_out(TOL)	Init_timer(TA1O)	M.Rec
M.Send	Phy_ABORT.req()	Stop_timer(TOL)	WTOAG
Sending	octet_sent_event & Size>0	Index=Index+1 Send_octet(Frame, Index) Size=Size-1	Sending
Sending	octet_sent_event & Size=0 & not(FlagRSO)	Stop_timer(TOE) init_timer(TA1O)	M.Rec

Table 7 – Physical-62056-31 state transitions: Secondary Station

Initial state	Triggering condition	Set of actions	Final state
<i>Initial</i>	energized()	MaxIndex=128 FlagRSO=FALSE FirstWinRSO=FALSE	Stopped
<i>Initial</i>	not(energized())	MaxIndex=18 FlagRSO=FALSE FirstWinRSO=TRUE	Stopped
Stopped	AG_received_event	Stop_timer(TOAG) init_timer(TA1O)	M.Rec
Stopped	Phy_ALARM.req()	TypeAG=ASB Send_AG(TypeAG)	W.ASB
Stopped	time_out(TOAG)	MaxIndex=18 FirstWinRSO=TRUE	Stopped
M.Rec	octet_received_event	Stop_timer(TA1O) Index=2 Frame="" Read_data(RecB) Concat(Frame, RecB) init_timer(TAO)	Receiving
M.Rec	time_out(TA1O)	Phy_ABORT(EP-1)	WTOAG
M.Rec	Phy_ABORT.req()	Stop_timer(TA1O)	WTOAG
Receiving	octet_received_event & Index<=MaxIndex	Stop_timer(TAO) Index=Index+1 Read_data(RecB) Concat(Frame, RecB) init_timer(TAO)	Receiving
Receiving	octet_received_event & Index>MaxIndex	Stop_timer(TAO) Phy_ABORT.ind(EP-4F)	WTOAG
Receiving	time_out(TAO)	Phy_DATA.ind(Frame) init_timer(TOL)	M.Send
Receiving	Phy_ABORT.req()	Stop_timer(TAO)	WTOAG
M.Send	Phy_DATA.req(Frame)	Stop_timer(TOL) MaxIndex=128 Size=size(Frame) Index=1 Send_octet(Frame, Index) Size=Size-1 init_timer(TOE)	Sending
M.Send	Phy_RSO.req(Frame, Window)	Stop_timer(TOL) MaxIndex=128 Wait_window(FirstWinRSO, Window) FirstWinRSO=FALSE Size=size(Frame) Index=1 Send_octet(Frame, Index) Size=Size-1 FlagRSO=TRUE init_timer(TOE)	Sending
M.Send	time_out(TOL)	init_timer(TA1O)	M.Rec
M.Send	Phy_ABORT.req()	Stop_timer(TOL)	WTOAG
Sending	octet_sent_event & Size>0	Index=Index+1 Send_octet(Frame, Index) Size=Size-1	Sending
Sending	octet_sent_event & Size=0 & not(FlagRSO)	Stop_timer(TOE) init_timer(TA1O)	M.Rec

Sending	octet_sent_event & Size=0 & FlagRSO	Stop_timer(TOE) Wait_time(TAO) FlagRSO=FALSE	WTOAG
Sending	Phy_ABORT.req()	Stop_timer(TOE)	WTOAG
Sending	time_out(TOE)	Phy_ABORT.ind(EP-3F)	WTOAG
W.ASB	time_out(TOAG)	MaxIndex=18 FirstWinRSO=TRUE	W.ASB
W.ASB	AG_sent_event	\$none()	Stopped
WTOAG	Not(energized)	init_timer(TOAG)	Stopped
WTOAG	energized	\$none()	Stopped

Tableau 8 – Signification des états mentionnés dans les tableaux précédents

Etat	Signification
<i>Initial</i>	Initialisation des variables de la couche
Stopped	Attente d'un signal «Appel Général»
W.ETABS (Wait for end of " Alarm-Bus ")	Attente de la fin d'un signal «Alarme-Bus» reçu dans l'état Stopped
W.AG (Wait for end of " Wakeup Call ")	Attente de la fin de l'émission d'un signal «Appel Général»
W.TAB (Wait " Alarm-Bus ")	Attente d'un signal «Alarme-Bus» la temporisation en fin d'émission d'un signal «Appel Général»
W.ETAB (Wait for end of " Alarm-Bus ")	Attente de la fin d'un signal «Alarme-Bus» reçu après l'émission d'un signal «Appel Général»
W.TASB	Attente du déclenchement du réveil TASB après le début de la réception d'un signal «Alarme-Bus»
W.TOL	Attente du déclenchement du réveil TOL
M.Send (Must Send)	Etat initial de l'émetteur, attente d'une trame à émettre
<i>T.Session</i>	Test du type de session (avec Station Secondaire alimentée ou télé-alimentée)
<i>SendFirst</i>	Envoi du premier octet de la trame à émettre
Sending	Etat récurrent de l'émetteur, émission octet par octet
<i>Answer</i>	Aiguillage en fonction du service demandé
M.Rec (Must Receive)	Etat initial du récepteur, attente du premier octet d'une trame
Receiving	Etat récurrent du récepteur, réception octet par octet
<i>Received</i>	Traitement de la trame reçue en fonction du service demandé
<i>T.RSO</i> (Test last RSO)	Test d'un intervalle de temps pour la réception d'une trame RSO
W.RSO (Wait for end of an RSO time slot)	Attente de la fin d'un intervalle de temps pour la réception d'une trame RSO
W.ASB	Attente de la fin d'émission d'un signal «Alarme Bus-Secondaire»
W.TOAG	Initialisation du délai de fin de session télé-alimentée TOAG si nécessaire
W.TOSEUIL	Attente du déclenchement du réveil TOSEUIL

Sending	octet_sent_event & Size=0 & FlagRSO	Stop_timer(TOE) Wait_time(TAO) FlagRSO=FALSE	WTOAG
Sending	Phy_ABORT.req()	Stop_timer(TOE)	WTOAG
Sending	time_out(TOE)	Phy_ABORT.ind(EP-3F)	WTOAG
W.ASB	time_out(TOAG)	MaxIndex=18 FirstWinRSO=TRUE	W.ASB
W.ASB	AG_sent_event	\$none()	Stopped
WTOAG	not(energized)	init_timer(TOAG)	Stopped
WTOAG	Energized	\$none()	Stopped

Table 8 – Meaning of the states listed in the previous tables

State	Meaning
<i>Initial</i>	Initialization of the variables of the layer
Stopped	Waiting state for a "Wakeup Call" signal
W.ETABS (Wait for end of "Alarm-Bus")	Waiting for the end of an "Alarm-bus" signal received in a Stopped state
W.AG (Wait for end of "Wakeup Call")	Waiting for the end of a "Wakeup Call" signal transmission
W.TAB (Wait "Alarm-Bus")	Waiting for an "Alarm-Bus" signal during safety delay at the end of transmission of a "Wakeup call" signal
W.ETAB (Wait for end of "Alarm-Bus")	Waiting for the end of an "Alarm-Bus" signal received after the transmission of a "Wakeup call" signal
W.TASB	Waiting for the triggering of wakeup TASB after the begin of the reception of an an "Alarm-Bus" signal
W.TOL	Waiting for the triggering of wakeup TOL
M.Send (Must Send)	Initial state of the transmitter waiting for a frame to send
<i>T.Session</i>	Testing the type of the session (with an energized or not energized Secondary Station)
<i>SendFirst</i>	Sending the first octet of the frame to be sent
Sending	Recurrent state of the transmitter transmitting one octet at a time
<i>Answer</i>	Branching depending on the service requested
M.Rec (Must Receive)	Initial state of the receiver waiting for the first octet of a frame
Receiving	Recurrent state of the receiver receiving one octet at a time
<i>Received</i>	Processing the received frame depending on the service requested
<i>T.RSO</i> (Test last RSO)	Testing the end of the last time slot for RSO frame reception
W.RSO (Wait for end of an RSO time slot)	Waiting for the end of a time slot for RSO frame reception
W.ASB	Waiting for the end of a "Alarm Secondary-Bus" signal transmission
W.TOAG	Initializing the "end of session" TOAG timer if needed
W.TOSEUIL	Waiting for the triggering of wakeup TOSEUIL

W.AGT	Attente d'un signal «Appel Général» AGT
W.Sel (Wait for preSelection)	Attente d'une trame de présélection
Select	Réception d'une trame de présélection
W.Answer	Attente d'une trame de réponse en provenance d'une station sélectionnée
Hide	Attente de la fin d'une sélection
W.Agend	Attente de la fin de la réception d'un AG
W.TVASB1	Attente du déclenchement du réveil TVASB pour un signal «Alarme Bus-Secondaire» durant une communication
W.TVASB2	Attente du déclenchement du réveil TVASB pour un signal «Alarme Bus-Secondaire» à la fin d'une communication
W.AB	Attente de la fin d'émission d'un signal «Alarme-Bus»

Tableau 9 – Définition des procédures, des fonctions et des événements classés dans l'ordre alphabétique

Procédure, fonction ou événement	Définition
AG_received_event	Événement issu du modem informant de la détection d'un signal «Appel Général» AGN
AG_sent_event	Événement issu du modem informant de la fin de l'émission d'un signal «Appel Général»
alarm_detection()	Vérification que la station a son mode alarme à Actif
collision_detected_event	Événement issu du modem informant de la détection d'une trame erronée sur réception d'un octet
concat(Frame, RecB)	Concaténation de l'octet RecB dans la trame Frame en cours de constitution
data_carrier_on, data_carrier_off	Occurrence de la détection de l'apparition ou de la disparition de la porteuse sur le bus
energized()	Vérification que la station est alimentée
init_timer(TOAPPEL), init_timer(TOSEUIL), init_timer(TAGT), init_timer(TOBAVARD), init_timer(TOPRE), init_timer(TOL), init_timer(TOE), init_timer(TAO), init_timer(TA1O), init_timer(TARSO), init_timer(TOAG), init_timer(TVASB) or init_timer(TAB)	Armement du réveil TOAPPEL, TOSEUIL, TAGT, TOBAVARD, TOPRE, TOL, TOE, TAO, TA1O, TARSO, TOAG, TVASB ou TAB

W.AGT	Waiting for an AGT “Wakeup Call” signal
W.Sel (Wait for preSelection)	Waiting for a preselection frame
Select	Receiving a preselection frame
W.Answer	Waiting for an answer frame from a selected station
Hide	Waiting for the end of selection
W.AGend	Waiting for the end of AG reception
W.TVASB1	Waiting for the triggering of wakeup TVASB for an “Alarm Secondary-Bus” signal during a session
W.TVASB2	Waiting for the triggering of wakeup TVASB for an “Alarm Secondary-Bus” signal at the end of session
W.AB	Waiting for the end of a “Alarm Bus” signal transmission

**Table 9 – Definition of the procedures, functions and events
classified in alphabetical order**

Procedure, function or event	Definition
AG_received_event	Event from the modem reporting that an AGN “Wakeup Call” signal has been correctly detected
AG_sent_event	Event from the modem reporting the end of the transmission of a “Wakeup Call” signal
alarm_detection()	Check that the station status of alarm mode is Active
collision_detected_event	Event from the modem reporting the detection of a framing error on reception of an octet
concat(Frame, RecB)	Concatenation of the octet RecB in the being built frame Frame
data_carrier_on, data_carrier_off	Occurrence of the detection on the bus of the data carrier on, the data carrier off
energized()	Check that the station is energized
init_timer(TOAPPEL), init_timer(TOSEUIL), init_timer(TAGT), init_timer(TOBAVARD), init_timer(TOPRE), init_timer(TOL), init_timer(TOE), init_timer(TAO), init_timer(TA1O), init_timer(TARSO), init_timer(TOAG), init_timer(TVASB) or init_timer(TAB)	setting of wakeup TOAPPEL, TOSEUIL, TAGT, TOBAVARD, TOPRE, TOL, TOE, TAO, TA1O, TARSO, TOAG, TVASB or TAB

occur(cpt_carrier_on), occur(cpt_carrier_off), occur(data_carrier_on), occur(data_carrier_off), occur(octet_received_event) or occur(octet_sent_event)	Occurrence (report sans consommation) de la détection de l'apparition ou de la disparition de la porteuse sur le bus secondaire, de la détection de l'apparition de la porteuse sur le bus, de la disparition de la porteuse, de la réception d'un octet ou de l'émission d'un octet
octet_received_event	Événement issu du modem informant qu'un octet a été reçu
octet_sent_event	Événement issu du modem informant qu'un octet a été émis
read_data(RecB)	Traitement de l'événement octet_received_event par lecture de l'octet RecB reçu (les bits sont transmis dans un ordre croissant)
send_AG(TypeAG)	Demande au modem d'effectuer l'émission d'un signal «Appel Général» d'une durée TypeAG (AGN ou AGT)
send_octet(Frame, Index)	Emission de l'octet de rang Index dans la trame Frame (les bits sont transmis dans un ordre croissant)
size(Frame)	Calcul du nombre d'octets de la trame Frame
station_power(ON) or station_power(OFF)	Met en marche ou à l'arrêt l'alimentation en énergie de l'équipement
station_signal(ON) or station_signal(OFF)	Met en marche ou à l'arrêt l'émission de signal vers l'équipement sur le bus secondaire
stop_timer(TOAPPEL), stop_timer(TOSEUIL), stop_timer(TAGT), stop_timer(TOBAVARD), stop_timer(TOPRE), stop_timer(TOL), stop_timer(TOE), stop_timer(TAO), stop_timer(TA10), stop_timer(TVASB) or stop_timer(TAB)	Désarmement du réveil TOAPPEL, TOSEUIL, TAGT, TOBAVARD, TOPRE, TOL, TOE, TAO, TA10, TVASB ou TAB
stop_timer(TOAG) or stop_timer(TARSO)	Désarmement du réveil TOAGT ou TARSO s'il a été armé au préalable
time_out(TOAPPEL), time_out(TOSEUIL), time_out(TAGT), time_out(TOBAVARD), time_out(TOPRE), time_out(TOL), time_out(TOE), time_out(TAO), time_out(TA10), time_out(TARSO), time_out(TOAG), time_out(TVASB) or stop_timer(TAB)	Déclenchement du réveil TOAPPEL, TOSEUIL, TAGT, TOBAVARD, TOPRE, TOL, TOE, TAO, TA10, TARSO, TOAG, TVASB ou TAB
wait_time(TAO), wait_time(TICB), wait_time(TOL) or wait_time(TOALR)	Temporisation pendant le temps TAO, TICB, TOL ou TOALR
wait_window(FirstWinRSO, Window)	Temporisation calculée de la façon suivante: FirstWinRSO=VRAI ou Window=0 ==> 0 ms FirstWinRSO=FAUX et Window>0 ==> 40 ms + (TARSO*Window) ms (Le délai de 40 ms permet de garantir que l'émission a bien lieu dans l'intervalle de temps prévu)

occur(cpt_carrier_on), occur(cpt_carrier_off), occur(data_carrier_on), occur(data_carrier_off), occur(octet_received_event) or occur(octet_sent_event)	Occurrence (reporting without consuming) of the detection on the secondary bus of the data carrier on, the data carrier off, on the bus of the data carrier on, the data carrier off, the reception of an octet or the emission of an octet
octet_received_event	Event from the modem reporting that an octet has been received
octet_sent_event	Event from the modem reporting that an octet has been sent
read_data(RecB)	Processing of the octet_received_event event by reading the received RecB octet (bits are transmitted in ascending order)
send_AG(TypeAG)	Request to the modem for transmission of a "Wakeup Call" signal of duration TypeAG (AGN or AGT)
send_octet(Frame, Index)	Transmission of the octet of rank Index in the frame Frame (bits are transmitted in ascending order)
size(Frame)	Calculation of the number of octets of the frame Frame
station_power(ON) or station_power(OFF)	Turning ON or OFF the energy supply to the device
station_signal(ON) or station_signal(OFF)	Turning ON or OFF the signal transmission to the device on the secondary bus
stop_timer(TOAPPEL), stop_timer(TOSEUIL), stop_timer(TAGT), stop_timer(TOBAVARD), stop_timer(TOPRE), stop_timer(TOL), stop_timer(TOE), stop_timer(TAO), stop_timer(TA1O), stop_timer(TVASB) or stop_timer(TAB)	Stopping of wakeup TOAPPEL, TOSEUIL, TAGT, TOBAVARD, TOPRE, TOL, TOE, TAO, TA1O, TVASB or TAB
stop_timer(TOAG) or stop_timer(TARSO)	Stopping of wakeup TOAG or TARSO only if it has previously been set
time_out(TOAPPEL), time_out(TOSEUIL), time_out(TAGT), time_out(TOBAVARD), time_out(TOPRE), time_out(TOL), time_out(TOE), time_out(TAO), time_out(TA1O), time_out(TARSO), time_out(TOAG), time_out(TVASB) or stop_timer(TAB)	Triggering of wakeup TOAPPEL, TOSEUIL, TAGT, TOBAVARD, TOPRE, TOL, TOE, TAO, TA1O, TARSO, TOAG, TVASB or TAB
wait_time(TAO), wait_time(TICB), wait_time(TOL) or wait_time(TOALR)	Calculated delay during time TAO, TICB, TOL or TOALR
wait_window(FirstWinRSO, Window)	Wait-time calculated as follows: when FirstWinRSO=TRUE or Window=0 ==> 0 ms when FirstWinRSO=FALSE and Window>0 ==> 40 ms + (TARSO*Window) ms (The 40 ms delay guarantees that the transmission has taken place in the time slot)

3.1.6 Répertoire et traitement des erreurs

Les erreurs sont répertoriées par le codage suivant:

EP = erreur de la couche *Physique*

– = séparateur

N = numéro de l'erreur

F = erreur fatale

Tableau 10 – Tableau récapitulatif des erreurs

EP-1	Délai du réveil TOL (Station Primaire) écoulé avant que la couche <i>Liaison</i> n'ait demandé l'envoi d'une trame, ou délai du réveil TA10 (Station Secondaire) écoulé avant la réception d'un caractère en provenance de la Station Primaire
	Cette erreur conduit à l'attente d'un signal «Appel Général» après avoir informé la couche <i>Liaison</i>
EP-2	Délai du réveil TOAG écoulé avant un signal «Appel Général»
	Cette erreur conduit à l'attente d'un signal «Appel Général» après avoir informé la couche <i>Liaison</i>
EP-3	Réception d'une alarme
	Cette erreur conduit à réinitialiser la couche <i>Physique</i> après avoir informé la couche <i>Liaison</i>
EP-3F	Durée anormale d'émission constatée après que le délai du réveil TOE est écoulé
	Cette erreur conduit à réinitialiser la couche <i>Physique</i> après avoir informé la couche <i>Liaison</i>
EP-4F	Nombre d'octets reçus supérieur à MaxIndex (Emetteur trop bavard)
	Cette erreur conduit à réinitialiser la couche <i>Physique</i> après avoir informé la couche <i>Liaison</i>
EP-5F	Délai du réveil TARSO écoulé en recevant une trame RSO (Station Primaire seulement)
	Cette erreur conduit à réinitialiser la couche <i>Physique</i> après avoir informé la couche <i>Liaison</i>

Toute occurrence de l'une de ces erreurs fatales est remontée localement grâce à la primitive de service *Phy_ABORT.ind*. La liste complète des numéros d'erreur fatale est fournie à l'annexe C.

3.1.6 List and processing of errors

Errors are listed using the following codes:

EP = error in the *Physical* layer

– = separator

N = error number

F = fatal error

Table 10 – Error summary table

EP-1	Expiry of TOL wakeup (Primary Station) before the <i>Data Link</i> layer requests a frame transmission or expiry of TA1O wakeup (Secondary Station) before any character has been received from the Primary station
	This error leads to the expectation of a “Wakeup Call” signal after having informed the <i>Data Link</i> layer
EP-2	Expiry of TOAG wakeup before any “Wakeup Call” signal
	This error leads to the expectation of a “Wakeup Call” signal after having informed the <i>Data Link</i> layer
EP-3	An alarm has been received
	This error leads to the reinitialization of the <i>Physical</i> layer after having informed the <i>Data Link</i> layer
EP-3F	Abnormal length of transmission detected after expiry of TOE wakeup
	This error leads to the reinitialization of the <i>Physical</i> layer after having informed the <i>Data Link</i> layer
EP-4F	Number of octets received higher than MaxIndex (Transmitter too talkative)
	This error leads to the reinitialization of the <i>Physical</i> layer after having informed the <i>Data Link</i> layer
EP-5F	Expiry of TARSO wakeup while receiving an RSO frame (Primary Station only)
	This error leads to the reinitialization of the <i>Physical</i> layer after having informed the <i>Data Link</i> layer

If any of these errors occurs, it is sent up locally by means of the *Phy_ABORT.ind* service primitive. The complete list of fatal error numbers is given in annex C.

3.2 Couche Liaison

3.2.1 Protocole Liaison-62056-31

Le protocole *Liaison-62056-31* de la couche *Liaison* de l'architecture d'échange de données par bus local sans DLMS adopte un comportement asymétrique. La machine d'état de la Station Primaire est donc différente de celle de la Station Secondaire.

La couche *Liaison* est chargée de transformer le canal physique exploité par la couche *Physique* en canal logique apte à transmettre de l'information fiable. Ses fonctions principales sont:

- effectuer une sérialisation et une désérialisation des données (dans la mesure où le canal physique fonctionne en série par bit);
- synchroniser les trames en émission et en réception;
- filtrer les trames en fonction des adresses primaire et secondaire;
- assurer une protection efficace contre les erreurs de transmission.

3.2.2 Gestion des échanges

Sur la Station Primaire, le protocole *Liaison-62056-31* prend en charge l'émission automatique des signaux «Appel Général» AGN ou AGT en fonction du type de la Station Secondaire. La détection d'une incompatibilité d'adresses dans un DSDU reçu de la couche supérieure entraîne une erreur fatale et l'arrêt du protocole *Liaison-62056-31*.

Sur la Station Secondaire, la réception d'une trame incorrecte ne donne lieu à aucun traitement, la récupération étant laissée à la charge de la Station Primaire.

Pour une station télé-alimentée, la détection d'un «Appel des Stations Oubliées» après un signal «Appel Général» AGT conduit à l'envoi d'une réponse RSO se situant toujours dans le premier intervalle de temps prévu à cet effet. La Station Primaire, lorsqu'elle détecte une collision au cours d'un tel échange, envoie un second «Appel des Stations Oubliées», mais cette fois-ci après un signal «Appel Général» AGN. Lorsqu'aucune collision n'est détectée après le premier appel, cela signifie qu'il n'y a au plus qu'une station oubliée, et il n'est pas nécessaire d'effectuer un deuxième appel.

3.2.3 Services et primitives de service de liaison

L'utilisateur du protocole *Liaison-62056-31* dispose des services et primitives de service donnés dans le tableau 11.

Tableau 11 – Services et primitives de service de liaison

Service	Primitive de service
DL_DATA	DL_DATA.req(DSDU) DL_DATA.ind(DSDU)
DL_ALARM	DL_ALARM.req() DL_ALARM.ind()
DL_ABORT	DL_ABORT.req() DL_ABORT.ind(ErrorNb)

3.2 Data Link layer

3.2.1 Link-62056-31 protocol

The *Link-62056-31* protocol of the *Data Link* layer of the local bus data exchange architecture without DLMS behaves asymmetrically. The state machine of the Primary Station is therefore different from that of the Secondary Station.

The *Data Link* layer transforms the physical channel used by the *Physical* layer to a logic channel able to transmit reliable information. Its main functions are:

- to carry out a serialization and a deserialization of the data (if the physical channel functions serially one bit at a time);
- to synchronize the transmission and reception frames;
- to filter the frames according to primary and secondary addresses;
- to ensure efficient protection against transmission errors.

3.2.2 Management of exchanges

On the Primary station, the *Link-62056-31* protocol takes over the transmission of an AGN or AGT “Wakeup Call” signal according to the type of Secondary Station. Detection of incompatibility in the addresses of a DSDU received from the upper layer indicates a fatal error and the stop of the *Link-62056-31* protocol.

On the Secondary Station, reception of an incorrect frame does not require any processing, as recovering is left to the Primary Station.

For non-energized stations, the detection of a “Forgotten Stations Call” after an AGT “Wakeup Call” signal leads to an RSO response which always takes place in the first RSO time slot. Thus, the Primary Station, when detecting a collision after such a sequence, performs a second “Forgotten Stations Call”, but this time after an AGN “Wakeup Call” signal. Nevertheless, when no collision is detected after the first call, there is one or no forgotten station and no need for a second call.

3.2.3 Data Link services and service primitives

The user of the *Link-62056-31* protocol can use the services and service primitives given in the table 11.

Table 11 – Data Link services and service primitives

Service	Service primitive
DL_DATA	DL_DATA.req(DSDU) DL_DATA.ind(DSDU)
DL_ALARM	DL_ALARM.req() DL_ALARM.ind()
DL_ABORT	DL_ABORT.req() DL_ABORT.ind(ErrorNb)

Le rôle attribué à chaque primitive est le suivant:

- DL_DATA.req(DSDU) permet à la couche *Application* de demander à la couche *Liaison* le transfert d'un paquet de données DSDU;
- DL_DATA.ind(DSDU) permet à la couche *Liaison* d'informer la couche *Application* de l'arrivée d'un paquet de données DSDU;
- DL_ALARM.req() permet à la couche *Application* de demander à la couche *Liaison* le transfert d'une alarme;
- DL_ALARM.ind() permet à la couche *Liaison* d'informer la couche *Application* de l'arrivée d'une alarme;
- DL_ABORT.req() permet à la couche *Application* de demander à la couche *Liaison* de mettre fin à son activité;
- DL_ABORT.ind(ErrorNb) permet à la couche *Liaison* d'informer la couche *Application* de l'occurrence d'une erreur fatale repérée par le numéro ErrorNb.

