IEC 61158-6-11

Edition 1.0 2007-12

INTERNATIONAL STANDARD

Industrial communication networks – Fieldbus specifications –
Part 6-11: Application layer protocol specification – Type 11 elements

IEC 61158-6-11:2007(E)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

About the IEC

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

About IEC publications

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

- Catalogue of IEC publications: www.iec.ch/searchpub
The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub
Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org
The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv
If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00

IEC 61158-6-11

Edition 1.0 2007-12

INTERNATIONAL STANDARD

Industrial communication networks – Fieldbus specifications – Part 6-11: Application layer protocol specification – Type 11 elements

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE V

ICS 35.100.70; 25.040.40

ISBN 2-8318-9486-7

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

____

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 6-11: Application layer protocol specification – Type 11 elements

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as “IEC Publication(s)”). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

NOTE Use of some of the associated protocol types is restricted by their intellectual-property-right holders. In all cases, the commitment to limited release of intellectual-property-rights made by the holders of those rights permits a particular data-link layer protocol type to be used with physical layer and application layer protocols in Type combinations as specified explicitly in the IEC 61784 series. Use of the various protocol types in other combinations may require permission from their respective intellectual-property-right holders.

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this standard may involve the use of patents concerning as follows:

TOSHIBA has the patent applications listed below:

– US Publication Number 6711131 and its counterpart patents in other countries
– US Publication Number 5414813 and its counterpart patents in other countries
– US Publication Number 4930121 and its counterpart patents in other countries

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

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 IEC. Information may be obtained from:

Toshiba Corporation
1-1, Shibaura 1-Chome
Minato-ku Tokyo 105-8001, Japan

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.

International standard IEC 61158-6-11 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

This first edition and its companion parts of the IEC 61158-6 subseries cancel and replace IEC 61158-6:2003. This edition of this part constitutes a technical addition. This part and its Type 11 companion parts also cancel and replace IEC/PAS 62406, published in 2005.

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

FDIS	Report on voting
65C/476/FDIS	65C/487/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.

This edition of IEC 61158-6 includes the following significant changes from the previous edition:

a) deletion of the former Type 6 fieldbus for lack of market relevance;

b) addition of new types of fieldbuses;

c) partition of part 6 of the third edition into multiple parts numbered -6-2, -6-3, …

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

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

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

NOTE The revision of this standard will be synchronized with the other parts of the IEC 61158 series.

The list of all the parts of the IEC 61158 series, under the general title *Industrial communication networks – Fieldbus specifications*, can be found on the IEC web site.

INTRODUCTION

This part of IEC 61158 is one of a series produced to facilitate the interconnection of automation system components. It is related to other standards in the set as defined by the "three-layer" fieldbus reference model described in IEC/TR 61158-1.

The application protocol provides the application service by making use of the services available from the data-link or other immediately lower layer. The primary aim of this standard is to provide a set of rules for communication expressed in terms of the procedures to be carried out by peer application entities (AEs) at the time of communication. These rules for communication are intended to provide a sound basis for development in order to serve a variety of purposes:

- as a guide for implementors and designers;
- for use in the testing and procurement of equipment;
- as part of an agreement for the admittance of systems into the open systems environment;
- as a refinement to the understanding of time-critical communications within OSI.

This standard is concerned, in particular, with the communication and interworking of sensors, effectors and other automation devices. By using this standard together with other standards positioned within the OSI or fieldbus reference models, otherwise incompatible systems may work together in any combination.

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 6-11: Application layer protocol specification – Type 11 elements

1 Scope

1.1 General

The fieldbus Application Layer (FAL) provides user programs with a means to access the fieldbus communication environment. In this respect, the FAL can be viewed as a “window between corresponding application programs.”

This standard provides common elements for basic time-critical and non-time-critical messaging communications between application programs in an automation environment and material specific to Type 11 fieldbus. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

This standard defines in an abstract way the externally visible behavior provided by the different Types of the fieldbus Application Layer in terms of

a) the abstract syntax defining the application layer protocol data units conveyed between communicating application entities,
b) the transfer syntax defining the application layer protocol data units conveyed between communicating application entities,
c) the application context state machine defining the application service behavior visible between communicating application entities; and
d) the application relationship state machines defining the communication behavior visible between communicating application entities; and.

The purpose of this standard is to define the protocol provided to

1) define the wire-representation of the service primitives defined in IEC 61158-5-11, and
2) define the externally visible behavior associated with their transfer.

This standard specify the protocol of the IEC fieldbus Application Layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498) and the OSI Application Layer Structure (ISO/IEC 9545).

FAL services and protocols are provided by FAL application-entities (AE) contained within the application processes. The FAL AE is composed of a set of object-oriented Application Service Elements (ASEs) and a Layer Management Entity (LME) that manages the AE. The ASEs provide communication services that operate on a set of related application process object (APO) classes. One of the FAL ASEs is a management ASE that provides a common set of services for the management of the instances of FAL classes.

Although these services specify, from the perspective of applications, how request and responses are issued and delivered, they do not include a specification of what the requesting and responding applications are to do with them. That is, the behavioral aspects of the applications are not specified; only a definition of what requests and responses they can send/receive is specified. This permits greater flexibility to the FAL users in standardizing such object behavior. In addition to these services, some supporting services are also defined in this standard to provide access to the FAL to control certain aspects of its operation.

1.2 Specifications

The principal objective of this standard is to specify the syntax and behavior of the application layer protocol that conveys the application layer services defined in IEC 61158-5-11.

A secondary objective is to provide migration paths from previously-existing industrial communications protocols. It is this latter objective which gives rise to the diversity of protocols standardized in parts of the IEC 61158-6 series.

1.3 Conformance

This standard does not specify individual implementations or products, nor does it constrain the implementations of application layer entities within industrial automation systems.

There is no conformance of equipment to the application layer service definition standard. Instead, conformance is achieved through implementation of this application layer protocol specification.

2 Normative references

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

IEC 60559, *Binary floating-point arithmetic for microprocessor systems*

IEC 61158-3-11, *Industrial communication networks – Fieldbus specifications – Part 3-11: Data-link layer service definition – Type 11 elements*

IEC 61158-5-11, *Industrial communication networks – Fieldbus specifications – Part 5-11: Application layer service definition – Type 11 elements*

IEC 61784-2, *Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3*

ISO/IEC 7498-1, *Information technology – Open Systems Interconnection – Basic Reference Model — Part 1: The Basic Model*

ISO/IEC 8822, *Information technology – Open Systems Interconnection – Presentation service definition*

ISO/IEC 8824, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1)*

ISO/IEC 8825, *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*

ISO/IEC 9545, *Information technology – Open Systems Interconnection – Application Layer structure*

3 Terms, definitions, symbols, abbreviations and conventions

3.1 Introduction

For the purposes of this documents, the followings apply.

