PUBLICLY AVAILABLE SPECIFICATION

IEC PAS 62406

First edition
2005-08

Real-time Ethernet TCnet (Time-Critical Control Network)

Reference number
IEC/PAS 62406:2005(E)

Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

Consolidated editions

The IEC is now publishing consolidated versions of its publications. 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.

Further information on IEC publications

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 this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. 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 also available from the following:

- **IEC Web Site (www.iec.ch)**

- **Catalogue of IEC publications**

  The on-line catalogue on the IEC web site (www.iec.ch/searchpub) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.

- **IEC Just Published**

  This summary of recently issued publications (www.iec.ch/online_news/ justpub) is also available by email. Please contact the Customer Service Centre (see below) for further information.

- **Customer Service Centre**

  If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

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

PUBLICLY AVAILABLE SPECIFICATION

IEC PAS 62406

First edition
2005-08

Real-time Ethernet TCnet (Time-Critical Control Network)

International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch

Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

PRICE CODE XG

For price, see current catalogue

CONTENTS

FIGURES

TABLES

INTERNATIONAL ELECTROTECHNICAL COMMISSION

____

Real-time Ethernet TCnet (Time-Critical Control Network)

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.

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of a patent concerning TCnet.

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 document 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.

A PAS is a technical specification not fulfilling the requirements for a standard but made available to the public .

IEC-PAS 62406 has been processed by subcommittee 65C: Digital communications, of IEC technical committee 65: Industrial-process measurement and control.

The text of this PAS is based on the following document:	This PAS was approved for publication by the P-members of the committee concerned as indicated in the following document
Draft PAS	**Report on voting**
65C/353/NP	65C/370/RVN

Following publication of this PAS, the technical committee or subcommittee concerned will transform it into an International Standard.

It is intended that the content of this PAS will be incorporated in the futures new editions of the various parts of IEC 61158 series according to the structure of this series.

This PAS shall remain valid for an initial maximum period of three years starting from 2005-08. The validity may be extended for a single three-year period, following which it shall be revised to become another type of normative document or shall be withdrawn.

Real-time Ethernet TCnet (Time-Critical Control Network)

1 Introduction

1.1 General

This PAS describes a set of the specifications essential for the ISO/IEC 8802-3 based “Time-critical Control Network (TCnet)”, which is one of the communication networks for Real- Time Ethernet (RTE) defined in IEC 61784-2 and is referred to as “RTE-TCnet” hereafter, and each specification in this PAS is to be classified into a separate part of IEC 61158 series.

This PAS meets the industrial automation market objective of providing predictable time deterministic and reliable time-critical data transfer and means, which allow co-existing 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 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 PAS 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 RTE-TCnet. The RTE-TCnet communication profile, shown in Figure 1 as one of the profile set, 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 compliant to the ISO/IEC 8802-3 MAC protocol in order to provide both services for time-critical communications and common memory applications respectively.

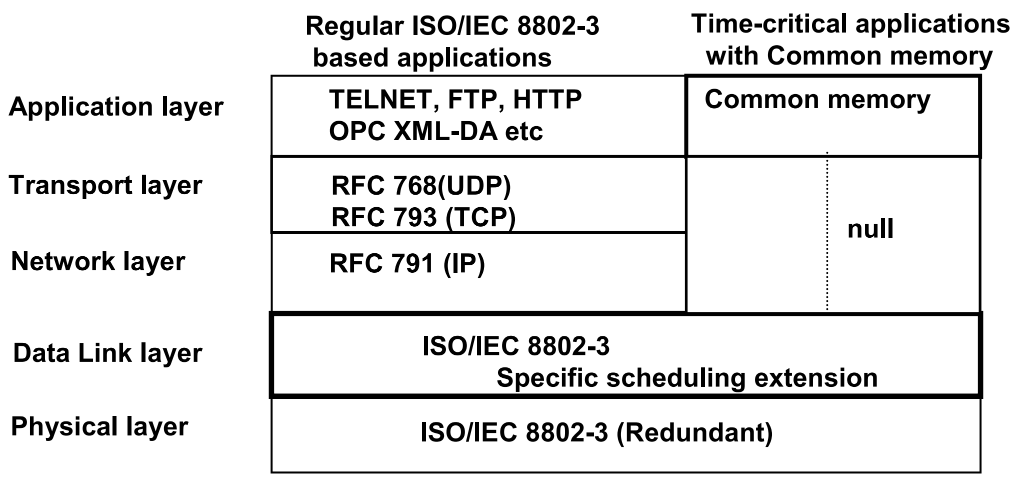


Figure 1 – RTE-TCnet communication profile

This PAS addresses the essential part of the RTE-TCnet profile, which are the extension part of ISO/IEC 8802-3 based Data Link layer and the Application layer exploiting the services of the Data Link layer immediately below.