3.2.4 Paramètres de liaison

Pour une Station Primaire, la valeur du nombre MaxRetry de réémissions d'une même trame avant déconnexion est fixée à 2.

La valeur du nombre MaxChain de séquences enchaînées sans signal «Appel Général» pour la télérelève et le télétransfert, est fixée à 5, de manière à assurer la compatibilité avec les Stations Secondaires utilisant une version antérieure du protocole.

La valeur du nombre maximal MaxRSO d'intervalles de temps RSO pour le traitement d'une trame «Appel des Stations Oubliées» est fixée à 3 après un «Appel Général» AGN, et à 1 après un «Appel Général» AGT.

La Station Secondaire doit connaître la liste des adresses de Stations Primaires et la liste des TABi auxquels elle a été sensibilisée.

Une station peut être sollicitée par l'intermédiaire d'une adresse primaire générale APG. Dans ce cas, elle répond avec la première adresse de la liste des adresses de Stations Primaires auxquelles elle a été sensibilisée.

The role assigned to each primitive is as follows:

- DL_DATA.req(DSDU) enables the *Application* layer to request the *Data Link* layer to transfer a DSDU data packet;
- DL_DATA.ind(DSDU) enables the *Data Link* layer to inform the *Application* layer of the arrival of a DSDU data packet;
- DL_ALARM.req() enables the *Application* layer to request the *Data Link* layer to transfer an alarm;
- DL_ALARM.ind() enables the *Data Link* layer to inform the *Application* layer of the arrival of an alarm;
- DL_ABORT.req() enables the *Application* layer to request the *Data Link* layer to end its activity;
- DL_ABORT.ind(ErrorNb) enables the *Data Link* layer to inform the *Application* layer of the occurrence of a fatal error identified by the number ErrorNb.

3.2.4 Data Link parameters

For the Primary Station, the value of the number of repeat transmissions for a given frame before disconnection, MaxRetry, is set to 2.

The value of the number of sequences linked with no “Wakeup Call” signal for remote reading and remote transfer, MaxChain, is set to 5, for compatibility with Secondary Stations using a previous version of the protocol.

The value of the maximum number, MaxRSO, of RSO time slots for the processing of a “Forgotten Stations Call” is set to 3 after an AGN “Wakeup Call” signal, at 1 after an AGT “Wakeup Call” signal.

The Secondary Station shall know the list of Primary Station addresses and the list of TABi to which it has been programmed.

The station may also be solicited by the general primary address APG. In this case, it replies with the first primary address to which it has been programmed.

3.2.5 Transitions d'état

Tableau 12 – Transitions d'état de *Liaison-62056-31*: Station Primaire

Etat initial	Condition de déclenchement	Ensemble d'actions	Etat final
<i>Initial</i>	\$true()	MaxRetry=2 MaxChain=5	Stopped
Stopped	DL_DATA.req(DSDU) & check_req(DSDU)	NbChain=0 MaxRSO=3 PreSel=FALSE NoRetry=FALSE RepeatASO=FALSE EP-1=FALSE Context(ADS, ADP, TypeAG) Com=command(DSDU) init(Com, TypeAG) Phy_APPG.req(TypeAG)	W.AG
Stopped	DL_DATA.req(DSDU) & not(check_req(DSDU))	DL_ABORT.ind(EL-1F)	Stopped
Stopped	Phy_ALARM.ind()	DL_ALARM.ind()	Stopped
W.AG	Phy_APPG.ind() & not(PreSel) & not(NoRetry) & not(RepeatASO)	\$none()	<i>T.Reg</i>
W.AG	Phy_APPG.ind() & PreSel	Index=MaxRetry+1 Fr=PRE Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr)	M.Rec
W.AG	Phy_APPG.ind() & NoRetry	Index=MaxRetry+1 Fr=DSDU Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) NbChain=1 Phy_DATA.req(Fr) NoRetry=FALSE	M.Rec
W.AG	Phy_APPG.ind() & RepeatASO	RepeatASO=FALSE Phy_ASO.req(Fr)	M.RSO
W.AG	DL_ABORT.req()	Phy_ABORT.req()	W.EndS
W.EndS	Phy_ABORT.ind(ErrorNb)	\$none()	<i>T.Error</i>
W.EndS	Phy_ALARM.ind()	DL_ALARM.ind()	Stopped
<i>T.Error</i>	((Error_Nb = EP-1 & TypeAG = AGN) (Error_Nb = EP-2 & TypeAG = AGT)) & Com <> IB & Com <> TRB	\$none()	Stopped
<i>T.Error</i>	((Error_Nb = EP-1 & TypeAG = AGN) (Error_Nb = EP-2 & TypeAG = AGT)) & (Com=IB Com=TRB)	DL_ABORT.ind(Error_Nb)	Stopped
<i>T.Error</i>	Error_Nb <> EP-1 & Error_Nb <> EP-2	DL_ABORT.ind(Error_Nb)	W.EndS
<i>T.Error</i>	(Error_Nb = EP-1) & TypeAG = AGT	\$none()	W.EndS
<i>T.Reg</i>	Com=IB Com=TRB	Fr=DSDU Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Phy_UNACK.req(Fr)	W.EndS

3.2.5 State transitions

Table 12 – Link-62056-31 state transitions: Primary Station

Initial state	Triggering condition	Set of actions	Final state
<i>Initial</i>	\$true()	MaxRetry=2 MaxChain=5	Stopped
Stopped	DL_DATA.req(DSDU) & check_req(DSDU)	NbChain=0 MaxRSO=3 PreSel=FALSE NoRetry=FALSE RepeatASO=FALSE EP-1=FALSE context(ADS, ADP, TypeAG) Com=command(DSDU) init(Com, TypeAG) Phy_APPG.req(TypeAG)	W.AG
Stopped	DL_DATA.req(DSDU) & not(check_req(DSDU))	DL_ABORT.ind(EL-1F)	Stopped
Stopped	Phy_ALARM.ind()	DL_ALARM.ind()	Stopped
W.AG	Phy_APPG.ind() & not(PreSel) & not(NoRetry) & not(RepeatASO)	\$none()	<i>T.Req</i>
W.AG	Phy_APPG.ind() & PreSel	Index=MaxRetry+1 Fr=PRE Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr)	M.Rec
W.AG	Phy_APPG.ind() & NoRetry	Index=MaxRetry+1 Fr=DSDU Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) NbChain=1 Phy_DATA.req(Fr) NoRetry=FALSE	M.Rec
W.AG	Phy_APPG.ind() & RepeatASO	RepeatASO=FALSE Phy_ASO.req(Fr)	M.RSO
W.AG	DL_ABORT.req()	Phy_ABORT.req()	W.EndS
W.EndS	Phy_ABORT.ind(ErrorNb)	\$none()	<i>T.Error</i>
W.EndS	Phy_ALARM.ind()	DL_ALARM.ind()	Stopped
<i>T.Error</i>	((Error_Nb = EP-1 & TypeAG = AGN) (Error_Nb = EP-2 & TypeAG = AGT)) & Com <> IB & Com <> TRB	\$none()	Stopped
<i>T.Error</i>	((Error_Nb = EP-1 & TypeAG = AGN) (Error_Nb = EP-2 & TypeAG = AGT)) & (Com=IB Com=TRB)	DL_ABORT.ind(Error_Nb)	Stopped
<i>T.Error</i>	Error_Nb <> EP-1 & Error_Nb <> EP-2	DL_ABORT.ind(Error_Nb)	W.EndS
<i>T.Error</i>	(Error_Nb = EP-1) & TypeAG = AGT	\$none()	W.EndS
<i>T.Req</i>	Com=IB Com=TRB	Fr=DSDU Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Phy_UNACK.req(Fr)	W.EndS

<i>T.Req</i>	Com=ASO & TypeAG=AGN	MaxRSO=3 NbRSO=1 ListRSO="" Collision=FALSE Fr=DSDU Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Phy_ASO.req(Fr)	M.RSO
<i>T.Req</i>	Com=ASO & TypeAG=AGT	MaxRSO=1 NbRSO=1 ListRSO="" Collision=FALSE Fr=DSDU Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Phy_ASO.req(Fr)	M.RSO
<i>T.Req</i>	((NbChain<MaxChain) & (Com=ENQ Com=TRF)) (NbChain=0 & Com=REC)	Fr=DSDU Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Index=1 NbChain=NbChain+1 Phy_DATA.req(Fr)	M.Rec
<i>T.Req</i>	Com=AUT	Fr=DSDU Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Index=1 NbChain=MaxChain Phy_DATA.req(Fr)	M.Rec
<i>T.Req</i>	(NbChain>=MaxChain) (NbChain<>0 & Com=REC)	NbChain=0 Phy_APPG.req(AGN)	W.AG
M.Rec	Phy_DATA.ind(Frame) & check_frame(Frame) & not(PreSel)	DSDU=extract_DSDU(Frame) DL_DATA.ind(DSDU)	M.Send
M.Rec	Phy_DATA.ind(Frame) & check_frame(Frame) & command(Frame)=SEL & PreSel	PreSel=FALSE	<i>T.Req</i>
M.Rec	Phy_DATA.ind(Frame) & check_frame(Frame) & command(Frame)<>SEL & PreSel	Phy_ABORT.req() DL_ABORT.ind(EL-2F)	W.EndS
M.Rec	Phy_DATA.ind(Frame) & not(check_frame(Frame)) & Index<=MaxRetry	Index=Index+1 Phy_DATA.req(Fr)	M.Rec
M.Rec	Phy_DATA.ind(Frame) & not(check_frame(Frame)) & Index>MaxRetry	Phy_ABORT.req() DL_ABORT.ind(EL-2F)	W.EndS
M.Rec	DL_ABORT.req()	Phy_ABORT.req()	W.EndS
M.Rec	Phy_ABORT.ind(ErrorNb)	DL_ABORT.ind(ErrorNb)	W.EndS
M.RSO	Phy_ASO.ind(Frame) & size(Frame)=0	\$none()	<i>T.RSO</i>
M.RSO	Phy_ASO.ind(Frame) & size(Frame)<>0 & check_frame(Frame) & command(Frame)=RSO	build_RSO(ListRSO, Frame)	<i>T.RSO</i>
M.RSO	Phy_ASO.ind(Frame) & size(Frame)<>0 & not(check_frame(Frame))	Collision=TRUE	<i>T.RSO</i>
M.RSO	Phy_COLL.ind()	Collision=TRUE	<i>T.RSO</i>
M.RSO	DL_ABORT.req()	Phy_ABORT.req()	W.EndS
M.RSO	Phy_ABORT.int(ErrorNb)	DL_ABORT.ind(ErrorNb)	W.EndS

<i>T.Reg</i>	Com=ASO & TypeAG=AGN	MaxRSO=3 NbRSO=1 ListRSO="" Collision=FALSE Fr=DSDU Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Phy_ASO.req(Fr)	M.RSO
<i>T.Reg</i>	Com=ASO & TypeAG=AGT	MaxRSO=1 NbRSO=1 ListRSO="" Collision=FALSE Fr=DSDU Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Phy_ASO.req(Fr)	M.RSO
<i>T.Reg</i>	((NbChain<MaxChain) & (Com=ENQ Com=TRF)) (NbChain=0 & Com=REC)	Fr=DSDU Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Index=1 NbChain=NbChain+1 Phy_DATA.req(Fr)	M.Rec
<i>T.Reg</i>	Com=AUT	Fr=DSDU Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Index=1 NbChain=MaxChain Phy_DATA.req(Fr)	M.Rec
<i>T.Reg</i>	(NbChain>=MaxChain) (NbChain<>0 & Com=REC)	NbChain=0 Phy_APPG.req(AGN)	W.AG
M.Rec	Phy_DATA.ind(Frame) & check_frame(Frame) & not(PreSel)	DSDU=extract_DSDU(Frame) DL_DATA.ind(DSDU)	M.Send
M.Rec	Phy_DATA.ind(Frame) & check_frame(Frame) & command(Frame)=SEL & PreSel	PreSel=FALSE	<i>T.Reg</i>
M.Rec	Phy_DATA.ind(Frame) & check_frame(Frame) & command(Frame)<>SEL & PreSel	Phy_ABORT.req() DL_ABORT.ind(EL-2F)	W.EndS
M.Rec	Phy_DATA.ind(Frame) & not(check_frame(Frame)) & Index<=MaxRetry	Index=Index+1 Phy_DATA.req(Fr)	M.Rec
M.Rec	Phy_DATA.ind(Frame) & not(check_frame(Frame)) & Index>MaxRetry	Phy_ABORT.req() DL_ABORT.ind(EL-2F)	W.EndS
M.Rec	DL_ABORT.req()	Phy_ABORT.req()	W.EndS
M.Rec	Phy_ABORT.ind(ErrorNb)	DL_ABORT.ind(ErrorNb)	W.EndS
M.RSO	Phy_ASO.ind(Frame) & size(Frame)=0	\$none()	<i>T.RSO</i>
M.RSO	Phy_ASO.ind(Frame) & size(Frame)<>0 & check_frame(Frame) & command(Frame)=RSO	build_RSO(ListRSO, Frame)	<i>T.RSO</i>
M.RSO	Phy_ASO.ind(Frame) & size(Frame)<>0 & not(check_frame(Frame))	Collision=TRUE	<i>T.RSO</i>
M.RSO	Phy_COLL.ind()	Collision=TRUE	<i>T.RSO</i>
M.RSO	DL_ABORT.req()	Phy_ABORT.req()	W.EndS
M.RSO	Phy_ABORT.int(ErrorNb)	DL_ABORT.ind(ErrorNb)	W.EndS

<i>T.RSO</i>	MaxRSO=1 & Collision	MaxRSO=3 Collision=FALSE RepeatASO=TRUE Phy_APPG.req(AGN)	W.AG
<i>T.RSO</i>	(MaxRSO=1 & not(Collision)) (MaxRSO<>1 & NbRSO>=MaxRSO)	DSDU=rso(RSO, Collision, ListRSO) DL_DATA.ind(DSDU)	W.EndS
<i>T.RSO</i>	NbRSO<MaxRSO	NbRSO=NbRSO+1	M.RSO
M.Send	DL_DATA.req(DSDU) & check_req(DSDU) & not(EP-1)	Com=command(DSDU)	<i>T.Req</i>
M.Send	DL_DATA.req(DSDU) & check_req(DSDU) & EP-1	Com=command(DSDU) NbChain=0 EP-1=FALSE Phy_APPG.req(AGN)	W.AG
M.Send	DL_DATA.req(DSDU) & not(check_req(DSDU))	Phy_ABORT.req() DL_ABORT.ind(EL-1F)	W.EndS
M.Send	Phy_ABORT.ind(EP-1)	EP-1=TRUE	M.Send
M.Send	Phy_ABORT.ind(EP-2)	\$none()	Stopped
M.Send	DL_ABORT.req() & not(EP_1 & TypAG=AGN)	Phy_ABORT.req()	W.EndS
M.Send	DL_ABORT.req() & EP_1 & TypAG=AGN	\$none()	Stopped
M.Send	Phy_ABORT.ind(ErrorNb) & ErrorNb<>EP-1 & ErrorNb<>EP-2	DL_ABORT.ind(ErrorNb)	W.EndS

<i>T.RSO</i>	MaxRSO=1 & Collision	MaxRSO=3 Collision=FALSE RepeatASO=TRUE Phy_APPG.req(AGN)	W.AG
<i>T.RSO</i>	(MaxRSO=1 & not(Collision)) (MaxRSO<>1 & NbRSO>=MaxRSO)	DSDU=rso(RSO, Collision, ListRSO) DL_DATA.ind(DSDU)	W.EndS
<i>T.RSO</i>	NbRSO<MaxRSO	NbRSO=NbRSO+1	M.RSO
M.Send	DL_DATA.req(DSDU) & check_req(DSDU) & not(EP-1)	Com=command(DSDU)	<i>T.Req</i>
M.Send	DL_DATA.req(DSDU) & check_req(DSDU) & EP-1	Com=command(DSDU) NbChain=0 EP-1=FALSE Phy_APPG.req(AGN)	W.AG
M.Send	DL_DATA.req(DSDU) & not(check_req(DSDU))	Phy_ABORT.req() DL_ABORT.ind(EL-1F)	W.EndS
M.Send	Phy_ABORT.ind(EP-1)	EP-1=TRUE	M.Send
M.Send	Phy_ABORT.ind(EP-2)	\$none()	Stopped
M.Send	DL_ABORT.req() & not(EP_1 & TypAG=AGN)	Phy_ABORT.req()	W.EndS
M.Send	DL_ABORT.req() & EP_1 & TypAG=AGN	\$none()	Stopped
M.Send	Phy_ABORT.ind(ErrorNb) & ErrorNb<>EP-1 & ErrorNb<>EP-2	DL_ABORT.ind(ErrorNb)	W.EndS

Tableau 13 – Transitions d'état de *Liaison-62056-31: Station Secondaire*

Etat initial	Condition de déclenchement	Ensemble d'actions	Etat final
<i>Initial</i>	\$true()	FlagDSO=TRUE Discovered=FALSE Flag_alarm=FALSE	Stopped
Stopped	Phy_DATA.ind(Frame) & check_frame(Frame) & check_address(Frame)	ADP=extract_ADP(Frame) Com=command(Frame)	<i>T.Com</i>
Stopped	Phy_DATA.ind(Frame) & check_frame(Frame) & not(check_address(Frame))	Phy_ABORT.req()	Stopped
Stopped	Phy_DATA.ind(Frame) & not(check_frame(Frame))	\$none()	Stopped
Stopped	Phy_ABORT.ind(ErrorNb)	DL_ABORT.ind(ErrorNb)	Stopped
Stopped	DL_ALARM.req()	Phy_ALARM.req() Flag_alarm=TRUE	Stopped
<i>T.Com</i>	Com=IB	FlagDSO=TRUE Discovered=FALSE Phy_ABORT.req()	Stopped
<i>T.Com</i>	Com=TRB	DSDU=extract_DSDU(Frame) DL_DATA.ind(DSDU) Phy_ABORT.req()	Stopped
<i>T.Com</i>	Com=PRE	Fr=SEL Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr)	Stopped
<i>T.Com</i>	Com=ASO & test_TABi(Frame, TAB)	Fr=concat(RSO, TAB, ADS) Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Phy_RSO.req(Fr, window_RSO())	Stopped
<i>T.Com</i>	Com=ASO & not(test_TABi(Frame, TAB))	Phy_ABORT.req()	Stopped
<i>T.Com</i>	Com=ENQ Com=REC Com=TRF COM=AUT	DSDU=extract_DSDU(Frame) DL_DATA.ind(DSDU)	M.Send
M.Send	DL_DATA.req(DSDU)	Fr=DSDU Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr) update_flag_DSO(command(Fr))	Stopped
M.Send	Phy_ABORT.ind(ErrorNb)	DL_ABORT.ind(ErrorNb)	Stopped

Table 13 – Link-62056-31 state transitions: Secondary Station

Initial state	Triggering condition	Set of actions	Final state
<i>Initial</i>	\$true()	FlagDSO=TRUE Discovered=FALSE Flag_alarm = FALSE	Stopped
Stopped	Phy_DATA.ind(Frame) & check_frame(Frame) & check_address(Frame)	ADP=extract_ADP(Frame) Com=command(Frame)	<i>T.Com</i>
Stopped	Phy_DATA.ind(Frame) & check_frame(Frame) & not(check_address(Frame))	Phy_ABORT.req()	Stopped
Stopped	Phy_DATA.ind(Frame) & not(check_frame(Frame))	\$none()	Stopped
Stopped	Phy_ABORT.ind(ErrorNb)	DL_ABORT.ind(ErrorNb)	Stopped
Stopped	DL_ALARM.req()	Phy_ALARM.req() Flag_alarm = TRUE	Stopped
<i>T.Com</i>	Com=IB	FlagDSO=TRUE Discovered=FALSE Phy_ABORT.req()	Stopped
<i>T.Com</i>	Com=TRB	DSDU=extract_DSDU(Frame) DL_DATA.ind(DSDU) Phy_ABORT.req()	Stopped
<i>T.Com</i>	Com=PRE	Fr=SEL Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr)	Stopped
<i>T.Com</i>	Com=ASO & test_TABi(Frame, TAB)	Fr=concat(RSO, TAB, ADS) Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Phy_RSO.req(Fr, window_RSO())	Stopped
<i>T.Com</i>	Com=ASO & not(test_TABi(Frame, TAB))	Phy_ABORT.req()	Stopped
<i>T.Com</i>	Com=ENQ Com=REC Com=TRF COM=AUT	DSDU=extract_DSDU(Frame) DL_DATA.ind(DSDU)	M.Send
M.Send	DL_DATA.req(DSDU)	Fr=DSDU Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr) update_flag_DSO(command(Fr))	Stopped
M.Send	Phy_ABORT.ind(ErrorNb)	DL_ABORT.ind(ErrorNb)	Stopped

Tableau 14 – Signification des états mentionnés dans les tableaux précédents

Etat	Signification
<i>Initial</i>	Initialisation des variables de la couche
Stopped	Attente de la première demande de la couche supérieure ou de la première indication de la couche inférieure
W.AG (Wait for end of " Wakeup Call ")	Attente de la fin d'un signal «Appel Général» demandé
W.EndS (Wait for End of Session)	Attente de la fin d'une session
T.Reg (Test Request)	Test de la nature d'une demande en provenance d'une couche supérieure
M.Rec (Must Receive)	Attente d'une réponse en provenance de la couche inférieure
M.RSO (Must receive RSO)	Attente d'une trame RSO après l'envoi d'une trame ASO
T.RSO (Test last RSO)	Test du dernier intervalle de temps pour la réception d'une trame RSO
M.Send (Must Send)	Attente d'une requête de la couche supérieure
T.Com (Test Command)	Test du code de commande d'une trame reçue

Table 14 – Meaning of the states listed in the previous tables

State	Meaning
<i>Initial</i>	Initialization of the variables of the layer
Stopped	Waiting state for the first request from the upper layer or the first indication from the lower layer
W.AG (Wait for end of "Wakeup Call")	Waiting for the end of a requested "Wakeup Call" signal
W.EndS (Wait for End of Session)	Waiting for the end of the session
<i>T.Req</i> (Test Request)	Test the nature of a request from the upper layer
M.Rec (Must Receive)	Waiting state for an indication from the lower layer
M.RSO (Must receive RSO)	Waiting for an RSO frame following an ASO frame sent
<i>T.RSO</i> (Test last RSO)	Testing the end of the last time slot for RSO frame reception
M.Send (Must Send)	Waiting state for a request from the upper layer
<i>T.Com</i> (Test Command)	Test of the command code of a received frame

Tableau 15 – Définition des procédures et des fonctions classées dans l'ordre alphabétique

Procédure ou fonction	Définition
build_RSO(ListRSO, Frame)	Extraction des éléments (champs TAB et ADS) de la trame RSO reçue Frame et concaténation avec la précédente liste ListRSO
check_address(Frame)	Vérification que les adresses ADP et ADS sont reconnues selon les critères suivants: – la station est sensibilisée à l'ADP ou ADP=APG; – ADS=ADG si la commande est ASO,IB ou TRB, – ADS est l'adresse de la station secondaire si la commande n'est pas ASO, IB ou TRB
check_frame(Frame)	Vérification que la trame Frame reçue est correcte: – nombre d'octets supérieur ou égal à 11 et inférieur ou égal à 128; – CRC correct; – nombre d'octets compatible avec le champ N – commande connue et nombre d'octets compatible avec cette commande
check_req(DSDU)	Vérification que la commande du DSDU est compatible avec les adresses ADP et ADS du contexte de communication
command(DSDU) ou command(Frame)	Extraction de la valeur de la commande codée dans le DSDU à envoyer ou de la trame Frame reçue
concat(N, ADS, ADP, DSDU) ou concat(Frame, CRC)	Concaténation des champs N, ADS, ADP et du DSDU ou concaténation du CRC à la fin de la trame Frame
context(ADS, ADP, TypeAG)	Extraction des valeurs correspondantes du contexte de communication
crc(Frame)	Calcul du CRC de la trame Frame à émettre
extract_ADP(Frame)	Si l'adresse de la Station Primaire utilisée dans la trame est différente de APG ou bien s'il s'agit de APG mais que la liste des adresses primaires auxquelles est sensibilisée la Station Secondaire est vide, extraction de cette valeur, sinon extraction de la première valeur ADP de la liste des adresses primaires auxquelles est sensibilisée la Station Secondaire
extract_DSDU(Frame)	Extraction du DSDU (champs COM et DATA) de la trame Frame reçue
init(COM, TypeAG)	Met PreSel à VRAI si on a à la fois TypeAG égal à AGT et une trame de taille supérieure à 18 octets. Met NoRetry à VRAI si on a à la fois TypeAG égal à AGT et une trame de taille inférieure ou égale à 18 octets