3.2 Terms and definitions from other ISO/IEC standards

3.2.1 Terms and definitions from ISO/IEC 7498-1

a) application entity
b) application process
c) application protocol data unit
d) application service element
e) application entity invocation
f) application process invocation
g) application transaction
h) real open system
i) transfer syntax

3.2.2 Terms and definitions from ISO/IEC 8822

a) abstract syntax
b) presentation context

3.2.3 Terms and definitions from ISO/IEC 9545

a) application-association
b) application-context
c) application context name
d) application-entity-invocation
e) application-entity-type
f) application-process-invocation
g) application-process-type
h) application-service-element
i) application control service element

3.2.4 Terms and definitions from ISO/IEC 8824

a) object identifier
b) type
c) value
d) simple type
e) structured type
f) component type
g) tag
h) Boolean type
i) true
j) false
k) integer type
l) bitstring type
m) octetstring type
n) null type
o) sequence type
p) sequence of type
q) choice type
r) tagged type
s) any type
t) module
u) production

3.2.5 Terms and definitions from ISO/IEC 8825

a) encoding (of a data value)
b) data value
c) identifier octets (the singular form is used in this standard)
d) length octet(s) (both singular and plural forms are used in this standard)
e) contents octets

3.3 Terms and definitions from IEC/TR 61158-1

a) application relationship
b) conveyance path
c) client
d) dedicated AR
e) dynamic AR
f) error class
g) error code
h) name
i) numeric identifier
j) peer
k) pre-defined AR endpoint
l) pre-established AR endpoint
m) publisher
n) subscriber
o) server

3.4 Other terms and definitions

NOTE The following definitions may apply to all the types except if the same item is used with different definitions in a specific type. In this case the latter has precedence.

The following terms and definitions are used in this series of standards.

3.4.1
called
service user or a service provider that receives an indication primitive or a request APDU

3.4.2
calling
service user or a service provider that initiates a request primitive or a request APDU

3.4.3
common memory
virtual common memory over the Type 11 fieldbus, which is shared with the nodes participating in the Type 11 fieldbus and is primarily used for the real-time communications by the TCC data service

3.4.4
interoperability
capability of User Layer entities to perform coordinated and cooperative operations using the services of the FAL

3.4.5
management information
network accessible information that supports the management of the Fieldbus environment

3.4.6
receiving

service user that receives a confirmed primitive or an unconfirmed primitive, or a service provider that receives a confirmed APDU or an unconfirmed APDU

3.4.7
resource
resource is a processing or information capability of a subsystem

3.4.8
sending
service user that sends a confirmed primitive or an unconfirmed primitive, or a service provider that sends a confirmed APDU or an unconfirmed APDU

3.5 Abbreviations and symbols

AE	Application Entity
AE-I	Application Entity Invocation
AL	Application Layer
AP	Application Process
Ap_	Prefix for Data types defined for AP ASE
Ar_	Prefix for Data types defined for AR ASE
APDU	Application Protocol Data Unit
AR	Application Relationship
AREP	Application Relationship End Point
ASE	Application Service Element
ASN.1	Abstract Syntax Notation One
BCD	Binary Coded Decimal
BER	Basic Encoding Rule
BNU-PEC	Buffered Network-Scheduled Uni-directional Pre-Established Connection
CM	Common Memory
cnf	confirmation primitive
Dl_	Prefix for Data types defined for data-link layer types
DL	Data-link
DLC	Data-link Connection
DLCEP	Data-link Connection End Point
DLPDU	Data-link Protocol Data Unit
DLSAP	Data-link Service Access Point
DLSDU	Data-link Service Data Unit
Dt_	Prefix for Data types defined for Data type ASE
Err	Error (used to indicate an APDU type)
Er_	Prefix for Error types defined
Ev_	Prefix for Data types defined for Event ASE
FAL	Fieldbus Application Layer
Fi_	Prefix for Data types defined for Function Invocation ASE
FIFO	First In First Out
Gn_	Prefix for Data types defined for general use
ID	Identifier
IEC	International Electrotechnical Commission
in	input primitive
ind	indication primitive
ISO	International Organization for Standardization
LAS	Link Active Scheduler
Lr_	Prefix for Data types defined for Load Region ASE
lsb	least significant bit
Mn_	Prefix for Data types defined for Management ASE
msb	most significant bit
out	output primitive
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
PICS	Protocol Implementation Conformance Statement
QoS	Quality Of Service
Req	Request (used to indicate an APDU type)
req	request primitive
Rsp	Response (used to indicate an APDU type)
rsp	response primitive

SAP	Service Access Point
SDU	Service Data Unit
TCC	Time-critical cyclic
ToS	Type Of Service
Vr_	Prefix for Data types defined for Variable ASE

3.6 Conventions

3.6.1 Conventions for class definitions

The data-link layer mapping definitions are described using templates. Each template consists of a list of attributes for the class. The general form of the template is defined in IEC 61158-5.

3.6.2 Abstract syntax conventions

When the "optionalParametersMap" parameter is used, a bit number which corresponds to each OPTIONAL or DEFAULT production is given as a comment.

3.6.3 Conventions used in state machines

The state machines are described in Table 1.

Table 1 – Conventions used for state machines

#	Current state	Event / condition => action	Next state
Name of this transition	The current state to which this state transition applies	Events or conditions that trigger this state transaction. => The actions that are taken when the above events or conditions are met. The actions are always indented below events or conditions	The next state after the actions in this transition is taken

The conventions used in the state machines are as follows:

:= Value of an item on the left is replaced by value of an item on the right. If an item on the right is a parameter, it comes from the primitive shown as an input event.

xxx A parameter name.

Example:
Identifier := reason
means value of a 'reason' parameter is assigned to a parameter called 'Identifier.'

"xxx" Indicates fixed value.

Example:
Identifier := "abc"
means value "abc" is assigned to a parameter named 'Identifier.'

= A logical condition to indicate an item on the left is equal to an item on the right.

< A logical condition to indicate an item on the left is less than the item on the right.

> A logical condition to indicate an item on the left is greater than the item on the right.

<> A logical condition to indicate an item on the left is not equal to an item on the right.

&& Logical "AND"

|| Logical "OR"

This construct allows the execution of a sequence of actions in a loop within one transition. The loop is executed for all values from start_value to end_value.

```
Example:
  for (Identifier := start_value to end_value)
     actions
  endfor
```

This construct allows the execution of alternative actions depending on some condition (which might be the value of some identifier or the outcome of a previous action) within one transition.

```
Example:
  If (condition)
    actions
  else
    actions
  endif
```

Readers are strongly recommended to refer to the subclauses for the AREP attribute definitions, the local functions, and the FAL-PDU definitions to understand protocol machines. It is assumed that readers have sufficient knowledge of these definitions, and they are used without further explanations.