This PAS describes the specifications essential for RTE-TCnet profile, specifically on the Data Link layer and the Application layer, in terms of the “three-layer” Fieldbus Reference Model which is based in part on the OSI Basic Reference Model. Other part of RTE-TCnet profile is not in the scope of this PAS.

1.2 Nomenclature for references within this PAS

Clauses, including annexes, can be referenced in their entirety, including any subordinate subclauses, as “Clause N” or “Annex N”, where N is the number of the clause or letter of the annex.

Subclauses can be referenced in their entirety, including any subordinate subclauses, as “N.M” or “N.M.P” and so forth, depending on the level of the subclause, where N is the number of the subclause or letter of the annex, and M, P and so forth represent the successive levels of subclause up to and including the subclause of interest.

When a clause or subclause contains one or more subordinate subclauses, the text between the clause or subclause heading and its first subordinate subclause can be referenced in its entirety as “N.0” or “N.M.0” or “N.M.P.0” and so forth, where N, M and P are as above.

Stated differently, a reference ending with “.0” designates the text and figures between a clause or subclause header and its first subordinate subclause.

2 Scope

2.1 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

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.

2.2 Scope and objectives

This PAS specifies the protocol set necessary for RTE-TCnet, specifically of the Data Link layer and the Application layer, which is mapped on top of the Data Link layer to exploit the services in terms of the “three-layer” Fieldbus Reference Model which is based in part on the OSI Basic Reference Model. Both Reference Models subdivide the area of standardization for interconnection into a series of layers of specification, each of manageable size. Throughout this PAS, the term “service” refers to the abstract capability provided by one layer of the OSI Basic Reference model to the layer immediately above.

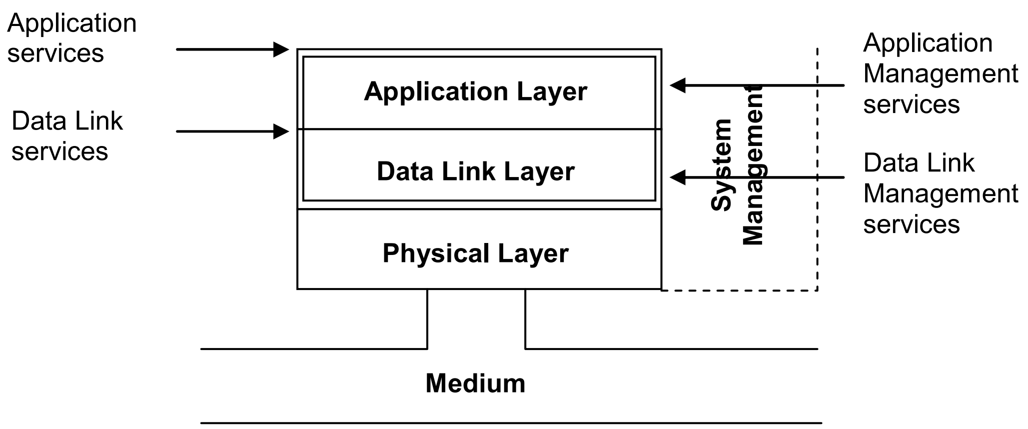


Figure 2 – Scope of this PAS

The relationship between the Data Link layer Services and Protocol, the Application Layer Services and Protocol, and the Systems management in this PAS is illustrated in Figure 2.

NOTE Systems management, as used in this PAS, is a local mechanism for managing the layer protocols.

This PAS consists of

- Data Link Layer Service definitions
- Data Link Layer Protocol specification
- Application Layer Service definitions
- Application Layer Protocol specification

The services both of the Data Link and the Application layer in this PAS is a conceptual architectural service, independent of administrative and implementation divisions.

The Data Link layer describes the extension for RTE-TCnet of ISO/IEC 8802-3 Data Link layer, and the Application layer describes the utilization of the global common memory, which is a conceptual virtual shared memory over the RTE-TCnet however is mapped onto the physical common memory provided in each node and is used for time-critical intercommunications among all participating application entities on each node over the RTE-TCnet in very simple fashion of output and input to/from regular memory configuration.