**Table 15 – Definition of the procedures and functions
classified in alphabetical order**

Procedure or function	Definition
build_RSO(ListRSO, Frame)	Extraction of the RSO elements (TAB and ADS fields) from the received RSO frame Frame and concatenation with the preceding list ListRSO
check_address(Frame)	Check that the ADP and ADS addresses are recognized according to the following criteria: – ADP is APG or the station has been programmed to the ADP address; – if the command code is ASO, IB or TRB, then ADS is ADG; – if the command code is not ASO, IB nor TRB, then ADS is the secondary station address
check_frame(Frame)	Check that the frame Frame received is correct: – number of octets greater than or equal to 11 and lower than or equal to 128; – CRC correct; – number of octets compatible with the field N; – command code recognized and number of octets compatible with the command code
check_req(DSDU)	Check that the requested command code inside a DSDU is compatible with ADP and ADS addresses defined in the communication context
command(DSDU) or command(Frame)	Extraction of the value of the command code of a DSDU to send or a received frame Frame
concat(N, ADS, ADP, DSDU) or concat(Frame, CRC)	Concatenation of the fields N, ADS and ADP with the DSDU or concatenation of the CRC at the end of the frame Frame
context(ADS, ADP, TypeAG)	Extraction of the corresponding values from the communication context
crc(Frame)	Calculation of the CRC of the frame Frame to send
extract_ADP(Frame)	If either the ADP value used in the frame is not APG or if it is the APG value but the list of the ADP values to which the Secondary Station has been programmed is empty, then extraction of this value, else extraction of the first ADP value to which the Secondary Station has been programmed
extract_DSDU(Frame)	Extraction of the DSDU (COM and DATA fields) from the frame Frame received
init(COM, TypeAG)	Set PreSel to TRUE if both TypeAG equals AGT and the size of the frame is greater than 18 octets. Set NoRetry to TRUE if both TypeAG equals AGT and the size of the frame is lower than or equal to 18 octets.

rso(RSO, Collision, ListRSO)	Concaténation de la commande RSO, de l'indicateur de collision et de la liste d'éléments RSO (champs TAB et ADS)
size(Frame)	Calcul de la taille de la trame Frame reçue
size_frame(DSDU)	Calcul de la taille de la trame associée à un DSDU à émettre (taille du DSDU + 10)
test_TABi(Frame, TAB)	Si le premier des TABi contenus dans la trame ASO Frame vaut 00, vérification que la variable Discovered est à FAUX et vérification, après tirage d'un nombre aléatoire entre 0 et 100, que ce nombre est inférieur à la probabilité de réponse souhaitée au deuxième des TABi. Dans ce cas la valeur 00 est mémorisée dans la variable TAB. Si le premier TABi contenu dans la trame ASO vaut FF, vérification que la variable Flag_alarm est à VRAI et dans ce cas affectation de la valeur FAUX à la variable Flag_alarm et mémorisation de la valeur FF dans la variable TAB. sinon vérification que la variable FlagDSO est à VRAI, que la Station Secondaire est sensibilisée à l'un des TABi contenus dans la trame Frame de type ASO reçue et mémorisation de la première de ces valeurs dans la variable TAB
update_flag_DSO(Com)	Affectation de la valeur FAUX à la variable FlagDSO et de la valeur VRAI à la variable Discovered si COM est la commande ENQ
window_RSO()	Fourniture d'une valeur aléatoire entière comprise entre 0 et MaxRSO-1 utilisée comme numéro de l'intervalle de temps RSO dans laquelle la station doit répondre (voir l'annexe F)

rso(RSO, Collision, ListRSO)	Concatenation of the RSO command code, the collision indicator and the list of the RSO elements (TAB and ADS fields)
size(Frame)	Calculation of the size of the frame Frame received
size_frame(DSDU)	Calculation of the size of the frame associated with a DSDU to be sent ($\text{size}(\text{DSDU}) + 10$)
test_TABi(Frame, TAB)	If the first TABi contained in the received ASO frame Frame equals 00, check that Discovered=FALSE then, after provision of a whole random integer between 0 and 100, checks that this integer is smaller than the response probability (second TABi). In this case, 00 is memorized in the TAB variable; if the first TABi contained in the received ASO frame Frame equals FF, check that Flag_alarm=TRUE then, in this case, Flag_alarm is set to FALSE and FF is memorized in the TAB variable; If the first TABi contained in the received ASO frame Frame does not equal 00 or FF, check that FlagDSO=TRUE and check that the Secondary Station has been programmed to one of the TABi contained in the received ASO frame Frame. In this case, the first of these values is memorized in the TAB variable
update_flag_DSO(COM)	Set FlagDSO to FALSE and Discovered to TRUE if COM equals ENQ
window_RSO()	Provision of a whole random integer between 0 and MaxRSO-1 used as the number of the RSO time slot in which the station shall reply (refer to annex F)

3.2.6 Répertoire et traitement des erreurs

Les erreurs sont répertoriées par le codage suivant:

- EL = erreur de la couche *Liaison*
- = séparateur
- N = numéro de l'erreur
- F = erreur fatale

Tableau 16 – Tableau récapitulatif des erreurs

EL-1F	Réception d'une commande incompatible avec les adresses primaire et secondaire mémorisées dans le contexte de communication (Station Primaire seulement)
	Cette erreur conduit à réinitialiser la couche <i>Liaison</i> après avoir informé la couche <i>Application</i>
EL-2F	Réponse incorrecte de la Station Secondaire après soit une trame de présélection pour une station télé-alimentée, soit MaxRetry répétitions d'une requête (Station Primaire seulement)
	Cette erreur conduit à réinitialiser la couche <i>Liaison</i> après avoir informé la couche <i>Application</i>

Toute occurrence de l'une de ces erreurs fatales est remontée localement grâce à la primitive de service DL_ABORT.ind. La liste complète des numéros d'erreur fatale est fournie à l'annexe C.

3.3 Couche Application

3.3.1 Protocole Application-62056-31

Le protocole *Application-62056-31* de la couche *Application* de l'architecture d'échange de données par bus local sans DLMS adopte un comportement asymétrique. La machine d'état de la Station Primaire est donc différente de celle de la Station Secondaire.

Le protocole *Application-62056-31* de la couche *Application* de l'architecture d'échange de données par bus local sans DLMS est chargé de contrôler et d'enchaîner les messages successifs par analyse d'un code commande fourni par l'application utilisatrice.

3.3.2 Services et primitives de service de Application

L'utilisateur du protocole *Application-62056-31* dispose des services et primitives de service donnés dans le tableau 17.

3.2.6 List and processing of errors

Errors are listed using the following codes:

EL = error in the *Data Link* layer

– = separator

N = error number

F = fatal error

Table 16 – Error summary table

EL-1F	Reception of a command code not compatible with the ADP and ADS addresses memorized in the communication context (Primary Station only)
	This error leads to the reinitialization of the <i>Data Link</i> layer after having informed the <i>Application</i> layer
EL-2F	Incorrect response from the Secondary Station after either a non-energized station preselection frame or MaxRetry repeated transmissions of a request (Primary Station only)
	This error leads to the reinitialization of the <i>Data Link</i> layer after having informed the <i>Application</i> layer

If any of these errors occurs, it is sent up locally by means of the DL_ABORT.ind service primitive. The complete list of fatal error numbers is given in annex C.

3.3 Application layer

3.3.1 Application-62056-31 protocol

The *Application-62056-31* protocol of the *Application* layer of the local bus data exchange architecture without DLMS behaves asymmetrically. The state machine of the Primary Station is therefore different from that of the Secondary Station.

The *Application-62056-31* protocol of the *Application* layer of the local bus data exchange architecture without DLMS controls and links successive messages by analysing the command code supplied by the user application.

3.3.2 Application services and service primitives

The user of the *Application-62056-31* protocol can use the services and service primitives given in the table 17.

Tableau 17 – Services et primitives de service de Application

Service	Primitive de service
A_DATA	A_DATA.req(COM, ASDU) A_DATA.ind(ASDU)
A_ALARM	A_ALARM.req() A_ALARM.ind()
A_ABORT	A_ABORT.req() A_ABORT.ind(ErrorNb)

Le rôle attribué à chaque primitive est le suivant:

- A_DATA.req(COM, ASDU) permet à l'application de demander à la couche *Application* le transfert d'une commande COM (ENQ, REC, TRF, TRB, IB ou ASO pour une Station Primaire et DAT, DRJ, EOS ou TRA pour une Station Secondaire) associée à un ASDU;
- A_DATA.ind(ASDU) permet à la couche *Application* d'informer l'application de l'arrivée d'un ASDU;
- A_ALARM.req() permet à l'application de demander à la couche *Application* l'envoi d'une alarme;
- A_ALARM.ind() permet à la couche *Application* d'informer l'application de l'arrivée d'une alarme;
- A_ABORT.req() permet à l'application de demander à la couche *Application* de mettre fin à son activité;
- A_ABORT.ind(ErrorNb) permet à la couche *Application* d'informer l'application de l'occurrence d'une erreur fatale repérée par le numéro ErrorNb.

3.3.3 Paramètres de Application

La Station Primaire doit posséder les clefs de cryptage DES associées à chacune des Stations Secondaires sur lesquelles une téléprogrammation doit être réalisée.

Dans le cas d'une téléprogrammation, la Station Secondaire doit posséder la clef de cryptage DES utilisée par la Station Primaire.

Table 17 – Application services and service primitives

Service	Service primitive
A_DATA	A_DATA.req(COM, ASDU) A_DATA.ind(ASDU)
A_ALARM	A_ALARM.req() A_ALARM.ind()
A_ABORT	A_ABORT.req() A_ABORT.ind(ErrorNb)

The role assigned to each primitive is as follows:

- A_DATA.req(COM, ASDU) enables the application to request the *Application* layer to transfer a COM (ENQ, REC, TRF, TRB, IB or ASO for a Primary Station and DAT, DRJ, EOS or TRA for a Secondary Station) command code linked with an ASDU information unit;
- A_DATA.ind(ASDU) enables the *Application* layer to inform the application of the arrival of an ASDU information unit;
- A_ALARM.req() enables the application to request the *Application* layer to send an alarm;
- A_ALARM.ind() enables the *Application* layer to inform the application of the arrival of an alarm;
- A_ABORT.req() enables the application to request the *Application* layer to end its activity;
- A_ABORT.ind(ErrorNb) enables the *Application* layer to inform the application of the occurrence of a fatal error identified by the number ErrorNb.

3.3.3 Application parameters

The Primary Station shall know the DES ciphering keys of all Secondary Stations for which a remote programming is to be carried out.

In case of remote programming, the Secondary Station shall know the DES ciphering key that could be used by the Primary Station.

3.3.4 Transitions d'état

Tableau 18 – Transitions d'état de *Application-62056-31*: Station Primaire

Etat initial	Condition de déclenchement	Ensemble d'actions	Etat final
Stopped	A_DATA.req(Com, ASDU) & (Com=ASO Com=ENQ Com=TRF)	APDU=concat(Com, _, _, ASDU) DL_DATA.req(APDU)	M.Rec
Stopped	A_DATA.req(Com, ASDU) & (Com=IB Com=TRB)	APDU=concat(Com, _, _, ASDU) DL_DATA.req(APDU)	W.EndS
Stopped	A_DATA.req(Com, ASDU) & Com=REC	Na1=randomize() Zdt=zdt(ASDU) APDU=concat(REC, Na1, 0, Zdt) DL_DATA.req(APDU) Na1k=cipher(Na1)	M.Rec
Stopped	DL_ALARM.ind()	A_ALARM.ind()	Stopped
Stopped	A_ABORT.req()	DL_ABORT.req()	Stopped
M.Rec	DL_DATA.ind(DSDU)	Resp=command(DSDU)	T.Resp
M.Rec	A_ABORT.req()	DL_ABORT.req()	Stopped
M.Rec	DL_ABORT.ind(ErrorNb) & Error_Nb<> EP_1 & Error_Nb<> EP_2	A_ABORT.ind(ErrorNb)	Stopped
M.Rec	DL_ABORT.ind(ErrorNb) & (Error_Nb = EP_1 Error_Nb = EP_2)	\$none()	M.Rec
W.EndS	DL_ABORT.ind(ErrorNb)	A_ABORT.ind(ErrorNb)	Stopped
W.EndS	A_ABORT.req()	DL_ABORT.req()	W.EndS
T.Resp	(Com=ASO & Resp=RSO) (Com=ENQ & Resp=DAT) (Com=ENQ & Resp=DRJ) (Com=TRF & Resp=TRA) (Com=TRF & Resp=DRJ) (Com=AUT & Resp=EOS) (Com=AUT & Resp=DRJ)	A_DATA.ind(DSDU)	Stopped
T.Resp	Com=REC & Resp=ECH & na1k(DSDU)=Na1k & Zdt=zdt(DSDU)	Na2=na2(DSDU) Na2k=cipher(Na2) Com=AUT APDU=concat(AUT, 0, Na2k, "") DL_DATA.req(APDU)	M.Rec
T.Resp	(Com=ASO & Resp<>RSO) (Com=ENQ & Resp<>DAT & Resp<>DRJ) (Com=TRF & Resp<>TRA & Resp<>DRJ) (Com=REC & Resp<>ECH) (Com=AUT & Resp<>ARJ & Resp<>DRJ & Resp<>EOS)	A_ABORT.ind(EA-1F) DL_ABORT.req()	Stopped
T.Resp	Com=REC & (Resp<>ECH (na1k(DSDU)<>Na1k) (Zdt<>zdt(DSDU)))	A_ABORT.ind(EA-2F) DL_ABORT.req()	Stopped
T.Resp	Com=AUT & Resp=ARJ	A_ABORT.ind(EA-3F) DL_ABORT.req()	Stopped

3.3.4 State transitions

Table 18 – Application-62056-31 state transitions: Primary Station

Initial state	Triggering condition	Set of actions	Final state
Stopped	A_DATA.req(Com, ASDU) & (Com=ASO Com=ENQ Com=TRF)	APDU=concat(Com, _, _, ASDU) DL_DATA.req(APDU)	M.Rec
Stopped	A_DATA.req(Com, ASDU) & (Com=IB Com=TRB)	APDU=concat(Com, _, _, ASDU) DL_DATA.req(APDU)	W.EndS
Stopped	A_DATA.req(Com, ASDU) & Com=REC	Na1=randomize() Zdt=zdt(ASDU) APDU=concat(REC, Na1, 0, Zdt) DL_DATA.req(APDU) Na1k=cipher(Na1)	M.Rec
Stopped	DL_ALARM.ind()	A_ALARM.ind()	Stopped
Stopped	A_ABORT.req()	DL_ABORT.req()	Stopped
M.Rec	DL_DATA.ind(DSDU)	Resp=command(DSDU)	T.Resp
M.Rec	A_ABORT.req()	DL_ABORT.req()	Stopped
M.Rec	DL_ABORT.ind(ErrorNb) & Error_Nb <> EP_1 & Error_Nb <> EP_2	A_ABORT.ind(ErrorNb)	Stopped
M.Rec	DL_ABORT.ind(ErrorNb) & (Error_Nb = EP_1 Error_Nb = EP_2)	\$none()	M.Rec
W.EndS	DL_ABORT.ind(ErrorNb)	A_ABORT.ind(ErrorNb)	Stopped
W.EndS	A_ABORT.req()	DL_ABORT.req()	W.EndS
T.Resp	(Com=ASO & Resp=RSO) (Com=ENQ & Resp=DAT) (Com=ENQ & Resp=DRJ) (Com=TRF & Resp=TRA) (Com=TRF & Resp=DRJ) (Com=AUT & Resp=EOS) (Com=AUT & Resp=DRJ)	A_DATA.ind(DSDU)	Stopped
T.Resp	Com=REC & Resp=ECH & na1k(DSDU)=Na1k & Zdt=zdt(DSDU)	Na2=na2(DSDU) Na2k=cipher(Na2) Com=AUT APDU=concat(AUT, 0, Na2k, "") DL_DATA.req(APDU)	M.Rec
T.Resp	(Com=ASO & Resp<>RSO) (Com=ENQ & Resp<>DAT & Resp<>DRJ) (Com=TRF & Resp<>TRA & Resp<>DRJ) (Com=REC & Resp<>ECH) (Com=AUT & Resp<>ARJ & Resp<>DRJ & Resp<>EOS)	A_ABORT.ind(EA-1F) DL_ABORT.req()	Stopped
T.Resp	Com=REC & (Resp<>ECH (na1k(DSDU)<>Na1k) (Zdt<>zdt(DSDU)))	A_ABORT.ind(EA-2F) DL_ABORT.req()	Stopped
T.Resp	Com=AUT & Resp=ARJ	A_ABORT.ind(EA-3F) DL_ABORT.req()	Stopped

Tableau 19 – Transitions d'état de *Application-62056-31*: Station Secondaire

Etat initial	Condition de déclenchement	Ensemble d'actions	Etat final
Stopped	DL_DATA.ind(DSDU) & (command(DSDU)=ENQ Command(DSDU)=TRF)	A_DATA.ind(DSDU) Req=command(DSDU)	M.Send
Stopped	DL_DATA.ind(DSDU) & command(DSDU)=TRB	A_DATA.ind(DSDU)	Stopped
Stopped	DL_DATA.ind(DSDU) & command(DSDU)=REC	Zdt=zdt(DSDU) Na1=na1(DSDU) Na1k=cipher(Na1) Na2=randomize() APDU=concat(ECH, Na1k, Na2, Zdt) DL_DATA.req(APDU) Na2k=cipher(Na2) Req=REC	W.AUT
Stopped	A_ALARM.req() & alarm_detection()	DL_ALARM.req()	Stopped
M.Send	A_DATA.req(COM, ASDU) & (((COM=DAT COM=DRJ) & Req=ENQ) ((COM=TRA COM=DRJ) & Req=TRF) (COM=DRJ & Req=REC))	APDU=concat(COM, _, _, ASDU) DL_DATA.req(APDU)	Stopped
M.Send	A_DATA.req(COM, ASDU) & (COM=EOS & Req=REC)	APDU=concat(EOS, 0, 0, "") DL_DATA.req(APDU)	Stopped
M.Send	DL_ABORT.ind(ErrorNb)	A_ABORT.ind(ErrorNb)	Stopped
W.AUT	DL_DATA.ind(DSDU) & Command(DSDU)=AUT & na2k(DSDU)=Na2k	ASDU=concat(REC, _, _, Zdt) A_DATA.ind(ASDU)	M.Send
W.AUT	DL_DATA.ind(DSDU) & Command(DSDU)=AUT & na2k(DSDU)<>Na2k	APDU=concat(ARJ, _, _, "") DL_DATA.req(APDU)	Stopped
W.AUT	DL_ABORT.ind(ErrorNb)	A_ABORT.ind(ErrorNb)	Stopped

Tableau 20 – Signification des états mentionnés dans les tableaux précédents

Etat	Signification
Stopped	Attente de la première requête de la couche supérieure ou de la première indication de la couche inférieure
M.Rec (Must Receive)	Attente de la réponse à la requête émise
T.Resp (Test Response)	Traitement de la réponse reçue
M.Send (Must Send)	Attente d'une réponse à une requête reçue
W.AUT (Wait for AUT frame)	Attente de la trame AUT consécutive à une réponse ECH envoyée

Table 19 – Application-62056-31 state transitions: Secondary Station

Initial state	Triggering condition	Set of actions	Final state
Stopped	DL_DATA.ind(DSDU) & (command(DSDU)=ENQ command(DSDU)=TRF)	A_DATA.ind(DSDU) Req=command(DSDU)	M.Send
Stopped	DL_DATA.ind(DSDU) & command(DSDU)=TRB	A_DATA.ind(DSDU)	Stopped
Stopped	DL_DATA.ind(DSDU) & command(DSDU)=REC	Zdt=zdt(DSDU) Na1=na1(DSDU) Na1k=cipher(Na1) Na2=randomize() APDU=concat(ECH, Na1k, Na2, Zdt) DL_DATA.req(APDU) Na2k=cipher(Na2) Req=REC	W.AUT
Stopped	A_ALARM.req() & alarm_detection()	DL_ALARM.req()	Stopped
M.Send	A_DATA.req(COM, ASDU) & (((COM=DAT COM=DRJ) & Req=ENQ) ((COM=TRA COM=DRJ) & Req=TRF) (COM=DRJ & Req=REC))	APDU=concat(COM, _, ASDU) DL_DATA.req(APDU)	Stopped
M.Send	A_DATA.req(COM, ASDU) & (COM=EOS & Req=REC)	APDU=concat(EOS, 0, 0, "") DL_DATA.req(APDU)	Stopped
M.Send	DL_ABORT.ind(ErrorNb)	A_ABORT.ind(ErrorNb)	Stopped
W.AUT	DL_DATA.ind(DSDU) & command(DSDU)=AUT & na2k(DSDU)=Na2k	ASDU=concat(REC, _, _, Zdt) A_DATA.ind(ASDU)	M.Send
W.AUT	DL_DATA.ind(DSDU) & command(DSDU)=AUT & na2k(DSDU) <> Na2k	APDU=concat(ARJ, _, _, "") DL_DATA.req(APDU)	Stopped
W.AUT	DL_ABORT.ind(ErrorNb)	A_ABORT.ind(ErrorNb)	Stopped

Table 20 – Meaning of the states listed in the previous tables

State	Meaning
Stopped	Waiting state for the first request from the upper layer or the first indication from the lower layer
M.Rec (Must Receive)	Standby for the response to the request transmitted
T.Resp (Test Response)	Processing of the response received
M.Send (Must Send)	Waiting for a response to a received request
W.AUT (Wait for AUT frame)	Waiting for an AUT frame following an ECH response frame sent

Tableau 21 – Définition des procédures et des fonctions classées dans l'ordre alphabétique

Procédure ou fonction	Définition
alarm_detection()	Vérification que le mode Alarme est Actif
cipher(Na1) or cipher(Na2)	Cryptage du nombre aléatoire Na1 ou Na2 N par l'algorithme DES avec la clef du contexte de communication
command(DSDU)	Extraction du champ commande d'un DSDU reçu
concat(COM, _, _, ASDU), concat(COM, ZA1, ZA2, ZDT) or concat(COM, 0, 0, _)	Concaténation d'un champ commande COM et d'un ASDU ou concaténation d'un champ commande COM, d'une valeur cryptée ZA1, d'une valeur cryptée ZA2 et d'une zone de données ZDT (champs TAB et DATA) ou concaténation d'un champ commande COM et des champs ZA1 = 0 et ZA2 = 0
na1(DSDU)	Extraction de la valeur Na1 du champ ZA1 d'une trame REC reçue
na1k(DSDU)	Extraction de la valeur Na1k du champ ZA1 d'une trame ECH reçue
na2(DSDU)	Extraction de la valeur Na2 du champ ZA2 d'une trame ECH reçue
na2k(DSDU)	Extraction de la valeur Na2k du champ ZA2 d'une trame AUT reçue
randomize()	Génération d'un nombre aléatoire suivant la procédure définie à l'annexe G
zdt(ASDU) or zdt(DSDU)	Extraction de données (champs TAB et DATA) d'une demande de type REC, d'une trame REC ou d'une trame ECH

**Table 21 – Definition of the procedures and functions
classified in alphabetical order**

Procedure or function	Definition
alarm_detection()	Check that the status of Alarm mode is Active
cipher(Na1) or cipher(Na2)	Ciphering of random number Na1 or Na2 by means of DES algorithm with key contained in the communication context
command(DSDU)	Extraction of the value of the command code of a received DSDU
concat(COM, _, _, ASDU), concat(COM, ZA1, ZA2, ZDT) or concat(COM, 0, 0, _)	Concatenation of a COM command code and an ASDU or concatenation of a COM command code with an encrypted value ZA1, an encrypted value ZA2 and a data field ZDT (TAB and DATA fields) or concatenation of a COM command code and the ZA1 = 0 and ZA2 = 0 fields
na1(DSDU)	Extraction of the Na1 value from the ZA1 field of a received REC frame
na1k(DSDU)	Extraction of the Na1k value from the ZA1 field of a received ECH frame
na2(DSDU)	Extraction of the Na2 value from the ZA2 field of a received ECH frame
na2k(DSDU)	Extraction of the Na2k value from the ZA2 field of a received AUT frame
randomize()	Generation of a random number according to the procedure described in annex G
zdt(ASDU) or zdt(DSDU)	Extraction of data (TAB and DATA fields) from a REC request, a REC frame or an ECH frame

3.3.5 Répertoire et traitement des erreurs

Les erreurs sont répertoriées par le codage suivant:

EA = erreur de la couche *Application*

– = séparateur

N = numéro de l'erreur

F = erreur fatale

Tableau 22 – Tableau récapitulatif des erreurs

EA-1F	Le code commande de la réponse reçue ne correspond pas à la requête émise (Station Primaire seulement)
	Cette erreur conduit à réinitialiser la couche <i>Application</i> après avoir informé l'application
EA-2F	Erreur d'authentification détectée sur la réponse de la Station Secondaire (Station Primaire seulement)
	Cette erreur conduit à réinitialiser la couche <i>Application</i> après avoir informé l'application
EA-3F	Erreur d'authentification détectée par la Station Secondaire (Station Primaire seulement)
	Cette erreur conduit à réinitialiser la couche <i>Application</i> après avoir informé l'application

Toute occurrence de l'une de ces erreurs fatales est remontée localement grâce à la primitive de service A_ABORT.ind. La liste complète des numéros d'erreur fatale est fournie à l'annexe C.