4 FAL syntax description

4.1 Concept

This standard specifies the Application layer protocol of the Type 11 essential for the ISO/IEC 8802-3-based Time-critical control network (TCnet), which is one of the communication networks for the Real-Time Ethernet(RTE) defined in IEC 61784-2 and is referred to as RTE-TCnet hereafter.

This standard meets the industrial automation market objective of providing predictable time deterministic and reliable time-critical data transfer and means, which allow co-existence with non-time-critical data transfer over the ISO/IEC 8802-3 series communications medium, for support of cooperation and synchronization between automation processes on field devices in a real-time application system. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty.

This standard specifies the part of the protocol set of the RTE-TCnet communication profile and/or of one or more communication profiles related to a common family of the RTE-TCnet. The RTE-TCnet communication profile, shown in Figure 1 as one of the profile sets, is based on the 7 layer OSI Basic Reference model. For regular ISO/IEC 8802-3 based applications the upper layers mapped over the data-link layer is in the ordinary way; on the other hand, for time-critical applications with Common Memory running in parallel, the specific application layer for RTE-TCnet is specified. The data-link layer for RTE-TCnet has the extension, but is compliant to the ISO/IEC 8802-3 MAC protocol in order to provide both services for time-critical communications and common memory applications respectively.

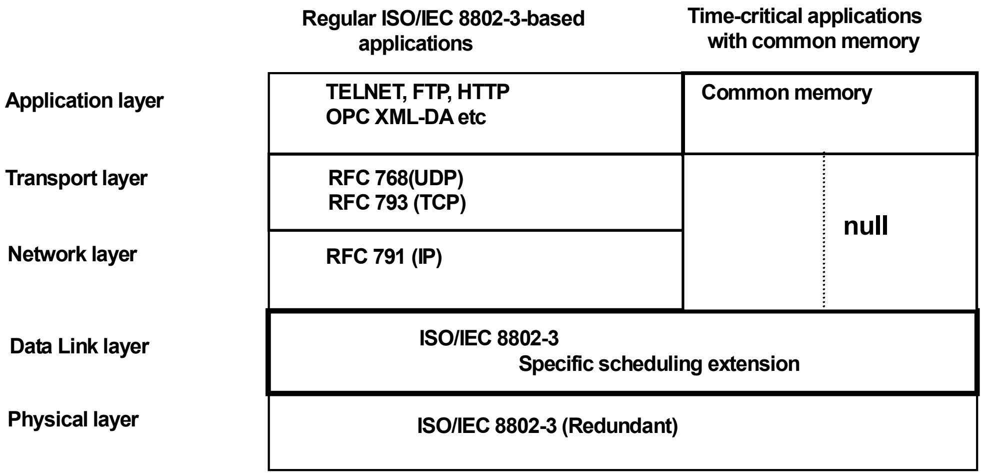

Figure 1 – RTE-TCnet communication profile

This standard specifies the data-link protocol as the essential parts of the RTE-TCnet profile, which are the extension part of the ISO/IEC 8802-3 based data-link layer and the Application layer exploiting the services of the data-link layer immediately below, in terms of the "three-layer" Fieldbus Reference Model which is based in part on the OSI Basic Reference Model. Other part of the RTE-TCnet profile is not in the scope of this document.

4.1.1 Field of applications

In industrial control systems, several kinds of field devices such as drives, sensors and actuators, programmable controllers, distributed control systems and human-machine interface devices are required to be connected with control networks. The process control data and the state data is transferred among these field devices in the system and the communications between these field devices requires simplicity in application programming and to be executed with adequate response time. In most industrial automation systems such as food, water, sewage, paper and steel, including a rolling mill, the control network is required to provide time-critical response capability for their application, as required in ISO/TR 13283 for time-critical communications architectures.

Plant production may be compromised due to errors, which could be introduced to the control system if the network does not provide a time-critical response. Therefore the following characteristics are required for a time-critical control network.

- – A deterministic response time between the control device nodes.
- – Ability to share process data seamlessly across the control system.

The RTE-TCnet is applicable to such industrial automation environment, in which time-critical communications is primarily required. The term "time-critical" is used to represent the presence of a time window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time-window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

4.2 General

FAL Syntax description of the RTE-TCnet defines unconfirmed send service and consists of the three parts as follows, FalArHeader, InvokeID and Unconfirmed Service Request.

4.3 FAL-AR PDU abstract syntax

4.3.1 Top level definition

FalArPDU ::= UnconfirmedSend-CommandPDU

4.3.2 Unconfirmed send service

```
UnconfirmedSend-CommandPDU ::= SEQUENCE {
  FalArHeader,
  InvokeID OPTIONAL,
  UnconfirmedServiceRequest
}
```

4.4 Abstract syntax of PDU body

4.4.1 FalArHeader

```
FalArHeader ::= Unsigned8 {
  -- bit 8      FAL Protocol Specifier     (Always 1)
  -- bit 7-4    Protocol Identifier        (Identifiers abstract syntax revision, and encoding rules)
  -- bit 3      Protocol Specific bit   (Reserved for each protocol to use)
  -- bit 2-1    PDU Identifier          (Identifies a PDU type within a Protocol Identifier)
}
```

4.4.2 InvokeID

```
InvokeID ::= Unsigned8
```

4.4.3 Unconfirmed PDUs

```
UnconfirmedServiceRequest ::= SEQUENCE {
    CMArep,   -- Block number
    CMData    -- content of CM segment
}
```

4.4.4 CMArep

```
CMArep ::= Unsigned16          -- DLCEP address
```

4.4.5 CMData

```
CMData ::= SEQUENCE {
    wlen  Unsigned16,       -- CM word length
    data  SEQUENCE {     ANY {    -- CM content
          BitString8, BitString16, BitString32,
          Integer16, Integer32,
          Unsigned16, Unsigned32,
          Floating32,
          OctetString2, OctetString4,
          VisibleString2, VisibleString4,
          BinaryTime0, BinaryTime1, BinaryTime2, BinaryTime3, BinaryTime4, BinaryTime5,
          BinaryTime6, BinaryTime7
          }
      }
 }
```

4.5 Data type

BitString8, BitString16, BitString32, Integer16, Integer32, Unsigned16, Unsigned32, Floating32, OctetString2, OctetString4, VisibleString2, VisibleString4, BinaryTime0, BinaryTime1, BinaryTime2, BinaryTime3, BinaryTime4, BinaryTime5, BinaryTime6.

5 Transfer syntax

5.1 Overview and FAL header

All the FAL PDUs shall have the common PDU-header called FalArHeader. The FalArHeader identifies abstract syntax, transfer syntax, and each of the PDUs. Table 2 defines how this header shall be used.

Table 2 – FAL header

Bit position of the FalArHeader			Abstract syntax	Encoding rule	PDU type	Revision
8	7 6 5 4 3	2 1				
1	1 1 1 1 1	10	ASN.1	RTE-TCnet	UnconfirmedSendPDU	Revision1
NOTE All other definitions are reserved.						

5.2 Encoding rule

5.2.1 Overview

The Encoding Rule of the RTE-TCnet is a preferable encoding rule that is compatible with existing standards. The FAL-PDUs encoded with the TER(Traditional Encoding Rule) shall have a uniform format. The FAL-PDUs shall consist of two major parts, the “APDU Header” part and the “APDU Body” part as shown in Figure 2.

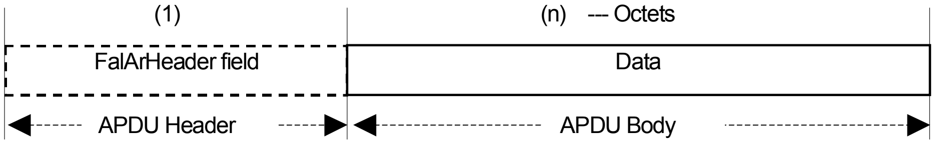

Figure 2 – APDU overview

5.2.2 APDU header encoding

The APDU Header part is always present in all APDUs which conform to this specification. It consists of one field: the FalArHeader Field. Refer to 4.4.1 for the encoding rule of the FalArHeader field.

5.2.3 APDU body encoding

The Encoding Rule of FAL is based on the terms and definitions of the ISO/IEC 8825, and consists of the three encoding components given below. For time-critical and using fixed length data, Identifier octet and Length octets like TER (Traditional Encoding Rule) do not exist.

- Identifier octet
- Length octets
- Contents octets

NOTE Identification Octet and Content Length Octets do not exist in RTE-TCnet.

5.2.4 Encoding of simple variable

5.2.4.1 Encoding of a Boolean value

a) The encoding of a Boolean value shall be primitive;

b) the Identifier octet and Length octet(s) shall not be present;

c) the Contents octets shall consist of a single octet;

d) if the Boolean value is FALSE, the Contents octets shall be 0 (zero);

e) if the Boolean value is TRUE, the Contents octets shall be 0xFF.

5.2.4.2 Encoding of a fixed length Integer value

a) The encoding of a fixed-length Integer value of Integer8, Integer16 and Integer32 types shall be primitive;

b) the Contents octets shall consist of exactly one, two or four octets, respectively;

c) the Identifier octet and the Length octet(s) shall not be present;

d) the Contents octets shall be a two's complement binary number equal to the integer value, and consist of bits 8 to 1 of the first octet, followed by bits 8 to 1 of the second octet, followed by bits 8 to 1 of each octet in turn up to and including the last octet of the Contents octets;

   1) the value of a two's complement binary number is derived by numbering the bits in the Content octets, starting with bit 1 of the first octet and ending the numbering with bit 8 of the last octet;

   2) each bit is assigned a numerical value of 2N-1, where N is its position in the above numbering sequence;

   3) the value of the two's complement binary number is obtained by adding the numerical values assigned to each bit for those bits which are set to one, excluding bit 8 of the last octet, and then reducing the value by the numerical value assigned to bit 8 of the last octet if that bit is set to one.

5.2.5 Encoding of a fixed length Unsigned value

a) The encoding of a fixed-length Integer value of Unsigned8, Unsigned16 and Unsigned32 types shall be primitive, and the Contents octets shall consist of exactly one, two or four octets, respectively;

b) the Identifier octet and Length octet(s) shall not be present;

c) the Contents octets shall be a binary number equal to the Unsigned value, and consist of bits 8 to 1 of the first octet, followed by bits 8 to 1 of the second octet, followed by bits 8 to 1 of each octet in turn up to and including the last octet of the Contents octets;

   1) the value of binary number is derived by numbering the bits in the Content octets, starting with bit 1 of the first octet as bit zero and ending the numbering with bit 8 of the last octet;

   2) each bit is assigned a numerical value of 2^{N-1}, where N is its position in the above numbering sequence;

   3) the value of the binary number is obtained by adding the numerical values assigned to each bit for those bits which are set to one.

5.2.6 Encoding of a Floating Point value

a) The encoding of a Floating32 value shall be primitive, and the Contents octets shall consist of exactly four octets;

b) the Identifier octet and Length octet(s) shall not be present;

NOTE The sign is encoded in bit 8 of the first octet. It is followed by the exponent starting from bit 7 of the first octet, and then the mantissa starting from bit 7 of the second octet for Floating32 and from bit 4 of the second octet for Floating64.

c) the Contents octets shall contain floating point values defined in conformance with IEC 60559.

5.2.7 Encoding of a fixed length BitString value

a) The encoding of a fixed-length Bitstring value of BitString8, BitString16, and BitString32 types shall be primitive, and the Contents octets shall consist of exactly one, two or four octets, respectively;

b) the Identifier Octet and Length Octet(s) shall not be present;

c) BitString value, commencing with the first bit and proceeding to the trailing bit, shall be placed in bits 8 to 1 of the first octet, followed by bits 8 to 1 of the second octet, followed by bits 8 to 1 of each octet up to and including the last octet of the Contents Octets.

5.2.8 Encoding of a fixed length Octet String value

a) The encoding of a fixed-length OctetString value of OctetString2, and OctetString4 types shall be primitive, and the Contents octets shall consist of exactly two, or four octets respectively;

b) the Identifier octet and Length octet(s) shall not be present;

c) the Contents octets shall be equal in value to the octets in the data value, in the order they appear in the data value, and with the most significant bit of an octet of the data value aligned with the most significant bit of an octet of the Contents octets.

5.2.9 Encoding of a fixed length Visible String value

a) The encoding of a fixed-length VisibleString value of VisibleString2, and VisibleString4 types shall be primitive, and the Contents octets shall consist of exactly two, or four octets respectively;

b) the Identifier octet and Length octet(s) shall not be present;

c) the Contents octets shall be equal in value to the octets in the data value, in the order they appear in the data value, and with the most significant bit of an octet of the data value aligned with the most significant bit of an octet of the Contents octets.