4 Echange de données par bus local avec DLMS

4.1 Couche Physique

Le protocole *Physique-62056-31* de la couche *Physique* de l'architecture d'échange de données par bus local avec DLMS est exactement le même que celui défini pour l'architecture sans DLMS.

4.2 Couche Liaison

4.2.1 Protocole Liaison-E/D

Le protocole *Liaison-E/D* de la couche *Liaison* de l'architecture d'échange de données par bus local avec DLMS adopte un comportement asymétrique. La machine d'état de la Station Primaire est donc différente de celle de la Station Secondaire.

La couche *Liaison* est chargée de transformer le canal physique exploité par la couche *Physique* en canal logique apte à transmettre de l'information fiable. Ses fonctions principales sont:

- gérer directement les services d'Initialisation de Bus et de Réponse des Stations Oubliées;
- effectuer la sérialisation et la désérialisation des données (dans la mesure où le canal physique fonctionne en série par bit);
- synchroniser les trames en émission et en réception;
- filtrer les trames en fonction des adresses primaire et secondaire;
- assurer une protection efficace contre les erreurs de transmission.

3.3.5 List and processing of errors

Errors are listed using the following codes:

EA = error in the *Application* layer

– = separator

N = error number

F = fatal error

Table 22 – Error summary table

EA-1F	The response command code of the received frame does not correspond with the request command code (Primary Station only)
	This error leads to the reinitialization of the <i>Application</i> layer after having informed the application
EA-2F	The Secondary Station has not been correctly authenticated (Primary Station only)
	This error leads to the reinitialization of the <i>Application</i> layer after having informed the application
EA-3F	The Primary Station has not been correctly authenticated (Primary Station only)
	This error leads to the reinitialization of the <i>Application</i> layer after having informed the application

If any of these errors occurs, it is sent up locally by means of the A_ABORT.ind service primitive. The complete list of fatal error numbers is given in annex C.

4 Local bus data exchange with DLMS

4.1 Physical layer

The *Physical*-62056-31 protocol of the *Physical* layer of the local bus data exchange architecture with DLMS is exactly the same as the one defined for the architecture without DLMS.

4.2 Data Link layer

4.2.1 Link-E/D protocol

The *Link-E/D* protocol of the *Data Link* layer of the local bus data exchange architecture with DLMS behaves asymmetrically. The state machine of the Primary Station is therefore different from that of the Secondary Station.

The *Data Link* layer transforms the physical channel used by the *Physical* layer into a logic channel able to transmit reliable information. Its main functions are:

- to manage directly the Bus Initialization and Forgotten Stations Call services;
- to carry out serialization and deserialization of the data (if the physical channel functions serially one bit at a time);
- to synchronize the transmission and reception of the frames;
- to filter the frames according to primary and secondary addresses;
- to ensure efficient protection against transmission errors.

4.2.2 Gestion des échanges

Les services d'Initialisation de Bus, Alarme et Appel des Stations Oubliées sont offerts par le protocole *Liaison-E/D* de la couche *Liaison* de l'architecture d'échanges de données par bus local avec DLMS, mais les opérations sont réalisées en dehors de la couche *Application*. En particulier, le drapeau de station oubliée peut être mis à jour par l'interface de programmation de l'application alors qu'une télérelève a lieu.

Sauf pour l'ouverture d'une session, durant l'Initialisation de Bus, la remontée d'Alarme et la gestion de l'Appel des Stations Oubliées, le protocole est complètement symétrique, et les deux stations sont à tour de rôle Emetteur et Récepteur.

Après l'envoi d'une trame, la couche *Liaison* du côté Emetteur attend toujours une trame de la couche *Liaison* du Récepteur avant d'émettre à nouveau. Cette attente est contrôlée par le délai de garde T1, d'une valeur de 10 s.

Après l'envoi d'une trame et la réception de l'acquiescement de l'envoi précédent, la trame courante est retransmise. Le nombre de retransmissions est limité à MaxRetry. Au-delà de ce nombre, la communication est interrompue au niveau *Liaison* et la couche *Application* en est informée.

Chaque fois qu'une trame est reçue, il y a émission immédiate d'une trame en réponse. Cette trame peut contenir des données de la couche *Application*. Elle contient toujours un numéro dans le champ *Send* et un numéro dans le champ *Confirm*, calculés en fonction des valeurs précédemment envoyées et reçues. L'algorithme de calcul de ces numéros est le suivant:

- si la dernière trame reçue est sans erreur et que son numéro *Send* est égal au complément à 1 du numéro *Confirm* précédent en émission, alors le paquet de données est transmis à la couche *Application* et la prochaine trame envoyée aura un numéro *Confirm* égal au numéro *Send* reçu. Sinon le numéro *Confirm* reste inchangé et le paquet de données n'est pas transmis à la couche *Application*;
- si la dernière trame reçue est sans erreur et que son numéro *Confirm* est identique au numéro *Send* précédent en émission, alors le numéro *Send* en émission est complété à 1 pour la prochaine trame dans l'hypothèse où un nouveau paquet de données doit être envoyé;
- si la dernière trame reçue est incorrecte ou si son numéro *Confirm* n'est pas identique au numéro *Send* précédent en émission, alors la même trame est de nouveau transmise sous réserve que le nombre de réémissions reste inférieur ou égal à MaxRetry.

4.2.3 Services et primitives de service de liaison

L'utilisateur du protocole *Liaison-E/D* dispose des services et primitives de service donnés dans le tableau 23.

Tableau 23 – Services et primitives de services de liaison

Service	Primitive de service
DL_DATA	DL_DATA.req(Pr, DSDU) DL_DATA.ind(Pr, DSDU)
DL_IB	DL_IB.req()
DL_ASO	DL_ASO.req(DSDU) DL_ASO.ind(Collision, List)
DL_ALARM	DL_ALARM.req()
DL_ABORT	DL_ABORT.req(Strong) DL_ABORT.ind(ErrorNb)

4.2.2 Management of exchanges

Bus Initialization, Alarm and Forgotten Stations Call services are provided by the *Link E/D* protocol of the *Data Link* layer of the local bus data exchange architecture with DLMS, but their operation takes place outside the *Application* layer. In particular, the forgotten station flag can be updated by the Application Programming Interface when a remote reading exchange occurs.

Except during opening of the communication session and Bus Initialization, Alarm report or management of Forgotten Stations Call, the protocol is perfectly symmetrical, and both stations take turns to act as Transmitter and Receiver.

After sending a frame, the *Data Link* layer of the Transmitter end always waits for a frame from the *Data Link* layer of the Receiver before transmitting again. This wait is controlled by the T1 wakeup, whose duration is 10 s.

After sending a frame and receiving an acknowledgement of the previously sent one, the current frame is retransmitted. The number of repeat transmissions is limited to MaxRetry. Above this number, the communication is stopped at the *Data Link* level and the *Application* layer is informed.

Each time a frame is received by one of the systems, an answer frame is immediately transmitted. This transmitted frame may contain data from the *Application* layer. It always contains a *Send* number and a *Confirm* number calculated according to the values of those previously sent and received. The following algorithm is used to calculate these numbers:

- if the last frame received is error-free and its *Send* number is equal to the 1's complement of the previous *Confirm* number in transmission, then the data packet is transmitted to the *Application* layer and the next frame sent will have a *Confirm* number equal to the received *Send* number. If not, the *Confirm* number is not modified and the data packet is not transmitted to the *Application* layer;
- if the last frame received is error-free and its *Confirm* number is identical to the previous *Send* number in transmission, then the *Send* number in transmission is made up to 1 for the next frame on the assumption that a new data packet has to be sent;
- if the last frame received is incorrect or if its *Confirm* number is not identical to the previous *Send* number in transmission, then the same frame is transmitted again, on condition that the number of repeat transmissions remains less than or equal to MaxRetry.

4.2.3 Data Link services and service primitives

The user of the *Link-E/D* protocol can use the services and service primitives given in the table 23.

Table 23 – Data Link services and service primitives

Service	Service primitive
DL_DATA	DL_DATA.req(Pr, DSDU) DL_DATA.ind(Pr, DSDU)
DL_IB	DL_IB.req()
DL_ASO	DL_ASO.req(DSDU) DL_ASO.ind(Collision, List)
DL_ALARM	DL_ALARM.req()
DL_ABORT	DL_ABORT.req(Strong) DL_ABORT.ind(ErrorNb)

Le rôle attribué à chaque primitive est le suivant:

- DL_DATA.req(Pr, DSDU) permet à la couche *Application* de demander à la couche *Liaison* le transfert d'un paquet de données DSDU avec la priorité Pr ¹⁾;
- DL_DATA.ind(Pr, DSDU) permet à la couche *Liaison* d'informer la couche *Application* de l'arrivée d'un paquet de données DSDU avec la priorité Pr;
- DL_IB.req() permet à l'*interface de programmation de l'Application* de demander à la couche *Liaison* l'envoi d'une trame Initialisation du Bus;
- DL_ASO.req(DSDU) permet à l'*interface de programmation de l'Application* de demander à la couche *Liaison* l'envoi d'une trame Appel des Stations Oubliées en cohérence avec un paquet de données DSDU;
- DL_ASO.ind(Collision, List) permet à la couche *Liaison* d'informer l'*interface de programmation de l'Application* du résultat d'un Appel des Stations Oubliées;
- DL_ALARM.req() permet à l'*interface de programmation de l'Application* de demander à la couche *Liaison* l'envoi d'une alarme;
- DL_ABORT.req(Strong) permet à la couche *Application* de demander à la couche *Liaison* de mettre fin à son activité avec la priorité Strong ²⁾;
- DL_ABORT.ind(ErrorNb) permet à la couche *Liaison* d'informer la couche *Application* de l'occurrence d'une erreur fatale repérée par le numéro ErrorNb.

4.2.4 Paramètres de liaison

La valeur du nombre MaxRetry de réémissions d'une même trame avant déconnexion est fixée à 2.

Pour une Station Primaire, la valeur du nombre maximal MaxRSO d'intervalles de temps RSO pour le traitement d'une trame «Appel des Stations Oubliées» est fixée à 3.

La Station Secondaire doit connaître la liste des adresses de Stations Primaires et la liste des TABi auxquelles elle a été sensibilisée.

Une station peut aussi être sollicitée par l'intermédiaire d'une adresse primaire générale APG. Dans ce cas, elle répond avec la première adresse de la liste des adresses de Stations Primaires auxquelles elle a été sensibilisée.

¹⁾ Le paramètre de priorité Pr différencie le traitement d'un service urgent tel que InformationReport (niveau Pr=1) d'un autre service DLMS (niveau Pr=0)

²⁾ Le paramètre de priorité Strong différencie le traitement des erreurs fatales (Strong=1) de celui d'une autre demande de déconnexion physique (Strong=0) initialisée par la sous-couche *Application*

The role assigned to each primitive is as follows:

- DL_DATA.req(Pr, DSDU) enables the *Application* layer to request the *Data Link* layer to transfer a DSDU data packet with the priority Pr ¹⁾;
- DL_DATA.ind(Pr, DSDU) enables the *Data Link* layer to inform the *Application* layer of the arrival of a DSDU data packet with the priority Pr;
- DL_IB.req() enables the *Application Programming Interface* to request the *Data Link* layer to send a Bus Initialization frame;
- DL_ASO.req(DSDU) enables the *Application Programming Interface* to request the *Data Link* layer to send a Forgotten Stations Call frame in accordance with a DSDU data packet;
- DL_ASO.ind(Collision, List) enables the *Data Link* layer to inform the *Application Programming Interface* of the result of a Forgotten Stations Call;
- DL_ALARM.req() enables the *Application Programming Interface* to request the *Data Link* layer to transfer an alarm;
- DL_ABORT.req(Strong) enables the *Application* layer to request the *Data Link* layer to terminate its activity with the priority Strong ²⁾;
- DL_ABORT.ind(ErrorNb) enables the *Data Link* layer to inform the *Application* layer of the occurrence of a fatal error identified by the number ErrorNb.

4.2.4 Data Link parameters

The value of the number of repeat transmissions for a given frame before disconnection, MaxRetry, is set to 2.

For the Primary Station, the value of the maximum number, MaxRSO, of RSO time slots for the processing of a "Forgotten Stations Call" is set to 3.

The Secondary Station shall know the list of Primary Station addresses and the list of TABi to which it has been programmed.

The station may also be solicited by the general primary address APG. In this case, it replies with the first primary address to which it has been programmed.

1) The priority level Pr differentiates the processing of emergency services such as InformationReport (level Pr=1) from that of the other DLMS services (level Pr=0).

2) The Strong parameter differentiates the processing of fatal errors (Strong=1) from that of the other physical disconnection requests (Strong=0) initialized by the *Application* sub-layer.

4.2.5 Transitions d'état

Tableau 24 – Transitions d'état de *Liaison-E/D*: Station Primaire

Etat initial	Condition de déclenchement	Ensemble d'actions	Etat final
<i>Initial</i>	\$true()	MaxRetry=2 MaxChain = 5 init_incrChain()	Stopped
Stopped	exist_dl_req()	NbChain = 0 RepeatASO=FALSE context(ADP, ADS, TypeAG) init(TypeAG) Phy_APPG.req(TypeAG)	W.AG
Stopped	Phy_ABORT.ind(ErrorNb)	DL_ABORT.ind(ErrorNb)	Stopped
Stopped	Phy_ALARM.ind()	create_alarm(TPDU) DL_DATA.ind(Pr = 1, TPDU)	Stopped
W.AG	Phy_APPG.ind() & not(RepeatASO) & NbChain = 0	\$none()	T.Reg
W.AG	Phy_APPG.ind() & not(RepeatASO) & NbChain <> 0	NbChain = 0	M.Send
W.AG	Phy_APPG.ind() & RepeatASO	RepeatASO=FALSE Phy_ASO.req(Fr)	M.RSO
W.AG	DL_ABORT.req(_)	Phy_ABORT.req()	W.EndS
W.EndS	(Phy_ABORT.ind(EP-2) & TypeAG=AGT) I (Phy_ABORT.ind(EP-1) & TypeAG=AGN)	\$none()	Stopped
W.EndS	Phy_ALARM.ind	create_alarm(TPDU) DL_DATA.ind(Pr = 1, TPDU)	Stopped
W.EndS	Phy_ABORT.ind(ErrorNb) & ErrorNb<>EP-1 & ErrorNb<>EP-2	DL_ABORT.ind(ErrorNb)	W.EndS
T.Reg	exist_dl_req(DL_IB.req())	Fr="" Size=size_frame(Fr) Fr=concat(Size, ADS, ADP, IB, Fr) Fr=concat(Fr, crc(Fr)) Phy_UNACK.req(Fr)	W.EndS
T.Reg	exist_dl_req(DL_ASO.req(DSDU)) & TypeAG=AGN	MaxRSO=3 NbRSO=1 ListRSO="" Collision=FALSE Fr=DSDU Size=size_frame(Fr) Fr=concat(Size, ADS, ADP, ASO, Fr) Fr=concat(Fr, crc(Fr)) Phy_ASO.req(Fr)	M.RSO
T.Reg	exist_dl_req(DL_ASO.req(DSDU)) & TypeAG=AGT	MaxRSO=1 NbRSO=1 ListRSO="" Collision=FALSE Fr=DSDU Size=size_frame(Fr) Fr=concat(Size, ADS, ADP, ASO, Fr) Fr=concat(Fr, crc(Fr)) Phy_ASO.req(Fr)	M.RSO

4.2.5 State transitions

Table 24 – Link-E/D state transitions: Primary Station

Initial state	Triggering condition	Set of actions	Final state
<i>Initial</i>	\$true()	MaxRetry=2 MaxChain = 5 init_incrChain()	Stopped
Stopped	exist_dl_req()	NbChain = 0 RepeatASO=FALSE context(ADP, ADS, TypeAG) init(TypeAG) Phy_APPG.req(TypeAG)	W.AG
Stopped	Phy_ABORT.ind(ErrorNb)	DL_ABORT.ind(ErrorNb)	Stopped
Stopped	Phy_ALARM.ind()	create_alarm(TPDU) DL_DATA.ind(Pr = 1, TPDU)	Stopped
W.AG	Phy_APPG.ind() & not(RepeatASO) & NbChain = 0	\$none()	T.Reg
W.AG	Phy_APPG.ind() & not(RepeatASO) & NbChain <> 0	NbChain = 0	M.Send
W.AG	Phy_APPG.ind() & RepeatASO	RepeatASO=FALSE Phy_ASO.req(Fr)	M.RSO
W.AG	DL_ABORT.req(_)	Phy_ABORT.req()	W.EndS
W.EndS	(Phy_ABORT.ind(EP-2) & TypeAG=AGT) I (Phy_ABORT.ind(EP-1) & TypeAG=AGN)	\$none()	Stopped
W.EndS	Phy_ALARM.ind	create_alarm(TPDU) DL_DATA.ind(Pr = 1, TPDU)	Stopped
W.EndS	Phy_ABORT.ind(ErrorNb) & ErrorNb<>EP-1 & ErrorNb<>EP-2	DL_ABORT.ind(ErrorNb)	W.EndS
T.Reg	exist_dl_req(DL_IB.req())	Fr="" Size=size_frame(Fr) Fr=concat(Size, ADS, ADP, IB, Fr) Fr=concat(Fr, crc(Fr)) Phy_UNACK.req(Fr)	W.EndS
T.Reg	exist_dl_req(DL_ASO.req(DSDU)) & TypeAG=AGN	MaxRSO=3 NbRSO=1 ListRSO="" Collision=FALSE Fr=DSDU Size=size_frame(Fr) Fr=concat(Size, ADS, ADP, ASO, Fr) Fr=concat(Fr, crc(Fr)) Phy_ASO.req(Fr)	M.RSO
T.Reg	exist_dl_req(DL_ASO.req(DSDU)) & TypeAG=AGT	MaxRSO=1 NbRSO=1 ListRSO="" Collision=FALSE Fr=DSDU Size=size_frame(Fr) Fr=concat(Size, ADS, ADP, ASO, Fr) Fr=concat(Fr, crc(Fr)) Phy_ASO.req(Fr)	M.RSO

<i>T.Req</i>	not (exist_dl_req(DL_IB.req()) exist_dl_req(DL_ASO.req(_)))	Pr=0 Send="00"B Confirm="11"B Fr="" Index=Index + 1 NbChain = NbChain + IncrChain Size=size_frame(Fr) Com=com(DATA+,Pr,Send,Confirm) Fr=concat(Size, ADS, ADP, Com, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr)	M.Rec
M.RSO	Phy_ASO.ind(Frame) & size(Frame)=0	\$none()	<i>T.RSO</i>
M.RSO	Phy_ASO.ind(Frame) & check_frame(Frame) & command(Frame)=RSO	build_RSO(ListRSO, Frame)	<i>T.RSO</i>
M.RSO	Phy_ASO.ind(Frame) & not(check_frame(Frame)) & size(Frame)<>0	Collision=TRUE	<i>T.RSO</i>
M.RSO	Phy_COLL.ind()	Collision=TRUE	<i>T.RSO</i>
M.RSO	DL_ABORT.req(_)	Phy_ABORT.req()	W.EndS
M.RSO	Phy_ABORT.ind(ErrorNb)	DL_ABORT.ind(ErrorNb)	W.EndS
<i>T.RSO</i>	MaxRSO=1 & Collision	MaxRSO=3 Collision=FALSE RepeatASO=TRUE Phy_APPG.req(AGN)	W.AG
<i>T.RSO</i>	(MaxRSO=1 & not(Collision)) (MaxRSO<>1 & NbRSO>=MaxRSO)	DL_ASO.ind(Collision, ListRSO)	W.EndS
<i>T.RSO</i>	NbRSO<MaxRSO	NbRSO=NbRSO+1	M.RSO
<i>M.Send</i>	exist_dl_data_req(DL_DATA.req(Pr=1, DSDU)) & NbChain < MaxChain	Send=incr(Send) Ack_expected=TRUE Fr=DSDU Index= Index + 1 NbChain = NbChain + IncrChain Size=size_frame(Fr) Com=com(DATA+,Pr,Send,Confirm) Fr=concat(Size, ADS, ADP, Com, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr)	M.Rec
<i>M.Send</i>	not(DL_DATA.req(Pr=1, _)) & exist_dl_data_req(DL_DATA.req(Pr=0, DSDU)) & NbChain < MaxChain	Send=incr(Send) Ack_expected=TRUE Fr=DSDU Index= Index + 1 NbChain = NbChain + IncrChain Size=size_frame(Fr) Com=com(DATA+,Pr,Send,Confirm) Fr=concat(Size, ADS, ADP, Com, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr)	M.Rec
<i>M.Send</i>	not(DL_DATA.req(_, _)) & NbChain < MaxChain	Pr=0 Fr="" Index= Index + 1 NbChain = NbChain + IncrChain Size=size_frame(Fr) Com=com(DATA+,Pr,Send,Confirm) Fr=concat(Size, ADS, ADP, Com, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr)	M.Rec

<i>T.Req</i>	not (exist_dl_req(DL_IB.req()) exist_dl_req(DL_ASO.req(_)))	Pr=0 Send="00"B Confirm="11"B Fr="" Index=Index + 1 NbChain = NbChain + IncrChain Size=size_frame(Fr) Com=com(DATA+,Pr,Send,Confirm) Fr=concat(Size, ADS, ADP, Com, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr)	M.Rec
M.RSO	Phy_ASO.ind(Frame) & size(Frame)=0	\$none()	<i>T.RSO</i>
M.RSO	Phy_ASO.ind(Frame) & check_frame(Frame) & command(Frame)=RSO	build_RSO(ListRSO, Frame)	<i>T.RSO</i>
M.RSO	Phy_ASO.ind(Frame) & not(check_frame(Frame)) & size(Frame)<>0	Collision=TRUE	<i>T.RSO</i>
M.RSO	Phy_COLL.ind()	Collision=TRUE	<i>T.RSO</i>
M.RSO	DL_ABORT.req(_)	Phy_ABORT.req()	W.EndS
M.RSO	Phy_ABORT.ind(ErrorNb)	DL_ABORT.ind(ErrorNb)	W.EndS
<i>T.RSO</i>	MaxRSO=1 & Collision	MaxRSO=3 Collision=FALSE RepeatASO=TRUE Phy_APPG.req(AGN)	W.AG
<i>T.RSO</i>	(MaxRSO=1 & not(Collision)) (MaxRSO<>1 & NbRSO>=MaxRSO)	DL_ASO.ind(Collision, ListRSO)	W.EndS
<i>T.RSO</i>	NbRSO<MaxRSO	NbRSO=NbRSO+1	M.RSO
<i>M.Send</i>	exist_dl_data_req(DL_DATA.req(Pr=1, DSDU)) & NbChain < MaxChain	Send=incr(Send) Ack_expected=TRUE Fr=DSDU Index= Index + 1 NbChain = NbChain + IncrChain Size=size_frame(Fr) Com=com(DATA+,Pr,Send,Confirm) Fr=concat(Size, ADS, ADP, Com, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr)	M.Rec
<i>M.Send</i>	not(DL_DATA.req(Pr=1, _)) & exist_dl_data_req(DL_DATA.req(Pr=0, DSDU)) & NbChain < MaxChain	Send=incr(Send) Ack_expected=TRUE Fr=DSDU Index= Index + 1 NbChain = NbChain + IncrChain Size=size_frame(Fr) Com=com(DATA+,Pr,Send,Confirm) Fr=concat(Size, ADS, ADP, Com, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr)	M.Rec
<i>M.Send</i>	not(DL_DATA.req(_, _)) & NbChain < MaxChain	Pr=0 Fr="" Index= Index + 1 NbChain = NbChain + IncrChain Size=size_frame(Fr) Com=com(DATA+,Pr,Send,Confirm) Fr=concat(Size, ADS, ADP, Com, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr)	M.Rec

<i>M.Send</i>	NbChain >= MaxChain	Phy_APPG.req(AGN)	W.AG
M.Rec	Phy_DATA.ind(Frame) & (check_frame(Frame) & check_address(Frame) & is_data+(Frame) & is_ack(Frame)) & is_text(Frame)	DL_DATA.ind(extract_prt(Frame), extract_text(Frame)) Confirm=incr(Confirm) Ack_expected=FALSE Index = 0	<i>M.Send</i>
M.Rec	Phy_DATA.ind(Frame) & (check_frame(Frame) & check_address(Frame) & is_data+(Frame) & is_ack(Frame)) & not(is_text(Frame))	Ack_expected=FALSE Index = 0	<i>M.Send</i>
M.Rec	Phy_DATA.ind(Frame) & not(check_frame(Frame) & check_address(Frame) & is_data+(Frame) & is_ack(Frame)) & Index<=MaxRetry	Phy_DATA.req(Fr) Index=Index+1	M.Rec
M.Rec	Phy_DATA.ind(Frame) & not(check_frame(Frame) & check_address(Frame) & is_data+(Frame) & is_ack(Frame)) & Index>MaxRetry	DL_ABORT.ind(EL-2F) Phy_ABORT.req()	W.EndS
M.Rec	DL_ABORT.req(Strong=0) & not(DL_DATA.req(,)) & Ack_expected=FALSE	Phy_ABORT.req()	W.EndS
M.Rec	DL_ABORT.req(Strong=1)	Phy_ABORT.req()	W.EndS
M.Rec	Phy_ABORT.ind(ErrorNb)	DL_ABORT.ind(ErrorNb)	W.EndS