5.2.10 Encoding of BinaryTime value

a) The encoding of a BinaryTime0, BinaryTime1, BinaryTime2, BinaryTime3, BinaryTime4, BinaryTime5, BinaryTime6, and BinaryTime7 value shall be primitive;

b) the Identifier octet and Length octet(s) shall not be present;

c) the Contents octets shall be a binary number equal to the binary time value, and consisting of bits 8 to 1 of the first octet, followed by bits 8 to 1 of the second octet, followed by bits 8 to 1 of each octet in turn up to and including the last octet of the Contents octets;

   1) the Contents octets of a BinaryTime0, BinaryTime1, BinaryTime2, and BinaryTime3 value shall consist of two octets;

   2) the Contents octets of a BinaryTime4, BinaryTime5, BinaryTime6, and BinaryTime7 value shall consist of four octets.

NOTE The value of the granularity of each BinaryTime type is defined in IEC 61158-5 subseries.

5.3 Encoding of structured types

5.3.1 General

When the structured type is also encoded, the identifier or length of the structure are not provided in RTE-TCnet.

5.3.2 Encoding of a SEQUENCE value

The SEQUENCE type is comparable to a record. It represents a collection of user data of the same or of different Data types.

A SEQUENCE type value may contain a simple variable. or a further structured variable as its components. If a SEQUENCE type contains another structured type value, it shall be counted as a single component even if it contains several components.

6 FAL protocol state machines structures

6.1 Overview

As shown in Figure 3, the protocol machine of FAL consists of three, the FAL service protocol machine (FSPM), the application relationship protocol machine (ARPM) and the data-link layer mapping protocol machine (DMPM).

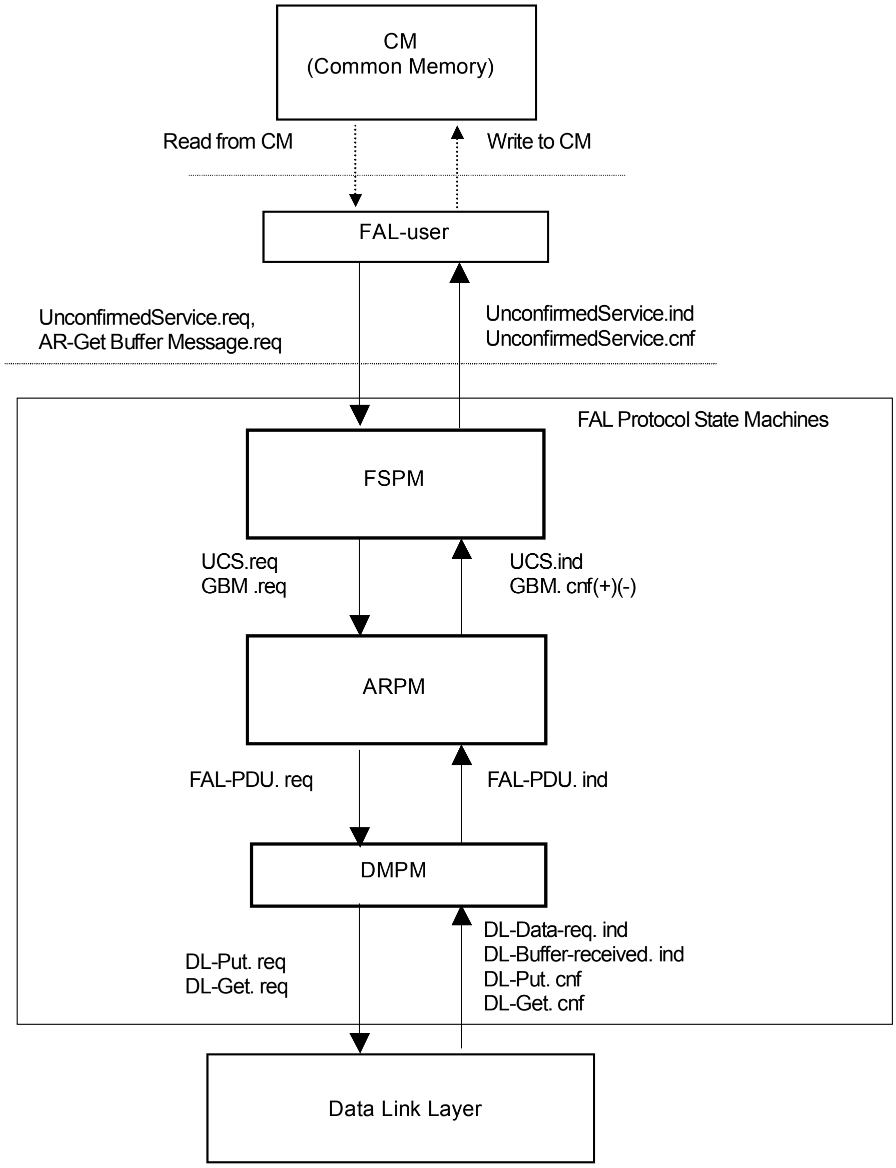

Figure 3 – Relationship between FSPM, ARPM, DMPM and external physical CM

7 FAL service protocol machine (FSPM)

7.1 General

The FAL Service Protocol Machine (FSPM) is common to all the AREP types. Only applicable primitives are different among different AREP types. It has one state called "ACTIVE".

NOTE Although now present, the type of AREP is only one.

7.2 Primitives definitions

7.2.1 Primitive exchanged between FAL user and FSPM

The primitive exchanged between the FAL user and the FSPM are described in Table 3 and Table 4.

Table 3 – Primitives issued by FAL user to FSPM

Primitive names	Source	Associated parameters	Function
UCS.req	FAL user	Arep-id, Data	This is an FAL internal primitive used to convey an Unconfirmed Send (UCS) request primitive from the FAL user to the FSPM
GBM.req	FAL user	Arep-id	This is an FAL internal primitive used to convey a Get-Buffered-Message (GBM) request primitive from the FAL user to the FSPM

Table 4 – Primitives issued by FSPM to FAL user

Primitive names	Source	Associated parameters	Function
UCS.ind	FSPM	Arep-id, Data	This is an FAL internal primitive used to convey an Unconfirmed Send (UCS) indication primitive from the FSPM to the FAL user
GBM.cnf(+)	FSPM	Arep-id, Data	This is an FAL internal primitive used to convey a Get-Buffered-Message (GBM) positive confirmation from the FSPM to the FAL user
GBM.cnf(-)	FSPM	Arep-id	This is an FAL internal primitive used to convey a Get-Buffered-Message (GBM) negative confirmation from the FSPM to the FAL user
FSTS.ind	FSPM	Arep-id, Reported-status	This is an FAL internal primitive used to convey a FAL-Status (FSTS) indication primitive from the FSPM to the FAL user

7.2.2 Parameters of FAL user /FSPM

All the parameters used in the primitives exchanged between the FAL user and the FSPM are identical to those defined in the "Operational Service" subclause.