<i>M.Send</i>	NbChain >= MaxChain	Phy_APPG.req(AGN)	W.AG
M.Rec	Phy_DATA.ind(Frame) & (check_frame(Frame) & check_address(Frame) & is_data+(Frame) & is_ack(Frame)) & is_text(Frame)	DL_DATA.ind(extract_prt(Frame), extract_text(Frame)) Confirm=incr(Confirm) Ack_expected=FALSE Index = 0	<i>M.Send</i>
M.Rec	Phy_DATA.ind(Frame) & (check_frame(Frame) & check_address(Frame) & is_data+(Frame) & is_ack(Frame)) & not(is_text(Frame))	Ack_expected=FALSE Index = 0	<i>M.Send</i>
M.Rec	Phy_DATA.ind(Frame) & not(check_frame(Frame) & check_address(Frame) & is_data+(Frame) & is_ack(Frame)) & Index<=MaxRetry	Phy_DATA.req(Fr) Index=Index+1	M.Rec
M.Rec	Phy_DATA.ind(Frame) & not(check_frame(Frame) & check_address(Frame) & is_data+(Frame) & is_ack(Frame)) & Index>MaxRetry	DL_ABORT.ind(EL-2F) Phy_ABORT.req()	W.EndS
M.Rec	DL_ABORT.req(Strong=0) & not(DL_DATA.req(_, _)) & Ack_expected=FALSE	Phy_ABORT.req()	W.EndS
M.Rec	DL_ABORT.req(Strong=1)	Phy_ABORT.req()	W.EndS
M.Rec	Phy_ABORT.ind(ErrorNb)	DL_ABORT.ind(ErrorNb)	W.EndS

Tableau 25 – Transitions d'état de *Liaison-E/D*: Station Secondaire

Etat initial	Condition de déclenchement	Ensemble d'actions	Etat final
<i>Initial</i>	\$true()	MaxRetry=2 FlagDSO=TRUE Discovered=FALSE Flag_alarm = FALSE	Stopped
Stopped	Phy_DATA.ind(Frame) & check_frame(Frame) & check_address(Frame)	ADP=extract_ADP(Frame) Com=command(Frame)	<i>T.Com</i>
Stopped	Phy_DATA.ind(Frame) & check_frame(Frame) & not(check_address(Frame))	Phy_ABORT.req()	Stopped
Stopped	Phy_DATA.ind(Frame) & not(check_frame(Frame))	\$none()	Stopped
Stopped	DL_ALARM.req() & alarm_detection()	Phy_ABORT.req() Flag_alarm = TRUE Phy_ALARM.req()	Stopped
<i>T.Com</i>	Com=IB	FlagDSO=TRUE Discovered=FALSE Phy_ABORT.req()	Stopped
<i>T.Com</i>	Com=ASO & test_TABi(Frame, TAB)	Fr=concat(RSO, TAB, ADS) Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Phy_RSO.req(Fr_window_RSO())	Stopped
<i>T.Com</i>	Com=ASO & not(test_TABi(Frame, TAB))	Phy_ABORT.req()	Stopped
<i>T.Com</i>	is_data+(Frame) & is_text(Frame)	Ack_expected=FALSE Send="11"B Confirm="00"B DL_DATA.ind(extract_prtty, extract_text(Frame))	<i>M.Send</i>
<i>T.Com</i>	is_data+(Frame) & not(is_text(Frame))	Ack_expected=FALSE Send="11"B Confirm="00"B	<i>M.Send</i>
<i>M.Send</i>	exist_dl_data_req(DL_DATA.req(Pr=1, DSDU))	Discovered = TRUE Send=incr(Send) Ack_expected=TRUE Fr=DSDU Index=1 Size=size_frame(Fr) Com=com(DATA+,Pr,Send,Confirm) Fr=concat(Size, ADS, ADP, Com, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr)	M.Rec
<i>M.Send</i>	not(DL_DATA.req(Pr=1, _)) & exist_dl_data_req(DL_DATA.req(Pr=0, DSDU))	Discovered = TRUE Send=incr(Send) Ack_expected=TRUE Fr=DSDU Index=1 Size=size_frame(Fr) Com=com(DATA+,Pr,Send,Confirm) Fr=concat(Size, ADS, ADP, Com, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr)	M.Rec

Table 25 – Link-E/D state transitions: Secondary Station

Initial state	Triggering condition	Set of actions	Final state
<i>Initial</i>	\$true()	MaxRetry=2 FlagDSO=TRUE Discovered=FALSE Flag_alarm = FALSE	Stopped
Stopped	Phy_DATA.ind(Frame) & check_frame(Frame) & check_address(Frame)	ADP=extract_ADP(Frame) Com=command(Frame)	<i>T.Com</i>
Stopped	Phy_DATA.ind(Frame) & check_frame(Frame) & not(check_address(Frame))	Phy_ABORT.req()	Stopped
Stopped	Phy_DATA.ind(Frame) & not(check_frame(Frame))	\$none()	Stopped
Stopped	DL_ALARM.req() & alarm_detection()	Phy_ABORT.req() Flag_alarm = TRUE Phy_ALARM.req()	Stopped
<i>T.Com</i>	Com=IB	FlagDSO=TRUE Discovered=FALSE Phy_ABORT.req()	Stopped
<i>T.Com</i>	Com=ASO & test_TABi(Frame, TAB)	Fr=concat(RSO, TAB, ADS) Fr=concat(size_frame(Fr), ADS, ADP, Fr) Fr=concat(Fr, crc(Fr)) Phy_RSO.req(Fr, window_RSO())	Stopped
<i>T.Com</i>	Com=ASO & not(test_TABi(Frame, TAB))	Phy_ABORT.req()	Stopped
<i>T.Com</i>	is_data+(Frame) & is_text(Frame)	Ack_expected=FALSE Send="11"B Confirm="00"B DL_DATA.ind(extract_prty, extract_text(Frame))	<i>M.Send</i>
<i>T.Com</i>	is_data+(Frame) & not(is_text(Frame))	Ack_expected=FALSE Send="11"B Confirm="00"B	<i>M.Send</i>
<i>M.Send</i>	exist_dl_data_req(DL_DATA.req(Pr=1, DSDU))	Discovered = TRUE Send=incr(Send) Ack_expected=TRUE Fr=DSDU Index=1 Size=size_frame(Fr) Com=com(DATA+,Pr,Send,Confirm) Fr=concat(Size, ADS, ADP, Com, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr)	M.Rec
<i>M.Send</i>	not(DL_DATA.req(Pr=1, _)) & exist_dl_data_req(DL_DATA.req(Pr=0, DSDU))	Discovered = TRUE Send=incr(Send) Ack_expected=TRUE Fr=DSDU Index=1 Size=size_frame(Fr) Com=com(DATA+,Pr,Send,Confirm) Fr=concat(Size, ADS, ADP, Com, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr)	M.Rec

M.Send	not(DL_DATA.req(_, _))	Pr=0 Fr="" Index=1 Size=size_frame(Fr) Com=com(DATA+,Pr,Send,Confirm) Fr=concat(Size, ADS, ADP, Com, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr)	M.Rec
M.Rec	Phy_DATA.ind(Frame) & (check_frame(Frame) & check_address(Frame) & is_data+(Frame) & is_ack(Frame)) & is_text(Frame)	stop_timer(T1) Confirm=incr(Confirm) Ack_expected=FALSE DL_DATA.ind(extract_prt, extract_text(Frame))	M.Send
M.Rec	Phy_DATA.ind(Frame) & (check_frame(Frame) & check_address(Frame) & is_data+(Frame) & is_ack(Frame)) & not(is_text(Frame))	stop_timer(T1) Ack_expected=FALSE	M.Send
M.Rec	Phy_DATA.ind(Frame) & not(check_frame(Frame) & check_address(Frame) & is_data+(Frame) & is_ack(Frame)) & Index<=MaxRetry	stop_timer(T1) Phy_DATA.req(Fr) Index=Index+1	M.Rec
M.Rec	Phy_DATA.ind(Frame) & not(check_frame(Frame) & check_address(Frame) & is_data+(Frame) & is_ack(Frame)) & Index>MaxRetry	stop_timer(T1) DL_ABORT.ind(EL-2F) Phy_ABORT.req()	Stopped
M.Rec	DL_ABORT.req(Strong=0) & not(DL_DATA.req(_, _)) & Ack_expected=FALSE	stop_timer(T1) Phy_ABORT.req()	Stopped
M.Rec	DL_ALARM.req() & alarm_detection()	stop_timer(T1) DL_ABORT.ind(EL_1F) Phy_ABORT.req() Flag_alarm = TRUE Phy_ALARM.req()	Stopped
M.Rec	DL_ABORT.req(Strong=1)	stop_timer(T1) Phy_ABORT.req()	Stopped
M.Rec	Phy_ABORT.ind(EP-1)	init_timer(T1)	M.Rec
M.Rec	Phy_ABORT.ind(ErrorNb) & ErrorNb <> EP-1	stop_timer(T1) DL_ABORT.ind(ErrorNb)	Stopped
M.Rec	time_out(T1)	DL_ABORT.ind(EL_3F)	Stopped

M.Send	not(DL_DATA.req(_, _))	Pr=0 Fr="" Index=1 Size=size_frame(Fr) Com=com(DATA+,Pr,Send,Confirm) Fr=concat(Size, ADS, ADP, Com, Fr) Fr=concat(Fr, crc(Fr)) Phy_DATA.req(Fr)	M.Rec
M.Rec	Phy_DATA.ind(Frame) & (check_frame(Frame) & check_address(Frame) & is_data+(Frame) & is_ack(Frame)) & is_text(Frame)	stop_timer(T1) Confirm=incr(Confirm) Ack_expected=FALSE DL_DATA.ind(extract_prt, extract_text(Frame))	M.Send
M.Rec	Phy_DATA.ind(Frame) & (check_frame(Frame) & check_address(Frame) & is_data+(Frame) & is_ack(Frame)) & not(is_text(Frame))	stop_timer(T1) Ack_expected=FALSE	M.Send
M.Rec	Phy_DATA.ind(Frame) & not(check_frame(Frame) & check_address(Frame) & is_data+(Frame) & is_ack(Frame)) & Index<=MaxRetry	stop_timer(T1) Phy_DATA.req(Fr) Index=Index+1	M.Rec
M.Rec	Phy_DATA.ind(Frame) & not(check_frame(Frame) & check_address(Frame) & is_data+(Frame) & is_ack(Frame)) & Index>MaxRetry	stop_timer(T1) DL_ABORT.ind(EL-2F) Phy_ABORT.req()	Stopped
M.Rec	DL_ABORT.req(Strong=0) & not(DL_DATA.req(_, _)) & Ack_expected=FALSE	stop_timer(T1) Phy_ABORT.req()	Stopped
M.Rec	DL_ALARM.req() & alarm_detection()	stop_timer(T1) DL_ABORT.ind(EL_1F) Phy_ABORT.req() Flag_alarm = TRUE Phy_ALARM.req()	Stopped
M.Rec	DL_ABORT.req(Strong=1)	stop_timer(T1) Phy_ABORT.req()	Stopped
M.Rec	Phy_ABORT.ind(EP-1)	init_timer(T1)	M.Rec
M.Rec	Phy_ABORT.ind(ErrorNb) & ErrorNb <> EP-1	stop_timer(T1) DL_ABORT.ind(ErrorNb)	Stopped
M.Rec	time_out(T1)	DL_ABORT.ind(EL_3F)	Stopped

Tableau 26 – Signification des états mentionnés dans les tableaux précédents

Etat	Signification
<i>Initial</i>	Initialisation des variables de la couche
Stopped	Attente de la première demande de la couche supérieure ou de la première indication de la couche inférieure
W.AG (Wait for end of «Appel Général»)	Attente de la fin d'un signal «Appel Général»
W.EndS (Wait for end of Session)	Attente de la fin d'une session
T.Reg (Test Request)	Test de la nature d'une requête provenant de la couche supérieure
M.RSO (Must Receive RSO)	Attente d'une trame de réponse RSO après l'envoi d'une trame ASO
T.RSO (Test last RSO)	Test du dernier intervalle de temps pour la réception d'une trame RSO
M.Send (Must Send)	Test de la trame à envoyer (le champ Text peut être vide)
M.Rec (Must Receive)	Attente d'une indication en provenance de la couche inférieure
T.Com (Test Command)	Test du champ COM d'une trame reçue

Table 26 – Meaning of the states listed in the previous tables

State	Meaning
<i>Initial</i>	Initialization of the variables of the layer
Stopped	Waiting state for the first request from the upper layer or the first indication from the lower layer
W.AG (Wait for end of "Wakeup Call")	Waiting for the end of a requested "Wakeup Call" signal
W.EndS (Wait for End of Session)	Waiting for the end of the session
<i>T.Reg</i> (Test Request)	Test the nature of a request from the upper layer
M.RSO (Must receive RSO)	Waiting for an RSO frame following an ASO frame sent
<i>T.RSO</i> (Test last RSO)	Testing the end of the last time slot for RSO frame reception
<i>M.Send</i> (Must Send)	A frame shall be sent (possibly with the Text field empty)
M.Rec (Must Receive)	Waiting state for an indication from the lower layer
<i>T.Com</i> (Test Command)	Test of the command code of a received frame

Tableau 27 – Définition des procédures et des fonctions classées dans l'ordre alphabétique

Procédure ou fonction	Définition
alarm_detection()	Vérification que le mode alarme est Actif
build_RSO(ListRSO, Frame)	Extraction des éléments (champs TAB et ADS) de la trame RSO reçue Frame et concaténation avec la précédente liste ListRSO
check_address(Frame)	Vérification que les adresses ADP et ADS sont reconnues selon les critères suivants: – la station est sensibilisée à l'ADP ou ADP = APG; – ADS=ADG si la commande est ASO ou IB, – ADS est l'adresse de la station secondaire si la commande n'est pas ASO ou IB
check_frame(Frame)	Vérification que la trame Frame reçue est correcte: – nombre d'octets supérieur ou égal à 11 et inférieur ou égal à 128; – CRC correct; – nombre d'octets compatible avec le champ Size; – commande connue
com(DATA+, Pr, Send, Confirm)	Concaténation des champs binaires correspondants pour obtenir une commande spécifique
command(Frame)	Extraction de la valeur de la commande codée dans le champ COM de la trame Frame reçue
concat(Size, ADS, ADP, COM, Text), ou concat(Frame, CRC)	Concaténation des champs Size, ADS, ADP, COM et Text ou concaténation du CRC à la fin de la trame Frame
context(ADS, ADP, TypeAG)	Extraction des valeurs correspondantes du contexte de communication
crc(Frame)	Calcul du CRC de la trame Frame à émettre
create_alarm(TPDU)	Calcul d'un TPDU avec STSAP = 0, DTSAP = 0 et un UnsolicitedReqPDU avec: client-type = FFFF, serveridentifiant = 0, object-name = FFFF variable type = boolean, value = TRUE
exist_dl_data-req(DL_DATA.req(Pr, DSDU))	Consommation d'un événement DL_DATA.req(Pr, DSDU)
exist_dl_req()	Vérification de l'existence d'un événement DL_IB.req(), DL_ASO.req(DSDU) ou DL_DATA.req(Pr, DSDU) et vérification de la compatibilité avec les adresses ADS et ADP du contexte de communication
exist_dl_req(DL_IB.req()) ou exist_dl_req(DL_ASO.req(DSDU))	Consommation d'un événement DL_IB.req() ou DL_ASO.req(DSDU)

Table 27 – Definition of the procedures and functions classified in alphabetical order

Procedure or function	Definition
Alarm_detection()	Check that the status of alarm mode is Active
build_RSO(ListRSO, Frame)	Extraction of the RSO elements (TAB and ADS fields) from the received RSO frame Frame and concatenation with the preceding list ListRSO
check_address(Frame)	Check that the ADP and ADS addresses are recognized according to the following criteria: – ADP is APG or the station has been programmed to the ADP address; – if the command code is ASO or IB, then ADS is ADG; – if the command code is not ASO nor IB, then ADS is the secondary station address
check_frame(Frame)	check that the frame Frame received is correct: – number of octets greater than or equal to 11 and lower than or equal to 128; – CRC correct; – number of octets compatible with the field Size; – command code recognized
com(DATA+, Pr, Send, Confirm)	Concatenation of the corresponding bit fields to produce a specific command code
command(Frame)	Extraction of the value of the command code of the received frame Frame
concat(Size, ADS, ADP, COM, Text) or concat(Frame, CRC)	Concatenation of the fields Size, ADS, ADP, COM and Text or concatenation of the CRC at the end of the frame Frame
context(ADS, ADP, TypeAG)	Extraction of the corresponding values from the communication context
crc(Frame)	Calculation of the CRC of the frame Frame to send
create_alarm(TPDU)	Calculation of a TPDU where STSAP = 0, DTSAP = 0 and an UnsolicitedReqPDU with: client-type = FFFF, serveridentifier = 0, object-name = FFFF variable type = boolean, value = TRUE
exist_dl_data_req(DL_DATA.req(Pr, DSDU))	Consumption of a DL_DATA.req(Pr, DSDU) event
exist_dl_req()	Check the existence of a DL_IB.req(), DL_ASO.req(DSDU) or DL_DATA.req(Pr, DSDU) event and check the compatibility with ADP and ADS addresses defined in the communication context
exist_dl_req(DL_IB.req()) or exist_dl_req(DL_ASO.req(DSDU))	Consumption of a DL_IB.req() or DL_ASO.req(DSDU) event

extract_ADP(Frame)	Si l'adresse de la Station Primaire utilisée dans la trame est différente de APG ou bien s'il s'agit de APG mais que la liste des adresses primaires auxquelles est sensibilisée la Station Secondaire est vide, extraction de cette valeur, sinon extraction de la première valeur ADP de la liste des adresses primaires auxquelles est sensibilisée la Station Secondaire
extract_prtty(Frame)	Extraction du champ Priority de la trame Frame reçue
extract_text(Frame)	Extraction du champ Text de la trame Frame reçue
init(TypeAG)	Met Index à MaxRetry si TypeAG vaut AGT, à 0 sinon
init_incrChain()	Met IncrChain à 0 si les alarmes ne sont pas gérées, à 1 sinon
init_timer(T1)	Armement du réveil T1
is_ack (Frame)	Vérification que la trame Frame reçue contient un champ Confirm égal au champ Send de la dernière trame émise
is_data+ (Frame)	Vérification que la trame Frame reçue contient un champ DATA+ correct ("111'B)
is_text(Frame)	Vérification que la trame Frame reçue contient un champ de données Text non vide et que le champ Send est égal au champ Confirm de la dernière trame émise
size(Frame)	Calcul de la taille de la trame Frame reçue
size_frame(DSDU)	Calcul de la taille de la trame à construire à partir de l'unité de données DSDU (taille de DSDU+11)
stop_timer(T1)	Désarmement du réveil T1
test_TABi(Frame, TAB)	Si le premier des TABi contenus dans la trame Frame de type ASO vaut 00, vérification que la variable Discovered est à FAUX et vérification, après tirage d'un nombre aléatoire entre 0 et 100, que ce nombre est inférieur à la probabilité de réponse souhaitée au deuxième des TABi. Dans ce cas la valeur 00 est mémorisée dans la variable TAB; si le premier TABi contenu dans la trame Frame de type ASO vaut FF, vérification que la variable Flag_alarm est à VRAI et dans ce cas affectation de la valeur FAUX à la variable Flag_alarm et mémorisation de la valeur FF dans la variable TAB; sinon vérification que la variable FlagDSO est à VRAI, que la Station Secondaire est sensibilisée à l'un des TABi contenus dans la trame Frame de type ASO reçue et mémorisation de la première de ces valeurs dans la variable TAB
time_out(T1)	Armement du réveil T1
window_RSO()	Récupération d'un nombre aléatoire entre 0 et MaxRSO-1, qui donne le numéro de l'intervalle de temps RSO pendant lequel la station doit répondre (voir annexe F)

extract_ADp(Frame)	If either the ADP value used in the frame is not APG or if it is the APG value but the list of the ADP values to which the Secondary Station has been programmed is empty, then extraction of this value, else extraction of the first ADP value to which the Secondary Station has been programmed
extract_prty(Frame)	Extraction of the Priority field from a received frame Frame
extract_text(Frame)	Extraction of the Text field from a received frame Frame
init(TypeAG)	Set Index to MaxRetry if TypeAG equals AGT, else to 0
init_incrChain()	Set IncrChain to 0 if alarm supported, else to 1
init_timer(T1)	Setting of wakeup T1
is_ack(Frame)	Check that the received frame Frame contains a Confirm field equal to the Send field of the last frame transmitted
is_data+(Frame)	Check that the received frame Frame contains a correct DATA+ field ("111"B)
is_text(Frame)	Check that the received frame Frame contains a non-empty Text field and that the Send field equals the Confirm field of the last frame transmitted
size(Frame)	Calculation of the size of the frame Frame received
size_frame(DSDU)	Calculation of the size of the frame to build with the DSDU data unit (size(DSDU) + 11)
stop_timer(T1)	Stopping of wakeup T1
test_TABi(Frame, TAB)	If the first TABi contained in the received ASO frame Frame equals 00, check that Discovered=FALSE then, after provision of a whole random integer between 0 and 100, check that this integer is smaller than the response probability (second TABi). In this case, 00 is memorized in the TAB variable if the first TABi contained in the received ASO frame Frame equals FF, check that Flag_alarm=TRUE then, in this case, Flag_alarm is set to FALSE and FF is memorized in the TAB variable; if the first TABi contained in the received ASO frame Frame does not equal 00 or FF, check that FlagDSO=TRUE and check that the Secondary Station has been programmed to one of the TABi contained in the received ASO frame Frame. In this case, the first of these values is memorized in the TAB variable
time_out(T1)	Triggering of wakeup T1
window_RSO()	Provision of a whole random integer between 0 and MaxRSO-1 used as the number of the RSO time slot in which the station shall reply (refer to annex F)

4.2.6 Répertoire et traitement des erreurs

Les erreurs sont répertoriées par le codage suivant:

EL = erreur de la couche *Liaison*

– = séparateur

N = numéro de l'erreur

F = erreur fatale

Tableau 28 – Tableau récapitulatif des erreurs

EL-1F	DL_ALARM.req() reçue durant une association
	Cette erreur conduit à réinitialiser la couche <i>Liaison</i> après avoir demandé à la couche <i>Physique</i> de transmettre une alarme
EL-2F	Réponse incorrecte de la Station Secondaire après MaxRetry répétitions de la requête
	Cette erreur conduit à réinitialiser la couche <i>Liaison</i> après avoir informé la couche <i>Application</i> et provoqué l'arrêt de la couche <i>Physique</i>
EL-3F	Fin de communication après un silence d'une durée T1
	Cette erreur conduit à réinitialiser la couche <i>Liaison</i> après avoir informé la couche <i>Application</i> et provoqué l'arrêt de la couche <i>Physique</i>

Toute occurrence de l'une de ces erreurs fatales est remontée localement grâce à la primitive de service DL_ABORT.ind. La liste complète des numéros d'erreur fatale est fournie à l'annexe C.

4.3 Couche Application

Les spécifications de la couche *Application* sont données dans la CEI 62056-51. Ce paragraphe clarifie simplement le profil utilisé pour l'architecture d'échange de données par bus local avec DLMS.

4.3.1 Sous-couche Transport

La valeur du nombre MaxPktSize (voir CEI 62056-51) taille maximale du champ *Packet*, doit être initialisée à 114.

4.3.2 Sous-couche Application

Comme cela est indiqué dans la CEI 62056-51, les fonctions client_connect et server_connect doivent être précisées en fonction du support de communication utilisé. Pour l'architecture d'échange de données par bus local avec DLMS, le service non sollicité n'étant pas supporté, la fonction server_connect n'est pas acceptée. La fonction client_connect est définie dans le tableau 29.

4.2.6 List and processing of errors

Errors are listed using the following codes:

EL = error in the *Data Link* layer

– = separator

N = error number

F = fatal error

Table 28 – Error summary table

EL-1F	DL_ALARM.req() received during an association
	This request leads to the reinitialization of the <i>Data Link</i> layer after having requested the <i>Physical</i> layer to transmit an Alarm
EL-2F	Incorrect response from the Secondary Station after MaxRetry repeated transmissions of a request
	This error leads to the reinitialization of the <i>Data Link</i> layer after having informed the <i>Application</i> layer and caused the <i>Physical</i> layer to abort
EL-3F	End of communication due to expiration of delay T1
	This error leads to the reinitialization of the <i>Data Link</i> layer after having informed the <i>Application</i> layer and caused the <i>Physical</i> layer to abort

If this error occurs, it is sent up locally by means of the DL_ABORT.ind service primitive. The complete list of fatal error numbers is given in annex C.

4.3 Application layer

The *Application* layer specification is described in IEC 62056-51. This subclause simply clarifies the operating profile for the local bus data exchange architecture with DLMS.

4.3.1 Transport sub-layer

The value of the number MaxPktSize (refer to IEC 62056-51), maximum size of *Packet* field, shall be set at 114.

4.3.2 Application sub-layer

As stated in IEC 62056-51, the client_connect and server_connect functions shall be clarified according to the communication medium used. For the local bus data exchange architecture with DLMS, the server_connect function is not accepted because the unsolicited service management is not supported. The client_connect function is defined in the table 29.

Tableau 29 – Définition de la fonction client_connect

Procédure ou fonction	Définition
client_connect(Ads, Adp)	S'il n'existe aucune association d'application active, la fonction établit les paramètres du contexte de communication: ADS, ADP et TypeAG, utilisés par les couches inférieures. La fonction est transparente s'il y a déjà une association d'application active avec les mêmes adresses (ADP, ADS) La fonction échoue s'il existe déjà une association d'application avec un couple d'adresse (ADP, ADS) différent.