7.3 FSPM state tables

7.3.1 General

The FSPM state machines are described in Figure 4, Table 5 and Table 6.

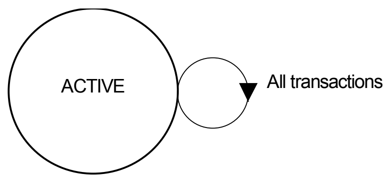

Figure 4 – State transition diagram of FSPM

Table 5 – FSPM state table – sender transactions

#	Current state	Event or condition => action	Next state

S1	ACTIVE	UCS.req
		&& SelectArep (Arep-id) = "True"
		=>
		UCS.req {
		user-data := Data
		}
			ACTIVE

S2	ACTIVE	GBM.req
		&& SelectArep (Arep-id) = "True"
		=>
		GBM.req {
		}
			ACTIVE

NOTE 1 A primitive parameter in the FSPM sender state machine is sent to an appropriate ARPM that is selected by the FSPM using the SelectArep function. NOTE 2 If the SelectArep function return the value of False, it is a local matter to report such instance and the FSPM does not generate any primitive for the ARPM.			

Table 6 – FSPM state table – receiver transactions

#	Current state	Event or condition => action	Next state

R1	ACTIVE	UCS.ind
		=>
		UCS.ind {
		Arep-id := arep-id,
		Data := user-data
		}
			ACTIVE

R2	ACTIVE	GBM.cnf(+)
		=>
		GBM.cnf (+) {
		Arep-id := arep,
		Data := user-data
		}
			ACTIVE

R3	ACTIVE	GBM.cnf(-)
		=>
		GBM.cnf (-) {
		Arep-id := arep-id
		}
			ACTIVE

R4	ACTIVE	FAL-STS.ind
		=>
		FSTS.ind {
		Arep-id := arep-id,
		Reported-status := reported-status
		}
			ACTIVE

7.3.2 Functions

The function used in this state machine is as shown in Table 7.

Table 7 – Function SelectArep

Name	SelectArep	Used in	FSPM
input		Output	
Arep-id		True || False	
Function			
Looks for the AREP entry that is specified by the Arep-id parameter. True means the AREP exists.			

8 Application relationship protocol machine (ARPM)

8.1 General

The RTE-TCnet define a Application Relation (AR) and their associated ARPM, which is the Buffered network-scheduled unidirectional - pre-established connection (BNU-PEC).

8.2 Primitive definitions

8.2.1 Primitives exchanged between ARPM and FSPM

Table 8 and Table 9 list the primitives exchanged between the FSPM and the ARPM.

Table 8 – Primitives issued by FSPM to ARPM

Primitive names	Source	Associated parameters	Function
UCS.req	FSPM	user-data	This is an FAL internal primitive used to convey Unconfirmed Send (UCS) request primitive from the FSPM to the ARPM
GBM.req	FSPM	(none)	This is an FAL internal primitive used to convey a Get-Buffered-Message (GBM) request primitive from the FSPM to the ARPM

Table 9 – Primitives issued by ARPM to FSPM

Primitive names	Source	Associated parameters	Function
UCS.ind	ARPM	arep-id, user-data	This is an FAL internal primitive used to convey an Unconfirmed Send (UCS) indication primitive from the ARPM to the FSPM
GBM.cnf(+)	ARPM	arep-id	This is an FAL internal primitive used to convey a Get-Buffered-Message (GBM) positive confirmation from the ARPM to the FSPM
GBM.cnf(-)	ARPM	arep-id	This is an FAL internal primitive used to convey a Get-Buffered-Message (GBM) negative confirmation from the ARPM to the FSPM
FSTS.ind	ARPM	arep-id, reported-status	This is an FAL internal primitive used to convey a FAL-Status (FSTS) indication primitive from the ARPM to the FSPM

8.2.2 Parameters of FSPM/ARPM primitives

The parameters used with the primitives exchanged between the FSPM and the ARPM are described in Table 10.

Table 10 – Parameters used with primitives exchanged between FSPM and ARPM

Parameter name	Description
arep-id	This parameter is used to unambiguously identify an instance of the AREP that has issued a primitive. A means for such identification is not specified by this specification
user-data	This parameter conveys a FAL-User data
identifier	This parameter conveys value that is used for the Identifier parameter
reason	This parameter conveys value that is used for the Reason-Code parameter
status	This parameter conveys value that is used for the Status parameter
reported-status	This parameter conveys a data-link layer event status

8.3 DLL mapping of BNU-PEC AREP class

This subclause describes the mapping of the BNU-PEC AREP class to the RTE data-link layer. It does not redefine the DLCEP attributes that are or will be defined in the data-link layer specification; rather, it defines how they are used by this AR class.

The DLL Mapping attributes, their permitted values and the DLL services used with the BNU-PEC AREP class are defined in this subclause.

CLASS: BNU-PEC
PARENT CLASS: BufferedNetworkScheduledUnidirectionalPre-EstablishedConnectionAREP
ATTRIBUTES:
1 (m) KeyAttribute: LocalDlcepIdentifier

2 (m) Attribute: Role (Publisher, Subscriber)

DLL SERVICES:

1 (m) OpsService: DL-Put

2 (m) OpsService: DL-Get

3 (m) OpsService: DL-Buffer-received

4 (m) OpsService: DL-Data-req

8.3.1 Attributes

8.3.1.1 LocalDlcepIdentifier

This attribute specifies the local DLCEP-identifier of a DL-Put or DL-Get primitive and thus it identifies the DLCEP.

8.3.1.2 Role

This attribute specifies the role of this AREP. The value of "Publisher" indicates that this AREP is used as a publisher. The value of "Subscriber" indicates that this AREP is used as a subscriber.

8.3.2 DLL services

Refer to 9.4, for the DLL service descriptions.

8.4 BNU-PEC ARPM states machine

8.4.1 BNU-PEC ARPM state

The defined states together with their descriptions of the BNU-PEC ARPM are listed in Table 11 and Figure 5.

Table 11 – BNU-PEC state descriptions

State name	Description
OPEN	The BNU-PEC in the OPEN state is defined and capable of sending or receiving FAL-PDUs

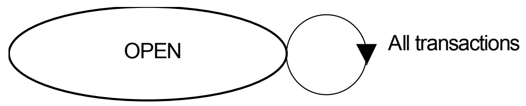

Figure 5 – State transition diagram of the BNU-PEC

8.4.2 BNU-PEC ARPM state table

Table 12 and Table 13 define the BNU-PEC state machines.