5 Echange des données par bus en local – Spécifications matérielles

5.1 Généralités

Le protocole spécifie l'échange de données entre une Station Primaire et des Stations Secondaires connectées en parallèle sur un bus physique. La Station Primaire est connectée au bus par un coupleur magnétique passif.

Cet article décrit les points suivants:

- a) les signaux échangés;
- b) les caractéristiques du bus;
- c) le coupleur magnétique;
- d) la Station Primaire;
- e) la Station Secondaire;
- f) les caractéristiques de l'alimentation en énergie.

5.2 Caractéristiques générales

5.2.1 Signal de transmission à 50 kHz

La transmission comprend:

- a) la transmission de données binaire;
- b) le mode de transmission bidirectionnel, half-duplex;
- c) la vitesse de modulation (en bauds): $1\,200\text{ Bd} \pm 1\%$;
- d) la durée des bits 0 et 1 identique;
- e) la modulation d'amplitude d'une porteuse à 50 kHz, avec une tolérance relative de $\pm 3\%$;
- f) la logique de codage négative:
 - bit 0 = présence de porteuse,
 - bit 1 = absence de porteuse.

Table 29 – Client_connect function definition

Procedure or function	Definition
client_connect(Adp, Ads)	If there is no active application association, the function sets the context parameters ADS, ADP and TypeAG used by the lower layers. The function is transparent if there is already an application association active with the same (ADP, ADS) addresses. The function fails if there is already an application association with different (ADP, ADS) addresses

5 Local bus data exchange – Hardware

5.1 General

The protocol describes the data exchanges between a Primary Station and Secondary Stations connected in parallel on a hardware bus. The Primary Station is connected to the bus by a passive magnetic plug.

This clause describes the following items:

- a) the signal characteristics,
- b) the bus characteristics,
- c) the magnetic plug,
- d) the Primary Station,
- e) the Secondary Station,
- f) the energy supply characteristics.

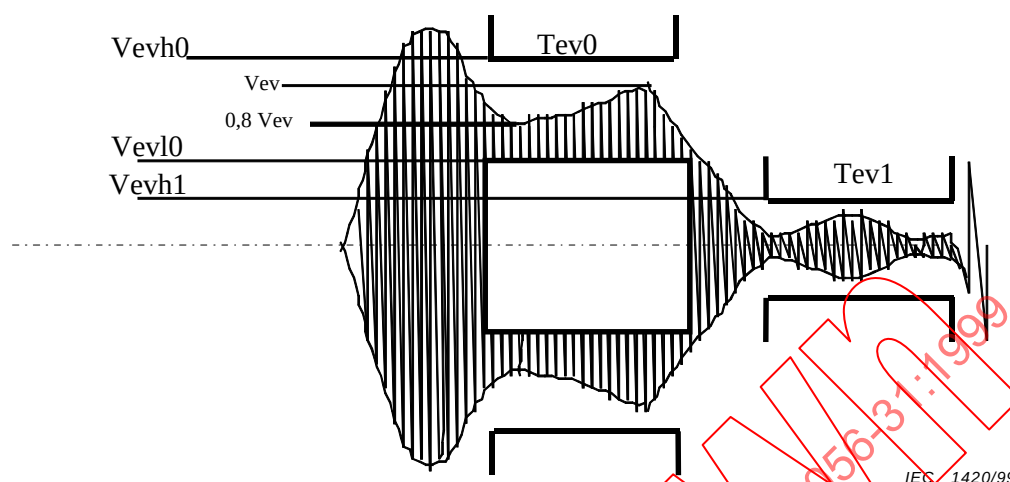
5.2 General characteristics

5.2.1 Signal transmission at 50 kHz

Transmission includes:

- a) binary data transmission
- b) bi-directional, half-duplex
- c) baud rate: 1 200 Bd $\pm$ 1 %
- d) equal duration of bits 0 and 1
- e) amplitude signal modulation (ASM) of a 50 kHz $\pm$ 3 % carrier
- f) polarity:
 - 0 = carrier detected,
 - 1 = carrier not detected

Les caractéristiques du signal sont définies par rapport à l'enveloppe de la porteuse, tel que cela est décrit à la figure 5.



Légende

Vevh1 est le niveau maximal de l'enveloppe pour la transmission d'un "1"

Vevl0 est le niveau minimal de l'enveloppe pour la transmission d'un "0"

Vevh0 est le niveau maximal de l'enveloppe pour la transmission d'un "0"

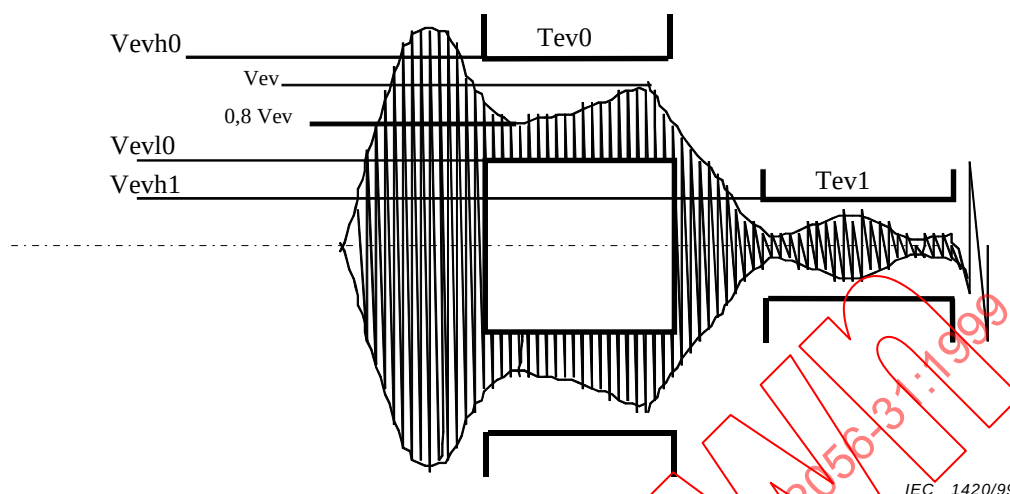
Tev1 est le temps minimal garanti pendant lequel l'enveloppe reste inférieure à Vevh1

Tev0 est le temps minimal garanti pendant lequel l'enveloppe reste comprise entre Vevl0 et Vevh0

Figure 5 – Enveloppe du signal sur le bus

- Vevl0 et Vevh0 ne sont pas les valeurs extrêmes de l'enveloppe, mais plutôt les limites basses et hautes garantissant un fonctionnement correct.
- Pendant la durée Tev0, le niveau de l'enveloppe ne doit pas varier de plus de 20 %.
- Durant les intervalles de temps qui s'écoulent entre Tev0 et Tev1, l'évolution croissante ou décroissante de l'enveloppe est de type exponentielle ou sinusoïde amortie, avec addition de transitoires de fréquence.
- Le taux de distorsion harmonique, pendant une émission continue de la porteuse est inférieur à 15 %, avec une résistance de 100 Ω ou une capacité de 31,8 nF à la place du bus.
- Toutes les tensions sont spécifiées en valeurs crêtes.
- Pour la définition de la Durée Binaire (durée d'un "1" ou d'un "0"), plusieurs paramètres sont à prendre en compte:
 - la durée maximale garantie pour la transmission d'un "0" signal $\geq$ Vevl0;
 - la durée maximale non garantie pour la transmission d'un "0" signal $\geq$ Vevh1;
 - la durée maximale garantie pour la transmission d'un "1" signal $\leq$ Vevh1;
 - la durée maximale non garantie pour la transmission d'un "1" signal $\leq$ Vevl0.

The signal characteristics are defined by the carrier envelope described in the figure 5.



Key

Vevh1 is the maximum level for transmission of a "1"

Vevl0 is the minimum level for transmission of a "0"

Vevh0 is the maximum level for transmission of a "0"

Tev1 is the minimum guaranteed time for an envelope to remain lower than Vevh1

Tev0 is the minimum guaranteed time for an envelope to remain between Vevl0 and Vevh0

Figure 5 – Signal envelope on the bus

- a) Vevl0 and Vevh0 are not the extremes of the envelope, but rather the low and high limits for correct operation.
- b) During Tev0 the level of the envelope shall not vary by more than 20 %.
- c) During the gaps between Tev0 and Tev1, the envelope rise or fall is exponential or damped sinusoid, with addition of frequency transients.
- d) Total harmonic distortion of the signal during continuous wave transmission is less than 15 %, with a resistor of 100 Ω or a capacitor of 31,8 nF in place of the bus.
- e) All the voltages are specified in peak values.
- f) For Time Bit definition, (time of "1" or "0"), several parameters have to be taken into account:

– maximum time guaranteed for "0" transmission	signal $\geq$ Vevl0
– maximum time non-guaranteed for "0" transmission	signal $\geq$ Vevh1
– maximum time guaranteed for "1" transmission	signal $\leq$ Vevh1
– maximum time non guaranteed for "1" transmission	signal $\leq$ Vevl0

5.2.2 Signal pour l'alimentation en énergie

5.2.2.1 Caractéristiques de la télé-alimentation

La représentation du bus en mode télé-alimenté est donnée à la figure 6.

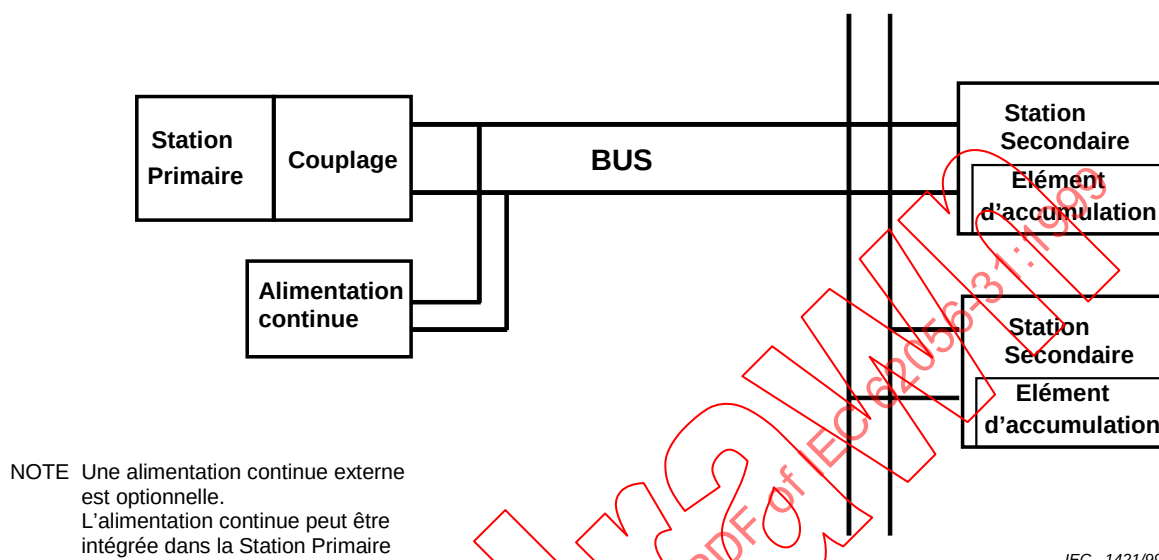


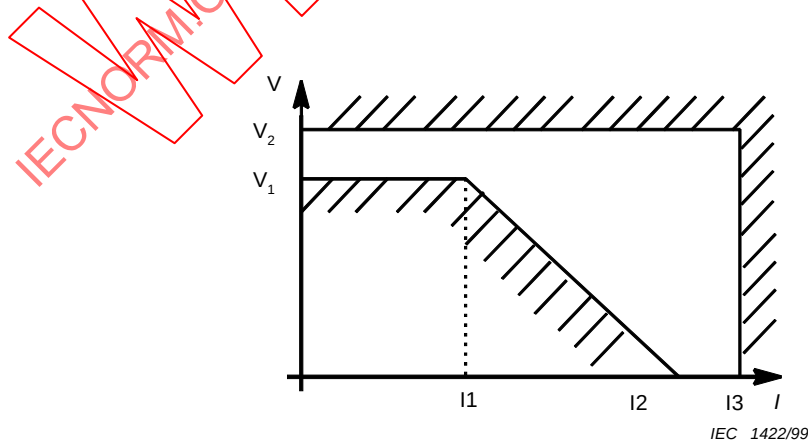
Figure 6 – Représentation du bus

Les trois éléments principaux pour la télé-alimentation sont:

- une source d'alimentation en courant continu;
- un élément d'accumulation d'énergie sur les Stations Secondaires;
- la consommation d'une Station Secondaire.

5.2.2.2 Source d'alimentation en énergie

La source d'alimentation en énergie fournit sur le bus des valeurs de tension et de courant respectant le gabarit de la figure 7.



Légende

$V_1 = 22 \text{ V}$, $V_2 = 35 \text{ V}$

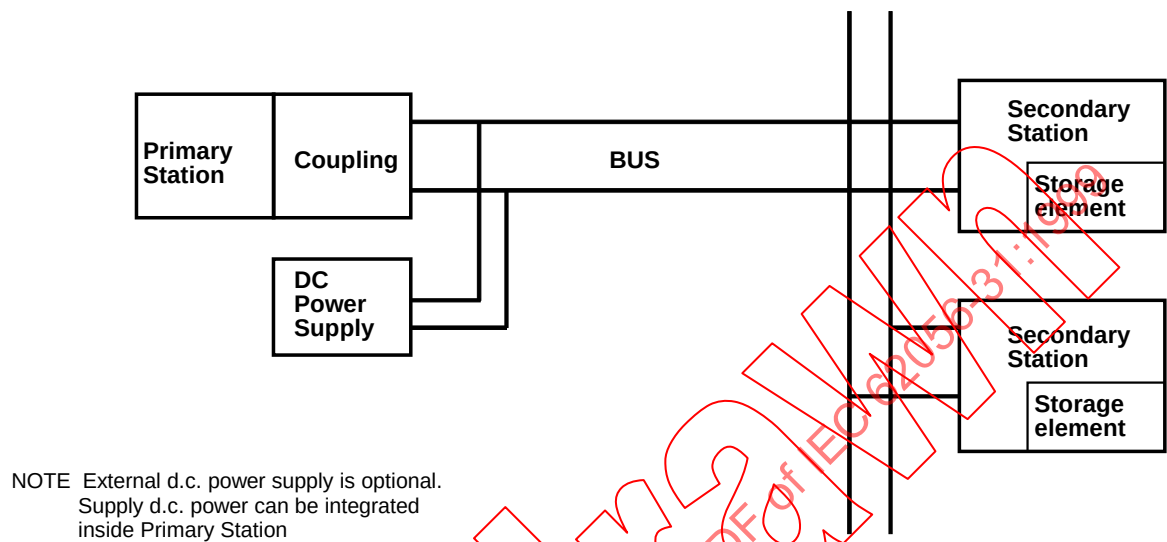
$I_1 = 80 \text{ mA}$, $I_2 = 250 \text{ mA}$, $350 \text{ mA} \leq I_3 \leq 1\,000 \text{ mA}$

Figure 7 – Caractéristiques de l'alimentation en énergie

5.2.2 Energy supply signal transmission

5.2.2.1 Characterization of the remote supply of energy

The figure 6 gives the representation of the bus for the remote supply of energy.



IEC 1421/99

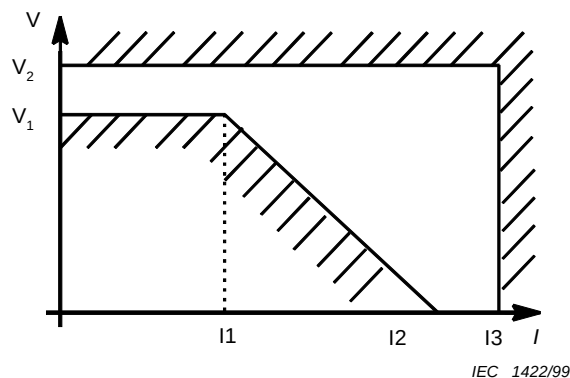
Figure 6 – Bus representation

The three main elements for the remote supply of energy are:

- a.d.c. energy supply source,
- a storage element in the Secondary Station,
- the consumption of a Secondary Station.

5.2.2.2 Energy supply source

The energy supply source provides the bus with the values of voltage and current according to the template of the figure 7.



Key

$V_1 = 22 \text{ V}$, $V_2 = 35 \text{ V}$

$I_1 = 80 \text{ mA}$, $I_2 = 250 \text{ mA}$, $350 \text{ mA} \leq I_3 \leq 1\,000 \text{ mA}$

Figure 7 – Power supply characteristics

Le niveau de bruit sera inférieur à 10 mV crête dans la bande de 1 kHz à 1 MHz et à 100 mV crête pour $f < 1$ kHz.

La nature de l'alimentation en énergie n'est pas précisée dans cette spécification.

5.2.2.3 Élément d'accumulation dans la Station Secondaire

Un élément d'accumulation d'énergie associé à chaque Station Secondaire permet une accumulation locale d'énergie de manière à absorber les pics de consommation durant les échanges et à optimiser la télé-alimentation. (Valeur maximale 470 μ F avec une tolérance relative de + 20 %.)

5.2.2.4 Consommation d'une Station Secondaire

Les états rencontrés lors d'une session sont décrits dans les figures 8 et 9.

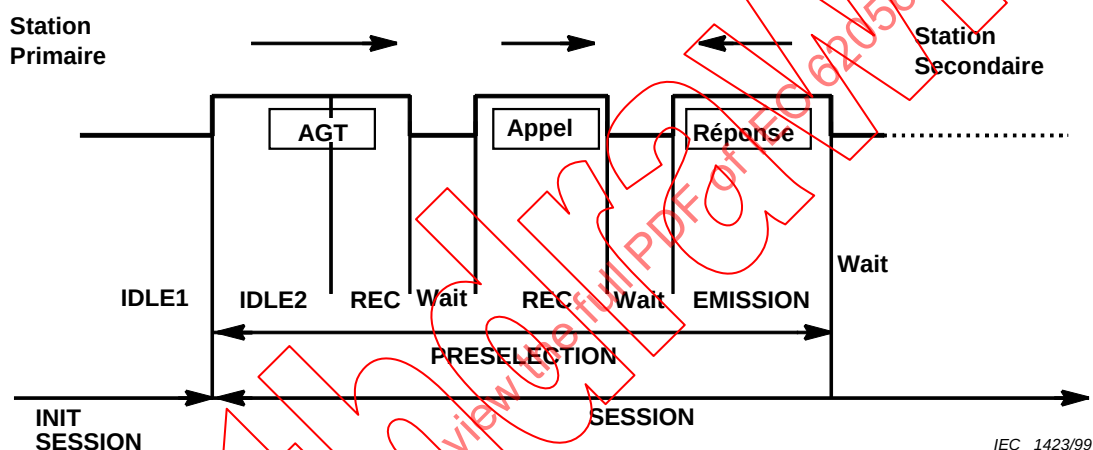


Figure 8 – Etats d'une session: sélection d'une Station Secondaire

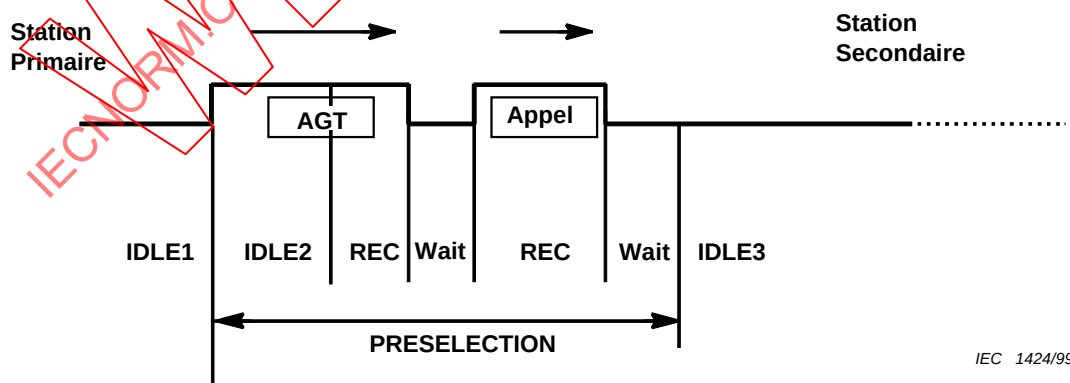


Figure 9 – Etats d'une session: Station Secondaire non sélectionnée

The ripple noise will be less than 10 mV peak from 1 kHz to 1 MHz and 100 mV peak for $f < 1$ kHz.

The nature of the energy supply is not taken in account in this specification.

5.2.2.3 Storage Element in Secondary Station

A storage element associated with each Secondary Station allows the local accumulation of energy in order to absorb peaks of consumption during exchanges and to optimize the remote energy supply source. (Maximum value of 470 μ F with a relative tolerance of + 20 %.)

5.2.2.4 Consumption of a Secondary Station

States associated to an exchange session are shown in the figures 8 et 9.

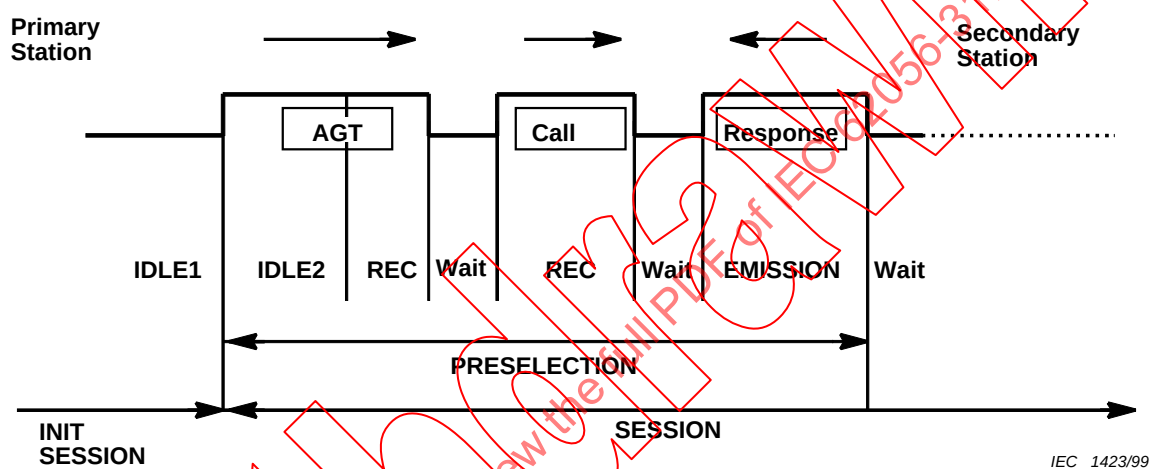


Figure 8 – States associated to a session: for selected Secondary Station

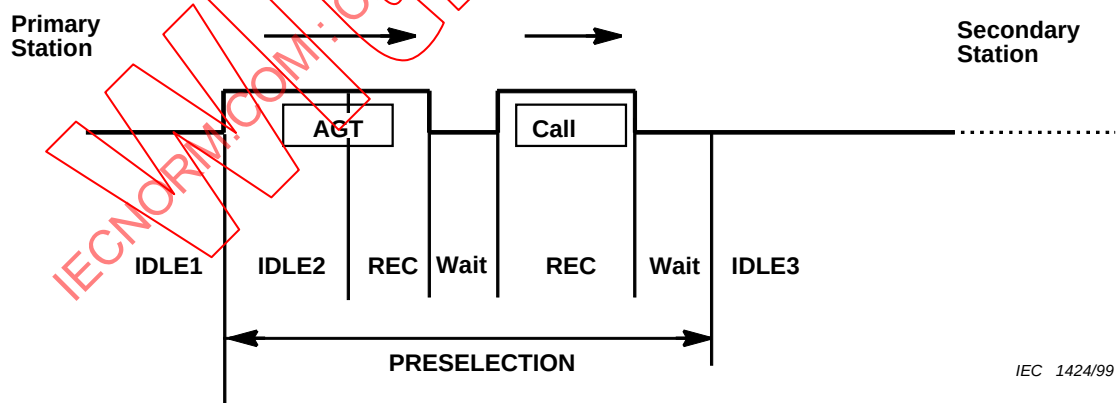


Figure 9 – States associated to a session: for non-selected Secondary Station

La consommation en fonction des états est donnée par les valeurs qui suivent.

Ces valeurs représentent le maximum pour la consommation moyenne: les valeurs crêtes peuvent être supérieures, mais les valeurs moyennes restent inférieures:

- Idle mode 1 15 mW max. (mode établi)
- Idle mode 2 15 mW max.
- Idle mode 3 15 mW max.
- mode Réception 40¹⁾ mW max. + nx10¹⁾ mW max. (n = nombre d'équipements)
- mode Wait 40 mW max. + nx10 mW max. (n = nombre d'équipements)
- Etat Emission 140¹⁾ mW max. + nx10 mW max. (n = nombre d'équipements)

Le pire des cas correspond à la transmission de 9/10 bits "0":

- mode Réception 50¹⁾ mW max. + nx14¹⁾ mW max. (n = nombre d'équipements)
- Etat Emission 180¹⁾ mW max. + nx14¹⁾ mW max. (n = nombre d'équipements)

La description détaillée de ces différents états (avec les temporisations associées) est donnée avec les tables de transitions d'états.