Table 12 – BNU-PEC ARPM state table – sender transactions

#	Current state	Event or condition => action	Next state

S1	OPEN	UCS.req
		&& Role = "Publisher"
		=>
		FAL-PDU.req {
		dmpm-service-name := "DMPM_Put_req",
		arep-id := GetArepId (),
		dlsdu := BuildFAL-PDU (
		fal-pdu-name := "UCS.PDU",
		fal-data := user-data)
		}
			OPEN

S2	OPEN	GBM.req
		&& Role = "Subscriber"
		=>
		FAL-PDU.req {
		dmpm-service-name := "DMPM_Get_req" ,
		arep-id := GetArepId ()
		}
			OPEN

Table 13 – BNU-PEC ARPM state table – receiver transactions

#	Current state	Event or condition => action	Next state

R1	OPEN	FAL-PDU.ind
		&& Role = "Subscriber"
		&& dmpm-service-name = "DMPM_Buffer_Received_ind"
		&& FAL_Pdu_Type (fal-pdu) = "UCS-PDU"
		=>
		UCS.ind {
		arep-id := GetArepID (),
		user-data := fal-pdu
		}
			OPEN

R2	OPEN	FAL-PDU.ind
		&& Role = "Subscriber"
		&& dmpm-service-name = "DMPM_Get_cnf "
		&& reason = "success "
		=>
		GBM.cnf (+) {
		arep-id := GetArepID (),
		user-data := fal-pdu
		}
			OPEN

R3	OPEN	FAL-PDU.ind
		&& Role = "Subscriber"
		&& dmpm-service-name = "DMPM_Get_cnf "
		&& reason <> "success "
		=>
		GBM.cnf(-) {
		arep-id := GetArepID ()
		}
			OPEN

R4	OPEN	FAL-PDU.ind
		&& Role = "Publisher"
		&& dmpm-service-name = "DMPM_DATA_REQ_ind"
		=>
		FSTS.ind {
		arep-id := GetArepID (),
		reported-status := "DATA-REQ"
		}
			OPEN

R5	OPEN	ErrorToARPM => (no action token, see note) NOTE It is a local matter to report this error status to network management entities.	OPEN

8.4.3 Functions used by BNU-PEC ARPM

Table 14, Table 15 and Table 16 define the function used by this service machine.

Table 14 – Function GetArepId ()

Name	GetArepId()	Used in	ARPM
input		Output	
(none)		AREP Identifier	
Function			
Returns a value that can unambiguously identify the current AREP.			

Table 15 – Function BuildFAL-PDU

Name	BuildFAL-PDU	Used in	ARPM
input		Output	
fal-pdu-name, fal-data		dlsdu	
Function			
Builds an FAL-PDU out of the parameters given as input variables.			

Table 16 – Function FAL_Pdu_Type

Name	FAL_Pdu_Type	Used in	ARPM
input		Output	
fal-pdu		One of the FAL-PDU types defined in the subclause 9.4.	
Function			
Decodes the FAL-PDU that is conveyed in the fal-pdu parameter and retrieves one of the FAL-PDU types.			

9 DLL mapping protocol machine (DMPM)

9.1 Overview

The DLL Mapping Protocol Machine is common to all the AREP types. Only applicable primitives are different among different AREP types.

NOTE Although now present, the type of AREP is only one.

a) Remarks about DLCEP-identifier

The RTE-TCnet data-link layer specification defines local DLCEP-identifiers to distinguish the pre-defined connections pre-established by a local means. This DLCEP-identifier is key property of FAL ARPMs, and included a parameter as DMPM primitive.

b) Remarks about DLSDU-length

The RTE-TCnet data-link layer specification defines the parameter of DLSDU-length to distinguish the end of each DLSDU.

In the specification of the RTE-TCnet data-link layer, the DLSDU-length is defined to partition the end of the DLSDU. In actual implementation, usage of this parameter depends on implementation. In this DMPM, DLSDU-length is not included.

c) Remarks about configuration and initialization of the data-link layer

The RTE-TCnet data-link layer specification defines the configuration service to set the resource of layer or the class of connection.

In the specification of the RTE-TCnet data-link layer, the configuration service is defined to set the class of resource and connection.

9.2 Primitive definitions

9.2.1 Primitives exchanged between DMPM and ARPM

Table 17 and Table 18 list the service primitives between the ARPM and the DMPM.

Table 17 – Primitives issued by ARPM to DMPM

Primitive names	Source	Associated parameters	Function
FAL-PDU.req	ARPM	dmpm-service-name, arep-id, local-dlcep-identifier, reason, response_request, dlsdu	This primitive is used to request the DMPM to transfer an FAL-PDU. It passes the FAL-PDU to the DMPM as a DLSDU. It also carries some of the data-link layer parameters that are referenced there

Table 18 – Primitives issued by DMPM to ARPM

Primitive names	Source	Associated parameters	Function
FAL-PDU.ind	DMPM	dmpm-service-name, reason, response_request, fal-pdu	This primitive is used to pass an FAL-PDU received as a data-link layer service data unit to a designated ARPM. It also carries some of the data-link layer parameters that are referenced in the ARPM
ErrorToARPM	DMPM	reason	This primitive is used to convey selected communication errors reported by the data-link layer to a designated ARPM

9.2.2 Parameters of ARPM/DMPM primitives

The parameters used with the primitives exchanged between the ARPM and the DMPM are described in Table 19.

Table 19 – Parameters used with primitives exchanged between ARPM and DMPM

Parameter name	Description
arep-id	This parameter carries a local identifier to specify the associated AR instance
dmpm-service-name	This parameter conveys a DMPM pseudo-service name or s data-link layer service name. Possible value are represented as DMPM_XXXX_yyy
dls_user_data	This parameter conveys the value of the dl-dls-user-data parameter
dlsdu	This parameter conveys the value of the dl-dls-user-data parameter
fal-pdu	This parameter conveys the value of the dl-dls-user-data parameter
local-dlcep-identifier	This parameter conveys the value of the Requesting pre-established AREP parameter. and the value of the Responding pre-established AREP parameter
reason	This parameter conveys the value of the dl_reason parameter
status	This parameter conveys the value of the dl-status parameter

9.2.3 Primitives exchanges between data-link layer and DMPM

Table 20 summarizes the primitives exchanged between the DLL and the DMPM.