5.2.3 Station Secondaire simple et Station Secondaire multiple

Une Station Secondaire simple correspond à une seule adresse logique ADS.

Une Station Secondaire multiple correspond à plusieurs adresses logiques ADS.

La notion de Station Secondaire multiple permet d'adresser par l'intermédiaire d'un bus secondaire différents équipements situés dans une Station Secondaire.

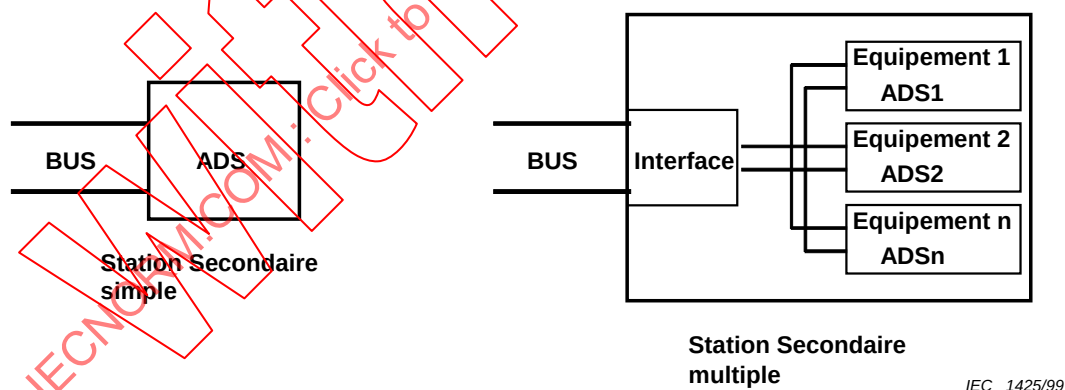


Figure 10 – Station Secondaire simple ou multiple

Les points fondamentaux de la structure du bus secondaire sont les suivants:

- a) La longueur du bus secondaire n'est pas comprise dans la longueur du bus principal.
- b) Le bus secondaire est formé de quatre fils, avec deux conducteurs pour le signal et deux conducteurs pour l'alimentation en énergie.

1) La consommation d'énergie en Emission et en Réception dépend du nombre de "0" dans le message. Ces valeurs sont des consommations moyennes avec un nombre égal de bits "0" et de bits "1".

The consumption, depending on the states, is specified below.

These values represent maximum average consumption: peak can go over but average value remains under:

- Idle mode 1 15 mW max. (established mode)
- Idle mode 2 15 mW max.
- Idle mode 3 15 mW max.
- Reception mode $40^{1)}$ mW max. + $n \times 10^{1)}$ mW max. (n = number of devices)
- Wait mode 40 mW max. + $n \times 10$ mW max. (n = number of devices)
- Emission state $140^{1)}$ mW max. + $n \times 10$ mW max. (n = number of devices)

Worst case is the transmission of 9/10 "0" bits:

- Reception mode $50^{1)}$ mW max. + $n \times 14^{1)}$ mW max. (n = number of devices)
- Emission state $180^{1)}$ mW max. + $n \times 14^{1)}$ mW max. (n = number of devices)

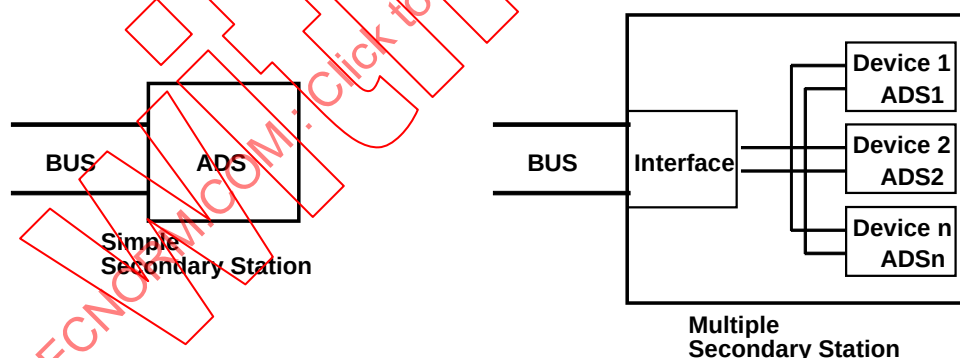
The detailed description of these different modes (with the associated timing) is given in the state transitions.

5.2.3 Simple Secondary Station and multiple Secondary Station

A simple Secondary Station is equivalent to a logical address ADS.

A multiple Secondary Station is equivalent to several logical addresses ADS.

The notion of multiple Secondary Station allows the addressing of different devices located inside a Secondary Station on a secondary bus.



IEC 1425/99

Figure 10 – Simple and multiple Secondary Station

The secondary bus structure fundamental points are as follows:

- a) The length of secondary bus is not included in the length of main bus.
- b) The secondary bus is four wires with two wires dedicated to signal and two wires dedicated to energy supply.

¹⁾ Reception and Emission consumption depend on the message content number of "0". Those values are average consumptions with an equal number of "0" and "1" bit.

- c) Une Station Secondaire télé-alimentée, simple ou multiple, est toujours équivalente au maximum à deux Stations Secondaires vis-à-vis des paramètres de communication physiques.
- d) Le protocole sur le bus primaire ou secondaire est le même, sauf pour l'aspect modulation: la transmission des données sur le bus secondaire se fait en bande de base (1 200 Bd), conformément à la EIA485 et à l'ISO/IEC 8482.
- e) La longueur maximale du bus secondaire est de 50 m.
- f) Le nombre maximal d'équipements sur le bus secondaire est de 6.
- g) Le câble est le même que celui utilisé pour le bus primaire.
- h) Il est possible de signaler une alarme depuis un équipement à une Station Primaire vers l'interface lorsqu'il y a alimentation d'énergie sur le bus; cette information est effectuée grâce à l'émission d'une porteuse TAB. La Station Primaire, l'interface et les équipements doivent être en mode Alarme Actif (voir 2.4.11).

Les stations suivantes peuvent se raccorder sur le bus principal:

- 1) une Station Secondaire simple alimentée;
- 2) une Station Secondaire simple télé-alimentée;
- 3) une Station Secondaire multiple télé-alimentée.

5.3 Spécification du bus

5.3.1 Caractéristiques générales

- a) Support spécifique pour la télérelève et la téléprogrammation, le bus est constitué au minimum d'une embase magnétique et d'une Station Secondaire.
- b) La topologie du bus n'est pas imposée; elle peut être linéaire, en étoile ou en arbre sans boucles, pourvu que la longueur totale du câble ne dépasse pas 500 m (câble de l'alimentation inclus). La longueur du bus secondaire, à l'intérieur d'une Station Secondaire, n'est pas comprise dans cette valeur.
- c) Le bus permet l'alimentation en énergie des Stations Secondaires. Il peut donc supporter des stations alimentées et télé-alimentées.
- d) Une isolation galvanique est maintenue entre le bus et toutes les électroniques de transmission et de réception, avec un niveau de tension correspondant à ce qui est spécifié pour les Stations Secondaires.
- e) L'alimentation peut être permanente sur le bus. Dans ce cas, le mode Alarme peut être actif.
- f) De 1 à 100 Stations Secondaires peuvent être connectées en parallèle sur le bus.

Le nombre maximal de Stations Secondaires télé-alimentées (simples ou multiples) connectées sur le bus est égal à 50.

Le nombre maximal de Stations Secondaires alimentées connectées sur le bus est égal à 100.

Le nombre maximal d'équipements associés à des Stations Secondaires (simples ou multiples) connectées sur le bus est égal à 50.

Lorsque des Stations Secondaires de différents types, alimentées ou télé-alimentées, sont connectées sur le même bus, le nombre maximal de Stations de chaque type est donné par la règle suivante:

si $N1$ est le nombre de Stations Secondaires multiples télé-alimentées sur le bus,

si $N2$ est le nombre de Stations Secondaires simples télé-alimentées sur le bus,

- c) A Secondary non-energized Station, simple or multiple, is always equivalent to a maximum of two Secondary Stations towards physical communication parameters.
- d) The primary and secondary bus protocol are the same, except for modulation aspects: signals on secondary bus are baseband (1 200 Bd): signal wires. They are in accordance with EIA485 and with ISO/IEC 8482.
- e) The maximum length of the secondary bus is 50 m.
- f) The maximum number of devices on the secondary bus is 6.
- g) The cable is the same as the one used on the primary bus.
- h) It is possible to transmit alarms from a device to a primary station through the interface, when power supply is permanent on this bus. This report is made by carrier transmission TAB. Primary station, interface and devices shall be in Alarm mode active (see 2.4.11).

Main bus is able to support the following stations:

- 1) Simple energized Secondary Station,
- 2) Simple non-energized Secondary Station,
- 3) Multiple non-energized Secondary Station.

5.3 Bus specification

5.3.1 General characteristics

- a) Specific support for remote reading and programming. The bus always has a magnetic socket and one Secondary Station (minimum).
- b) The bus topology is unimportant, and can be linear, or star or tree without loop, provided that the total wired length of cable does not exceed 500 m (cable of power supply included). The secondary bus, inside a Secondary Station, is not included in this value.
- c) The bus allows energy supply for Secondary Stations. In this way, the bus can support energized and non-energized stations.
- d) Galvanic isolation is maintained between the bus and all the electronics of the transmitters and receivers, with voltage ratings which are required in the standards applied to Secondary Stations.
- e) Supply can be permanent on the bus, in which case Alarm mode can be active.
- f) From 1 to 100 Secondary Stations can be connected in parallel on the bus.

The maximum number of non-energized Secondary Stations (simple or multiple) on the bus is 50.

The maximum number of energized Secondary Stations on the bus is 100.

The maximum number of devices associated to Secondary Stations (simple or multiple) on the bus is 50.

In case of a mix of Secondary Station types, Energized and non-energized, connected on a same bus, a rule gives the maximum number of each type:

if $N1$ is the number of non-energized multiple Secondary Stations on the bus,

if $N2$ is the number of non-energized simple Secondary Stations on the bus,

si N_3 est le nombre de Stations Secondaires alimentées sur le bus, les inégalités suivantes doivent être respectées:

- | | |
|--------------------------------|--|
| $2*(N_1 + N_2) + N_3 \leq 100$ | quand l'alimentation est intégrée dans la Station Primaire |
| $2*(N_1 + N_2) + N_3 \leq 98$ | quand l'alimentation est externe, connectée directement sur le bus |

- g) Une des Stations Secondaires peut rester accidentellement en basse impédance (mode transmission mais sans émission de "0"), sans interrompre la communication.
- h) La communication avec la Station Primaire se fait en utilisant un coupleur magnétique (un seul sur le bus).
- i) Le bus doit pouvoir supporter une connexion accidentelle au 230 V du secteur. La procédure de contrôle consiste à appliquer cinq fois successivement 250 V a.c. Chaque application dure 5 min, avec 5 s d'intervalle entre chaque application.

5.3.2 Caractéristiques du câble

Câble téléphonique intérieur de type:

- paire torsadée simple avec écran (aluminium) et conducteur de drain,
- conducteur: cuivre étamé de diamètre nominal compris entre 0,5 mm et 0,6 mm,
- isolant PVC.

Caractéristiques électriques:

Résistance de boucle en continu à 20 °C: 117 Ω/km à 192 Ω/km

En alternatif à 50 kHz, entre –15 °C et +45 °C:

- | | |
|--|-----------------------|
| a) résistance linéaire de boucle: | 154 Ω/km à 220 Ω/km |
| b) inductance linéaire de boucle: | 500 μH/km à 800 μH/km |
| c) capacité linéaire mutuelle: | 80 nF/km à 130 nF/km |
| d) facteur de perte de la capacité: | 5 % maximum |
| e) capacité déséquilibrée, conducteur-écran: | 5 % maximum |
| f) impédance caractéristique complexe: | 74 Ω à 115 Ω |
| g) déphasage linéaire à 50 kHz: | 150°/km maximum |

Les caractéristiques ci-dessus sont données pour une source symétrique isolée de l'écran du câble avec des impédances Z et Z' supérieures à 1 000 Ω à 50 kHz.

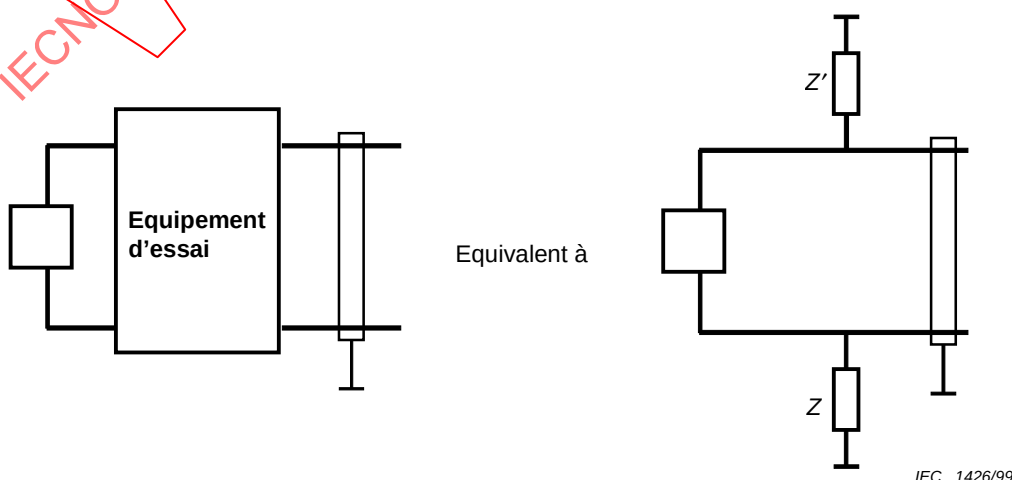


Figure 11 – Diagramme équivalent de l'équipement d'essai

if N_3 is the number of energized Secondary Stations on the bus,
then these inequations have to be respected:

$$2*(N_1 + N_2) + N_3 \leq 100 \quad \text{when power supply is integrated to the Primary Station}$$

$$2*(N_1 + N_2) + N_3 \leq 98 \quad \text{when power supply is external, connected directly on the bus}$$

- g) One of these Secondary Stations can accidentally stay in low impedance (transmission mode but without emission of "0" transmission), without interrupting the communication.
- h) Communication with the Primary Station is via a magnetic plug (one only).
- i) The bus shall withstand the accidental connection of the 230 V mains. Control procedure is to apply five times successively 250 V a.c. Each application lasts 5 min and 5 s between each application.

5.3.2 Cable characteristics

Indoor telephone cable of type:

- single twisted pair and screen (aluminum) with drain wire,
- conductor: solid tinned copper of 0,5 mm to 0,6 mm nominal diameter,
- insulation PVC.

Electrical characteristics:

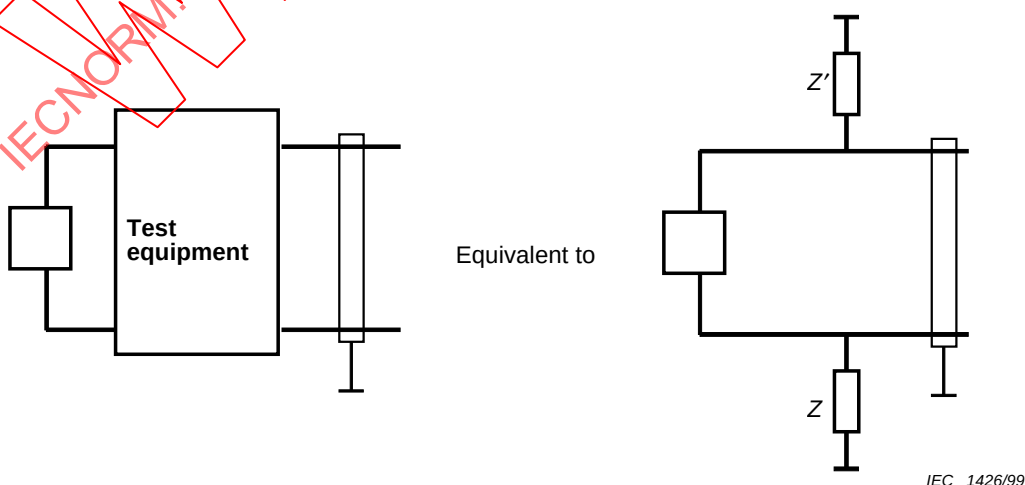
DC looped resistance at 20 °C:

117 Ω /km to 192 Ω /km

AC 50 kHz, between –15 °C to +45 °C:

- a) linear looped resistance: 154 Ω /km to 220 Ω /km
- b) linear looped inductivity: 500 μ H/km to 800 μ H/km
- c) linear mutual capacity: 80 nF/km to 130 nF/km
- d) loss factor of capacity: 5 % maximum
- e) capacity unbalance, wires to screen: 5 % maximum
- f) complex characteristic impedance: 74 Ω to 115 Ω
- g) linear phase shift (50 kHz): 150°/km maximum

The above characteristics are given for a symmetric source isolated from screen with the impedances Z and Z' greater than 1 000 Ω at 50 kHz.



IEC 1426/99

Figure 11 – Equivalent diagram of the test equipment

5.3.3 Raccordements

- a) Le raccordement des Stations Secondaires doit assurer la continuité du conducteur de drain (par exemple: les boîtiers de dérivation doivent être à trois circuits).
- b) Le conducteur de drain doit être relié en un point à la terre, si elle existe, ou à une référence de potentiel équivalente.
- c) Aucune impédance (autre que le câble lui-même) de moins de $1\ 000\ \Omega$ à 50 kHz, ne doit être connectée entre les conducteurs du bus et l'écran ou la terre.

NOTE – Lors de l'utilisation de câbles ne répondant pas exactement aux spécifications ci-dessus, il convient de tenir compte de ce qui suit:

- Un câble avec une capacité ou une résistance linéique plus importante a une longueur maximale plus petite. La longueur maximale varie approximativement comme l'inverse de la capacité ou de la résistance.
- Un câble avec une capacité ou une résistance linéique plus faible peut conduire à des surtensions sur les entrées d'un récepteur raccordé sur un bus long et vide. Ce problème peut être résolu en plaçant entre les conducteurs du bus, du côté opposé au coupleur magnétique, une résistance d'amortissement ($330\ \Omega$ à $1\ 000\ \Omega$, 0,25 W, en fonction de la surtension). Il convient de placer une capacité de 47 nF et de tension de claquage adaptée en série avec cette résistance, pour résister à une connexion accidentelle du bus au 230 V du secteur.

5.4 Couplage magnétique

5.4.1 Fonction

5.4.1.1 Couplage magnétique simple

Le couplage magnétique se compose d'un coupleur mobile côté Station Primaire et d'une embase fixe côté Station Secondaire.

Quand les deux parties sont accolées, le couplage magnétique permet l'échange bi-directionnel entre le HHU relié au coupleur et le bus connecté à l'embase.

Chaque partie consiste en un demi-transformateur à ferrite avec un entrefer dans le circuit magnétique.

Pour compenser les fortes inductances séries et les faibles impédances parallèles d'un tel transformateur, un condensateur d'accord et une résistance d'amortissement placés de chaque côté, convertit ce transformateur en un filtre passe-bande de quatrième ordre centré sur 50 kHz, avec un facteur de surtension Q inférieur à 3.

Cela permet l'utilisation d'une simple source générant un signal carré pour les transmissions, et élimine les transitoires de fréquence.

5.4.1.2 Couplage magnétique avec alimentation en énergie

En plus des caractéristiques précédentes, le couplage magnétique avec alimentation en énergie permet la transmission d'un signal entre 400 kHz et 600 kHz. La valeur par défaut est fixée à 500 kHz. Ce signal est redressé et filtré avant l'injection du signal continu sur le bus.

5.3.3 Wiring

- a) Connection of Secondary Stations shall ensure the continuity of the drain wire (e.g. three terminal distribution boxes).
- b) One point of the drain wire shall be connected to earth, if any, or to an equivalent reference potential.
- c) No impedance (except the cable), of less than $1\,000\ \Omega$ at 50 kHz, shall be connected between wires of the bus and screen or earth.

NOTE – For use of cables slightly outside the above specifications, the following should be noted:

- A cable with a higher linear capacity or resistance needs a lower length of wired cable. Ratio of length is approximately as the inverse ratio of linear capacity or resistance.
- A cable with a lower linear capacity or resistance could give overvoltage on receiver inputs for a long empty bus. This can be overcome by connecting between the wires of the bus, near the end opposite the magnetic plug, a damping resistor ($330\ \Omega$ to $1\,000\ \Omega$, 0,25 W, depending on the overvoltage ratio). To ensure that the bus withstands an accidental connection to 230 V mains, a 47 nF capacitor of proper voltage capability should be in series with this resistor.

5.4 Magnetic plug

5.4.1 Function

5.4.1.1 Simple magnetic plug

The magnetic plug consists of a mobile plug (primary) and a fixed socket (secondary).

When the two halves are joined, the magnetic plug transfers signals between the HHU connected to the plug and the bus connected to the socket, in both directions.

Each part consists of half a ferrite transformer with an air-gap in the magnetic circuit.

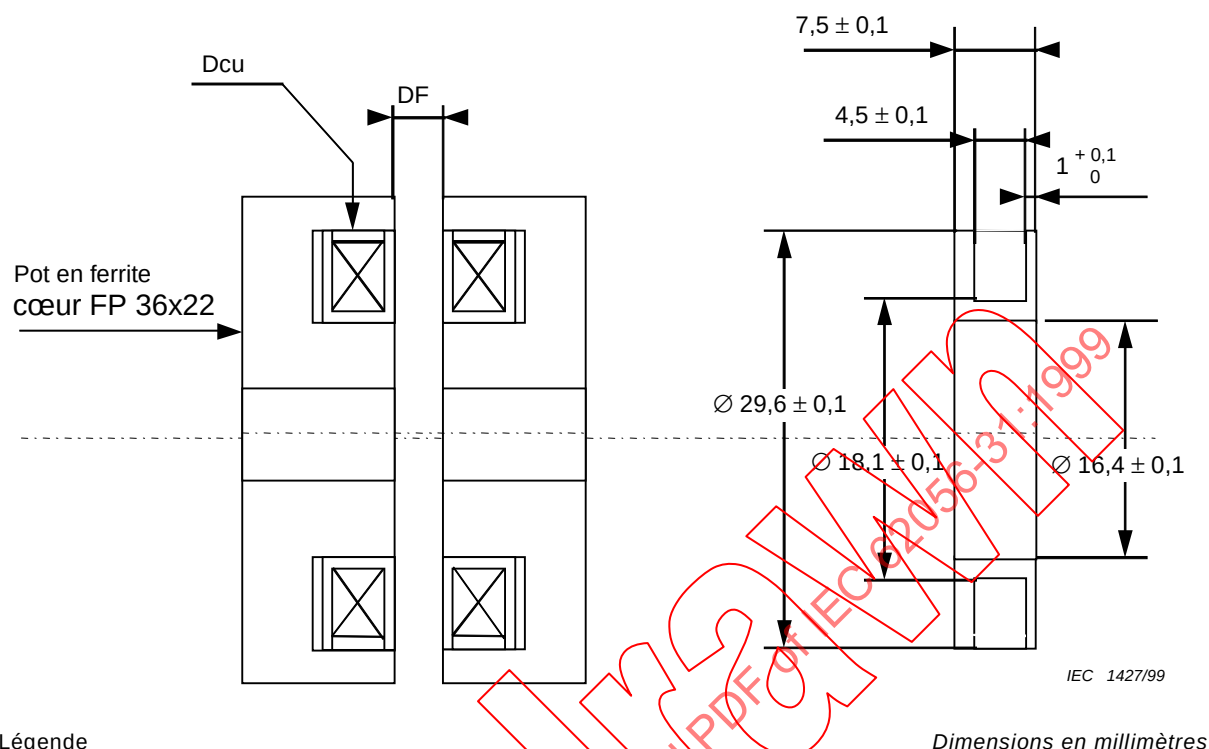
To compensate the high series inductance and the low parallel inductance of such a transformer, a resonating capacitor and a damping resistor on both sides convert this transformer into a fourth order band-pass filter centered near 50 kHz, with a Q factor <3 .

This allows the use of a simple square wave source for the transmission, and eliminates the frequency transients.

5.4.1.2 Magnetic plug with energy supply

In addition to the previous characteristics, the magnetic plug with energy supply allows the transmission between 400 kHz and 600 kHz signal. The default value is fixed at 500 kHz. This signal is rectified and filtered before injection of d.c. signal on the bus.