Table 20 – Primitives exchanged between data-link layer and DMPM

Primitive names	Source	Associated parameters
DL-Buffer-received.ind	Data-link layer	dmpm-service-name, reason, response_request, fal-pdu
DL-Data-req.ind	Data-link layer	reason
DL-Get.req (out)	Data-link layer	dl_dlcep_identifier, dl_dlsdu, dl-status
DL-Put.req (out)	Data-link layer	dl_dlcep_identifier, dl-status
DL-Get.req (in)	DMPM	dl_dlcep_identifier,
DL-Put.req (in)	DMPM	dl_dlcep_identifier, dl-dls-user-data

9.2.4 Parameters of DMPM/data-link layer primitives

The parameters used with the primitives exchanged between the DMPM and the data-link layer are identical to those defined in IEC 61158-3-11. They are prefixed by "dl_" to indicate that they are used by the FAL.

9.3 DLL mapping protocol machine (DMPM)

9.3.1 DMPM States

The defined state of the DMPM together with its description are listed in Table 21 and Figure 6.

Table 21 – DMPM state descriptions

State name	Description
ACTIVE	The DMPM in the ACTIVE state is ready to transmit or receive primitives to or from the data-link layer and the ARPM

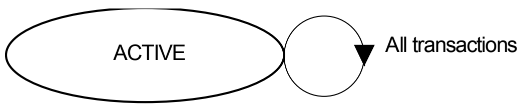

Figure 6 – State transition diagram of DMPM

9.3.2 DMPM state table

The DMPM state machines are defined in Table 22 and Table 23.

NOTE Although each primitive contains all the available parameters, only those applicable to particular ARPM are relevant.

Table 22 – DMPM state table – sender transactions

#	Current state	Event or condition => action	Next state

S1	ACTIVE	FAL-PDU.req
		&& dmpm-service-name = "DMPM_Put_req"
		=>
		PickArep (arep-id),
		
		DL-Put.req (in) {
		dl-local-dlcep-identifier := local-dlcep-identifier,
		dl-dls-user-data := dlsdu
		}
		
		DL-Put.req (out) -- immediate response
		&& dl-status = "success"
		=>
		FAL-PDU.ind {
		dmpm-service-name := "DMPM_Put_cnf" ,
		reason := dl-status
		}
		
		DL-Put.req (out) -- immediate response
		&& dl-status <> "success"
		=>
		FAL-PDU.ind {
		dmpm-service-name := "DMPM_Put_cnf" ,
		reason := dl-status
		}
		
		ErrorToARPM {
		reason := dl-status
		}
			ACTIVE

S2	ACTIVE	FAL-PDU.req
		&& dmpm-service-name = "DMPM_Get_req"
		=>
		PickArep (arep-id),
		
		DL-Get.req (in) {
		dl-local-dlcep-identifier := local-dlcep-identifier
		}
		
		DL-Get.req (out) -- immediate response
		=>
		FAL-PDU.ind {
		dmpm-service-name := "DMPM_Get_cnf" ,
		reason := dl-status
		}
			ACTIVE

Table 23 – DMPM state table – receiver transactions

#	Current state	Event or condition => action	Next state
R1	ACTIVE	DL-Buffer-received.ind && FindAREP () = "False" => (no action token)	ACTIVE

R2	ACTIVE	DL-Buffer-received.ind
		&& FindAREP () = "True"
		=>
		DL-Get.req (in) {
		dl-local-dlcep-identifier := local-dlcep-identifier
		}
		DL-Get.req (out) -- immediate response
		&& dl-status = "success"
		=>
		FAL-PDU.ind {
		dmpm-service-name := "DMPM_Buffer_Received_ind" ,
		fal-pdu := dl-dls-user-data,
		reason := dl-status
		}
		DL-Get.req (out) -- immediate response
		&& dl-status <> "success"
		=>
		FAL-PDU.ind {
		dmpm-service-name := "DMPM_Get_cnf" ,
		reason := dl-status
		}
		ErrorToAPM {
		reason := dl-status
		}
			ACTIVE

R3	ACTIVE	DL-Data-req.ind && FindAREP () = "False" => (no action token)	ACTIVE

R4	ACTIVE	DL-Data-req.ind
		&& FindAREP () = "True"
		=>
		FAL-PDU.ind {
		dmpm-service-name := "DMPM_DATA_REQ"
		}
			ACTIVE

9.3.3 Function used by DMPM

Table 24 and Table 25 define the function used by the DMPM.

Table 24 – Function PickArep

Name	PickArep	Used in	DMPM
input		Output	
arep-id		(all the attributes of the specified AREP)	
Function	Selects the attributes for the AREP specified by the arep-id parameter. After this function is executed, the attributes of the selected AREP are available to the state machine.		

Table 25 – Function FindAREP

Name	FindAREP	Used in	DMPM
input		Output	
local-dlcep-identifier		True || False	
Function	Identifies the AREP that shall be bound with an active DMPM. True means the AREP exists. If it does, this function also returns a means to send a DMPM primitive to that AREP.		

9.4 Data-link layer service selection

9.4.1 General

This subclause briefly describes the data-link layer service utilized by the FAL. These data-link layer services are fully defined in IEC 61158-3-11.

NOTE The FAL assumes that a resource, such as buffers, is set up prior to any operations of FAL protocol machines by a local means. Therefore , this service is not listed in this subclause.

9.4.2 DL-Put

This service is used to copy an FAL-PDU to a buffer. It refers to the Put service.

9.4.3 DL-Get

This service is used to read an FAL-PDU from a buffer. It refers to the Get service.

9.4.4 DL-Buffer-received

The DL-Buffer-received service is used to inform the FAL of new update on the specified receive buffer.

9.4.5 DL-Data-req

The DL-Data-req service is used to inform the FAL that the specified buffer content became update timing.

Bibliography

IEC 61131-3, *Programmable controllers – Part 3: Programming languages*

IEC/TR 61158-1 (Ed.2.0), *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61158-4-11, *Industrial communication networks – Fieldbus specifications – Part 4-11: Data-link layer protocol specification – Type 11 elements*

ISO/IEC 7498-3, *Information technology – Open Systems Interconnection – Basic Reference Model — Part 3: Naming and addressing*

ISO/IEC 8649, *Information technology – Open Systems Interconnection – Service definition for the Association Control Service Element*

ISO/IEC 8650 (all parts), *Information technology – Open Systems Interconnection – Connection-oriented protocol for the Association Control Service Element: Protocol specification*

ISO/IEC 8802-3, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*

ISO/IEC 9506-2, *Industrial automation systems – Manufacturing Message Specification – Part 2: Protocol specification*

ISO/IEC 10646 (all parts), *Information technology – Universal Multiple-Octet Coded Character Set (UCS)*

ISO/IEC 10731, *Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services*

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
P.O. Box 131
CH-1211 Geneva 20
Switzerland

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