5.4.2 Caractéristiques mécaniques communes



Légende

- FP pot ferromagnétique
- DF distance entre les ferrites
- Dcu diamètre du cœur

Figure 12 – Pot de ferrite et bobine

Chacune des deux parties du couplage magnétique contient une bobine placée dans le cœur d'un demi-pot de ferrite, le tout intégré dans un boîtier en plastique robuste. Une fois réunis, les bobines et les cœurs de ferrite sont pratiquement coaxiaux et symétriques par rapport au plan médian, avec:

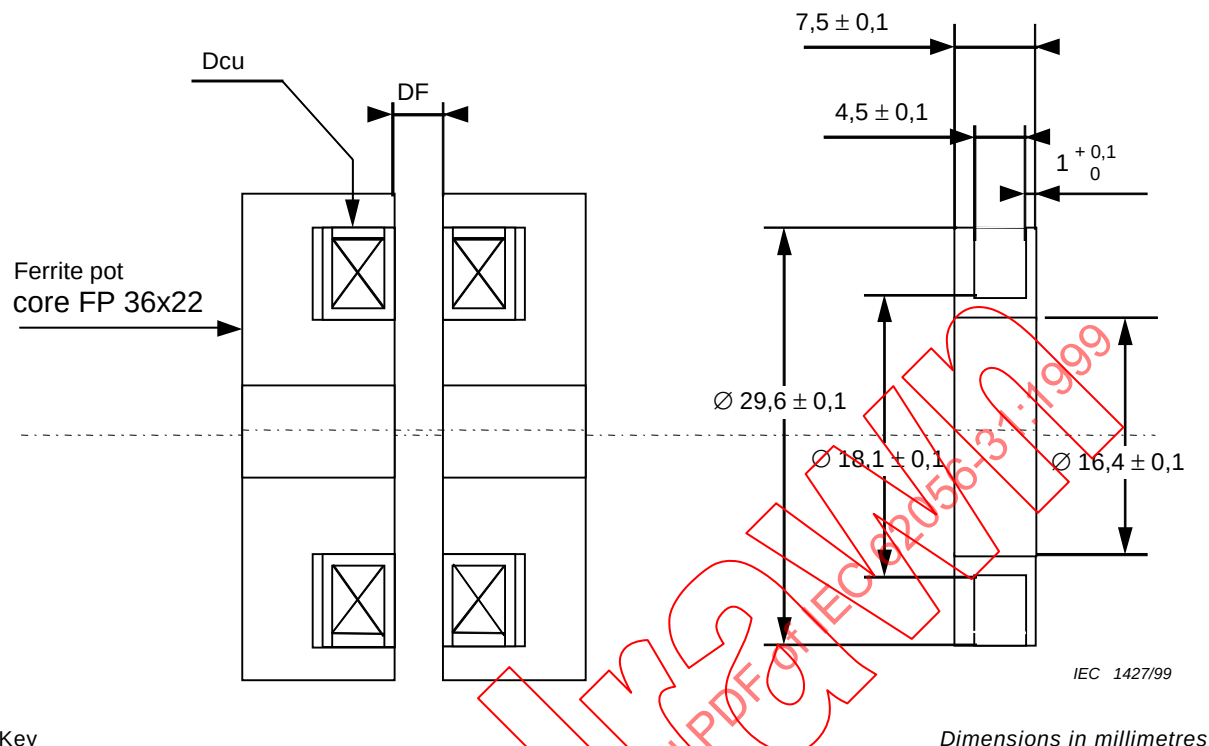
- entrefer dans le circuit magnétique $DF = 4,25 \text{ mm} \pm 0,25 \text{ mm}$;
- erreur maximale de centrage, près du point milieu: 0,25 mm:

Une fois assemblés et joints, les cœurs de ferrite et les bobines doivent être fixés ou maintenus très proches du plan médian. Par exemple, l'espace axial entre bobine et cœur dû à des tolérances dimensionnelles doit être situé côté face arrière de la bobine.

La ferrite est d'un type standard pour une fréquence inférieure à 100 kHz:

- Perméabilité initiale supérieure à 1 800;
- Tangente delta maximale à 100 kHz aux environs de 2 %;
- Pas de saturation due aux champs de fuite de l'accrochage magnétique (en pratique saturation autour de 0,4 T).

5.4.2 Common mechanical characteristics



Key

- FP ferromagnetic pot
- DF distance between ferrites
- Dcu diameter of core unit

Figure 12 – Ferrite pot and bobbin

Each half of the magnetic plug comprises a bobbin in a half pot ferrite core, enclosed in a robust plastic housing. When joined, the bobbins and ferrite cores are nearly coaxial and symmetric about the mid plane, with:

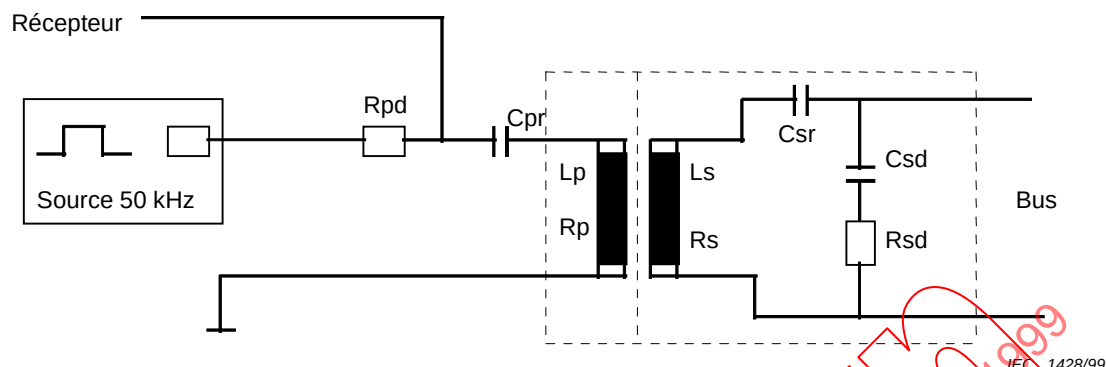
- magnetic air gap DF = 4,25 mm ± 0,25 mm;
- maximum error of coaxiality, near mid point: 0,25 mm.

When assembled and joined, both ferrite cores and bobbins shall be fixed or pushed closer to the mid plane. For example, the axial gap between bobbin and core, due to tolerances or sizes shall be at the rear side of the bobbin.

Ferrite is a standard type for a maximum frequency of less than 100 kHz:

- Initial permeability more than 1 800;
- Tan delta maximum at 100 kHz about 2 %;
- No saturation due to leakage fields (practically saturation comes around 0,4 T).

5.4.3 Diagramme électrique avec couplage simple



Légende

Rpd	résistance d'amortissement primaire
Cpr	capacité d'accord au primaire
Csr	capacité d'accord au secondaire
Csd	capacité d'amortissement au secondaire
Rsd	résistance d'amortissement secondaire
Lp	self du primaire
Ls	self du secondaire
Rp	résistance du primaire
Rs	résistance du secondaire

Figure 13 – Composants associés au couplage magnétique

5.4.3.1 Composants de l'embase, côté bus

Capacité série Csr accordée avec Ls.

Résistance d'amortissement en parallèle Rsd.

Capacité parallèle Csd en série avec Rsd pour assurer la protection contre un raccordement accidentel au 230 V du secteur.

5.4.3.2 Composants côté coupleur

Capacité série Cpr accordée avec Lp.

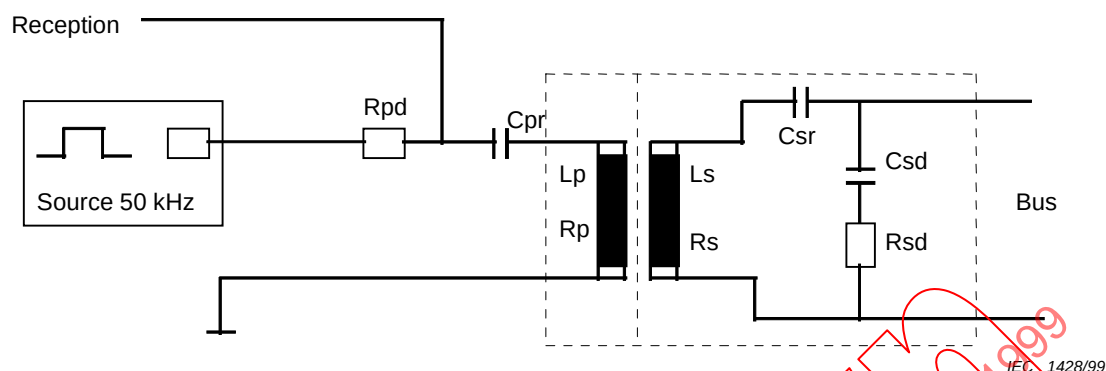
Résistance d'amortissement en série Rpd: (dépend de la source 50 kHz, et inclut toutes les résistances en série (à 50 kHz), telles que le cordon flexible, les connecteurs, etc.).

Source 50 kHz qui peut être une source générant un signal carré.

Circuit récepteur, démodulation et mise en forme, connecté au nœud Cpr, Rpd (par exemple, le circuit redresseur et le circuit de seuil).

En mode réception, l'impédance de sortie de la source 50 kHz doit être maintenue à moins de quelques ohms, même pour les faibles niveaux de tension, pour assurer l'amortissement du circuit primaire.

5.4.3 Electrical Block diagram with simple plug



Key

Rpd	primary damping resistor
Cpr	primary resonance capacitor
Csr	secondary resonance capacitor
Csd	secondary damping capacitor
Rsd	secondary damping resistor
Lp	primary inductance
Ls	secondary inductance
Rp	primary resistance
Rs	secondary resistance

Figure 13 – Associated components of the magnetic plug

5.4.3.1 Associated components in socket, bus side

Serial capacitor Csr resonating with Ls.

Parallel damping resistor Rsd.

Parallel capacitor Csd in series with Rsd to ensure immunity to accidental connection to 230 V mains.

5.4.3.2 Associated components, plug side

Serial capacitor Cpr resonating with Lp.

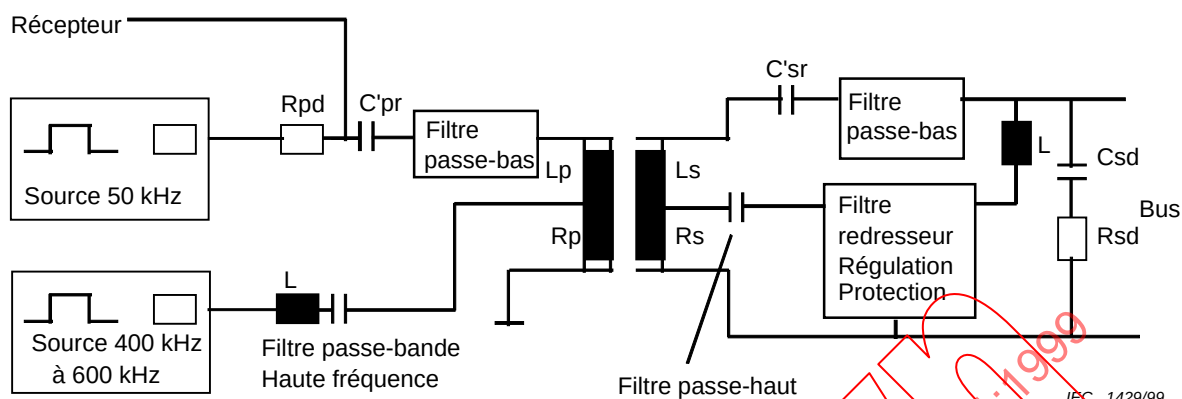
Serial damping resistor Rpd: (depending on the 50 kHz source, and including all series resistances (at 50 kHz), such as flexible cord, connector, etc.).

A 50 kHz source which can be a square-wave source.

Receiver circuit, demodulation and squaring, connected to the Cpr Rpd node (e.g. full wave rectifier and threshold circuit).

In receiving mode, the output impedance of the 50 kHz source shall be maintained at less for a few ohms, even for low voltage, to ensure damping of the primary circuit.

5.4.4 Diagramme électrique avec couplage alimenté



Légende

Rpd	résistance d'amortissement primaire	Lp	self du primaire
Cpr	capacité d'accord au primaire	Ls	self du secondaire
Csr	capacité d'accord au secondaire	Rp	résistance du primaire
Csd	capacité d'amortissement au secondaire	Rs	résistance du secondaire
Rsd	résistance d'amortissement secondaire		

Figure 14 – Composants associés au couplage alimenté

La partie 50 kHz fonctionne comme le circuit de couplage simple, mis à part:

- une atténuation supplémentaire, dans les deux sens, due aux filtres passe-bas. L'équipement récepteur de la source 50 kHz doit prendre en compte cette atténuation.
- C'pr et C'sr diffèrent légèrement de Cpr et Csr à cause de l'impédance du filtre passe-bas à 50 kHz.

5.5 Spécifications fonctionnelles de l'émetteur de la Station Primaire (pour le signal 50 kHz)

L'émetteur côté Station Primaire est composé des circuits d'émission de la Station Primaire et du coupleur magnétique.

Le signal émis sur le bus doit rester dans les limites spécifiées (voir figure 5) dans toute la gamme de température, avec:

- Tev1 = 750 μ s
Tev0 = 750 μ s
- Vevh1 = 0,25 V

Ceci concerne uniquement les signaux avec une fréquence $1 \text{ kHz} \leq f \leq 1 \text{ MHz}$.

5.4.4 Electrical Block Diagram with energy supply plug

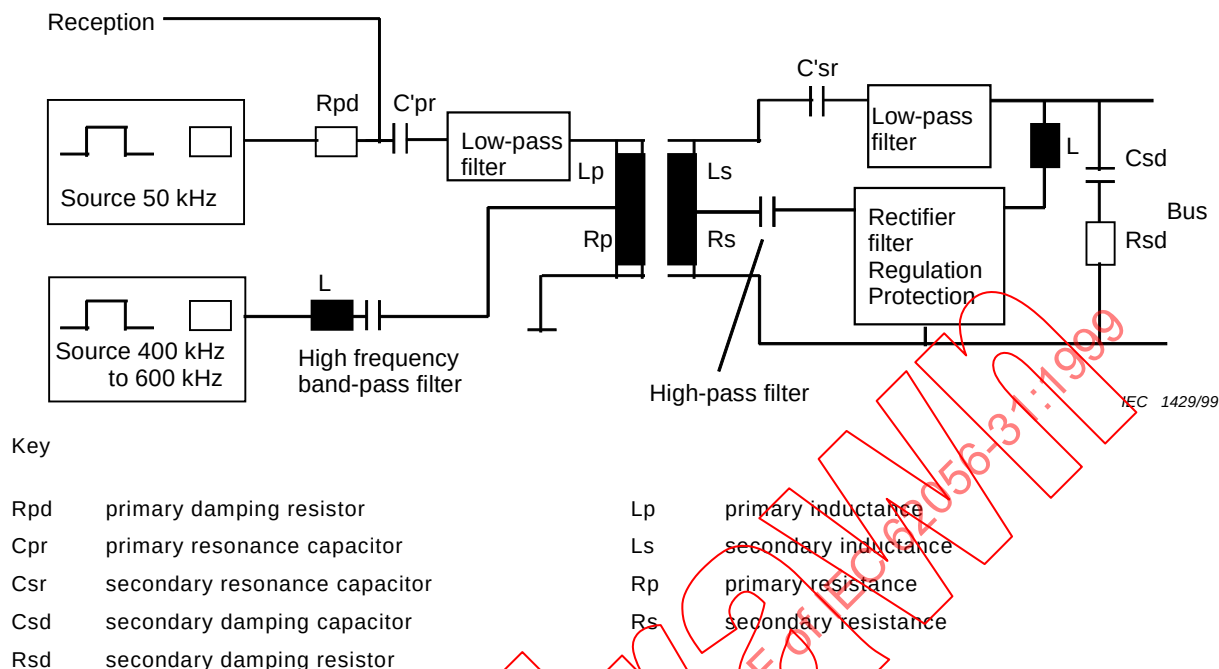


Figure 14 – Associated components of the energy supply plug

The 50 kHz part works like the simple plug circuit, except for:

- extra attenuation, both ways, due to the low-pass filters. The 50 kHz source receiver device shall take this attenuation into account.
- C'pr and C'sr differ slightly from Cpr and Csr due to the low-pass filter impedance at 50 kHz.

5.5 Functional specifications of Primary Station transmitter (for 50 kHz signal)

The Primary Station transmitter is the assembly of a Primary Station transmitting source and the magnetic plug.

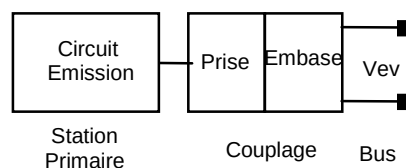
The signal transmitted at the bus outputs shall fit into the limits (see figure 5), of the whole temperature range, with:

- Tev1 = 750 μ s
Tev0 = 750 μ s
- Vevh1 = 0,25 V

This concerns only signals with frequency $1 \text{ kHz} \leq f \leq 1 \text{ MHz}$.

Avec les bornes de sortie sur le bus en circuit ouvert:

- c) $V_{evl0} = 5,8 \text{ V}$
- d) $V_{evh0} = 7,5 \text{ V}$

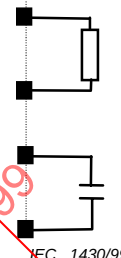


Avec une résistance de 100Ω à la place du bus:

- e) $V_{evl0} = 4 \text{ V}$
- f) $V_{evh0} = 4,8 \text{ V}$

Avec une capacité de $31,8 \text{ nF}$ à la place du bus:

- g) $V_{evl0} = 5,2 \text{ V}$
- h) $V_{evh0} = 6,5 \text{ V}$



De plus, il convient que les bornes de sortie soient en circuit ouvert, avec une résistance de 100Ω ou une capacité de $31,8 \text{ nF}$.

- i) Le bruit émis sur les sorties du bus, dans toutes les conditions et à toutes les fréquences jusqu'à 1 MHz , ne doit pas dépasser 10 mV crête entre 1 kHz et 1 MHz , après extinction des transitoires;
- j) Les parasites dus au passage du mode émission au mode réception et inversement, ne doivent pas dépasser $0,25 \text{ V}$ crête, dans toutes les conditions.

Ces valeurs sont à respecter avec un entrefer dans le circuit magnétique $DF = (4,25 \text{ mm} \pm 0,25 \text{ mm}) \pm_{0,15}^{0,15} \text{ mm}$.

5.6 Spécifications fonctionnelles du récepteur de la Station Primaire (pour le signal 50 kHz)

Le récepteur côté Station Primaire est composé des circuits de réception de la Station Primaire (démodulation et mise en forme) et du coupleur magnétique.

Le récepteur doit fonctionner correctement (taux de répétition de trame $< 10^{-5}$) pour un signal d'entrée ayant une porteuse sinusoïdale conforme aux caractéristiques spécifiées précédemment, et dans toute la gamme de température.

Ce signal doit être appliqué aux bornes de raccordement au bus du coupleur, à travers les impédances séries spécifiées, dans les conditions suivantes:

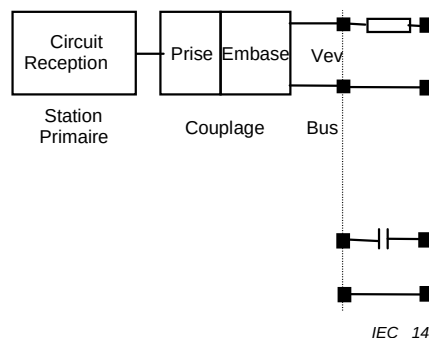
- a) $T_{ev0} = 700 \mu\text{s}$
 $T_{ev1} = 700 \mu\text{s}$

A travers une résistance de 100Ω :

- b) $V_{evh1} = 0,25 \text{ V}$
- c) $V_{evl0} = 0,7 \text{ V}$
- d) $V_{evh0} = 3,2 \text{ V}$

A travers une capacité de $31,8 \text{ nF}$:

- e) $V_{evh1} = 0,20 \text{ V}$
- f) $V_{evl0} = 0,55 \text{ V}$
- g) $V_{evh0} = 2,5 \text{ V}$



Open circuit at bus output:

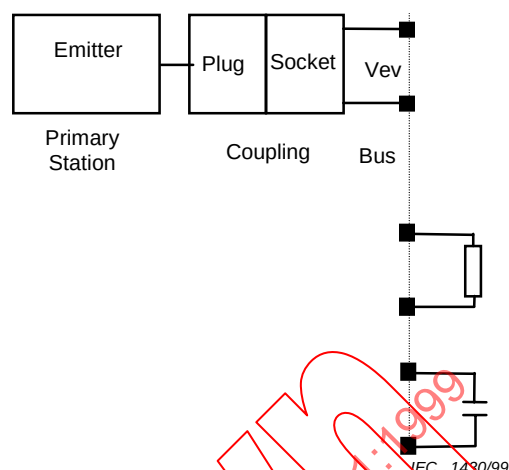
- c) $V_{evl0} = 5,8 \text{ V}$
- d) $V_{evh0} = 7,5 \text{ V}$

With a resistor of 100Ω in place of the bus:

- e) $V_{evl0} = 4 \text{ V}$
- f) $V_{evh0} = 4,8 \text{ V}$

With a capacity of $31,8 \text{ nF}$ in place of the bus:

- g) $V_{evl0} = 5,2 \text{ V}$
- h) $V_{evh0} = 6,5 \text{ V}$



In addition, the output terminals should be open-circuit or matched to a 100Ω resistance or a capacitance of $31,8 \text{ nF}$.

- i) The noise transmitted at the bus output, in all conditions and at all frequencies up to 1 MHz , after extinction of transients, shall not exceed 10 mV peak between 1 kHz and 1 MHz
- j) The overvoltage spike due to the switch from transmission mode to reception mode, or the reverse, shall not exceed $0,25 \text{ V}$ peak, in all conditions.

These values have to be respected with a magnetic air gap $DF = (4,25 \text{ mm} \pm 0,25 \text{ mm}) \pm_{0,15}^{0,15} \text{ mm}$.

5.6 Functional specifications of Primary Station receiver (for 50 kHz signal)

The Primary Station receiver is the assembly of a Primary Station receiving circuit (demodulation and squaring) and the magnetic plug.

The receiver shall function correctly (defined as frame repetition rate $< 10^{-5}$) for a sinusoidal input signal whose characteristics are defined above and for the complete temperature range.

The signal shall be applied to the bus terminals of the magnetic plug through the serial impedances given below:

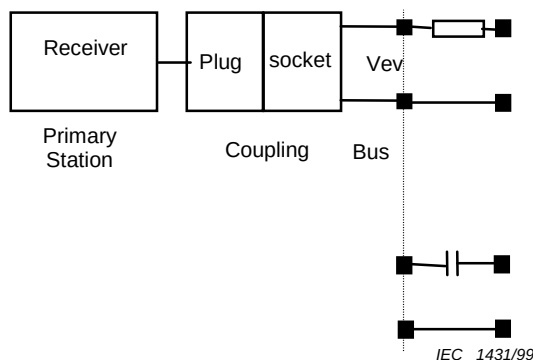
- a) $T_{ev0} = 700 \mu\text{s}$
 $T_{ev1} = 700 \mu\text{s}$

Through 100Ω :

- b) $V_{evh1} = 0,25 \text{ V}$
- c) $V_{evl0} = 0,7 \text{ V}$
- d) $V_{evh0} = 3,2 \text{ V}$

Through $31,8 \text{ nF}$:

- e) $V_{evh1} = 0,20 \text{ V}$
- f) $V_{evl0} = 0,55 \text{ V}$
- g) $V_{evh0} = 2,5 \text{ V}$



De plus, le récepteur dans les conditions b) ($100\ \Omega$ en série) doit fonctionner correctement (taux de répétition de trame $< 10^{-5}$) avec:

- h) un signal sinusoïdal permanent de valeur crête de 0,1 V pour les fréquences de 1 kHz à 1 MHz;
- i) une impulsion de 20 V durant 5 μ s;
- j) une impulsion de 3,5 V durant 200 μ s.

5.7 Spécifications fonctionnelles de l'émetteur de la Station Secondaire (pour le signal 50 kHz)

Le signal émis sur les sorties du bus doit rester dans les limites spécifiées, dans toute la gamme de température, avec:

- a) $T_{ev0} = 750\ \mu$ s
 $T_{ev1} = 750\ \mu$ s
- b) $V_{evh1} = 0,1\ V$

Ceci concerne uniquement les signaux avec une fréquence $1\ kHz \leq f \leq 1\ MHz$.

Avec une résistance de $100\ \Omega$ à la place du bus:

- c) $V_{evl0} = 1,2\ V$
- d) $V_{evh0} = 1,8\ V$

Avec un condensateur de 31,8 nF à la place du bus, le signal de sortie mesuré aux bornes d'une résistance de $1\ \Omega$ placée en série avec le condensateur, et le résultat multiplié par 100:

- e) $V_{evl0} = 1,5\ V$
- f) $V_{evh0} = 2,5\ V$

De plus, que les bornes de sortie soient raccordées à une résistance de $100\ \Omega$ ou à une capacité de 31,8 nF:

- g) les parasites dus au passage du mode émission au mode réception et inversement, ne doivent pas dépasser 0,75 V crête, dans toutes les conditions;
- h) le bruit émis sur les sorties du bus, dans toutes les conditions et à toutes les fréquences jusqu'à 1 MHz, ne doit pas dépasser 10 mV crête entre 1 kHz et 1 MHz, après extinction des transitoires.

De plus:

- i) le courant de court-circuit maximal est inférieur à 26 mA crête à 50 kHz;
- j) les circuits d'émission doivent supporter un court-circuit permanent et la connexion au 230 V du secteur aux bornes de sortie sur le bus;
- k) la valeur maximale de la capacité en mode commun entre les bornes du bus et les autres bornes de l'appareil comportant la station secondaire est fixée à 15 pF.

5.8 Spécifications fonctionnelles du récepteur de la Station Secondaire (pour le signal 50 kHz)

Le récepteur doit fonctionner correctement (taux de répétition de trame $< 10^{-5}$) pour un signal d'entrée ayant une porteuse sinusoïdale conforme aux caractéristiques spécifiées précédemment, et dans toute la gamme de température, avec:

- a) $T_{ev0} = 700\ \mu$ s
 $T_{ev1} = 700\ \mu$ s