3 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 61158-3:2003, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 3: Data link service definition*

IEC 61158-4:2003 3, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 4: Data link protocol specification*

IEC 61158-5:2003, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 5: Application layer service definition*

IEC 61158-6:2003, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 6: Application layer protocol specification*

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

ISO/IEC 8802-3:2000, *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 8822:1994, *Information technology – Open Systems Interconnection – Presentation service definition*

ISO/IEC 8824-1:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*

ISO/IEC 8825-1:2002, *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*

ISO/IEC 8886:1996, *Information technology – Open Systems Interconnection – Data link service definition*

ISO/IEC 9545:1994, *Information technology – Open Systems Interconnection – Application Layer structure*

ISO/IEC 10731:1994, *Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services*

ISO/TR 13283:1998, *Industrial automation – Time critical communications archiectures – User requirements and network management for time-critical communications systems*

4 Terms and definitions

Terms and definitions of this PAS are described separately in 6.4 for Data Link Layer Service definitions, in 7.4 for Data Link Layer Protocol specification, in 8.4 for Application Layer Service definition and in 9.4 for Application Layer Protocol specification.

5 Symbols and abbreviations

Symbols and abbreviations of this PAS are described separately in 6.5 for Data Link Layer Service definitions, in 7.5 for Data Link Layer Protocol specification, in 8.4.8 for Application Layer Service definition and in 9.4.3 for Application Layer Protocol specification.

6 Data Link Layer Service definitions

6.1 Introduction

The Data Link layer service of RTE-TCnet is defined in this section. This section of the Data Link layer service definitions is to be one of type of IEC 61158-3 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, which is based in part on the Basic Reference Model for Open Systems Interconnection. Both Reference Models subdivide the area of standardization for interconnection into a series of layers of specification, each of manageable size.

The Data Link Service is provided by the Data Link Protocol making use of the services available from the Physical Layer. This section defines the Data Link Service characteristics that the immediately higher-level protocol may exploit. The relationship between the International Standards for Fieldbus Data Link Service, Fieldbus Data Link Protocol, Fieldbus Application Service, Fieldbus Application Protocol and Systems Management is illustrated in Figure 3.

NOTE Systems Management, as used in this PAS, is a local mechanism for managing the layer protocols.

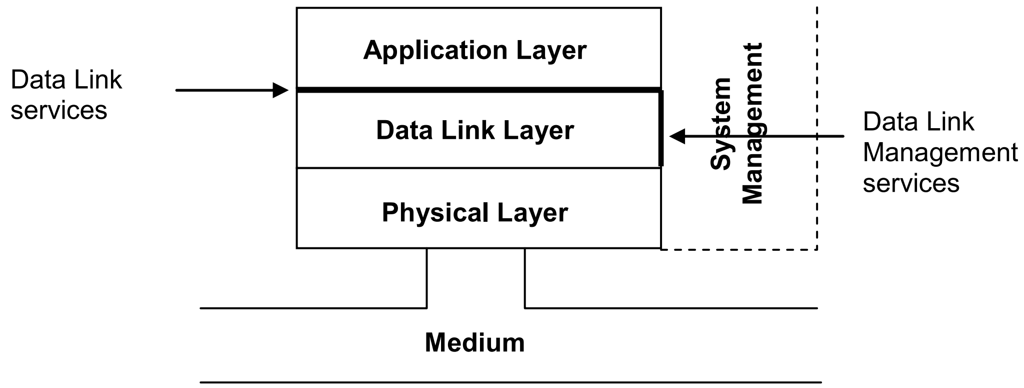


Figure 3 – Relationship of the RTE-TCnet Data link layer to other RTE layers and to users of RTE Data link service

6.2 Scope

6.2.1 Overview

The Data Link service of RTE-TCnet provides basic time-critical messaging communications between devices in an automation environment. 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 section defines in an abstract way the externally visible services provided by the Data Link Layer in terms of

a) The primitive actions and events of the service;

b) The parameters associated with each primitive action and event, and the form which they take; and

c) The interrelationship between these actions and events, and their valid sequences

The purpose of this section is to define the services provided to

1) The Application layer at the boundary between the Application and Data Link layers of the Fieldbus Reference Model, and
2) Systems Management at the boundary between the Data Link layer and Systems Management of the Fieldbus Reference Model

Seven distinct types of services are defined in IEC 61158-3; each has a corresponding protocol in IEC 61158-4. This RTE-TCnet Data Link service is to be included as an additional type into IEC 61158-3 as well as the corresponding RTE-TCnet Data Link protocol, defined in section 7 as an additional type into IEC 61158-4.

As additional type of IEC 61158 the RTE-TCnet Data Link service provides a superset of those services expected of OSI Data Link Protocol as specified in ISO/IEC 8886. Thus the RTE-TCnet Data Link service is able to support:

a) The OSI Network Layer at the boundary between the Network and Data Link Layers of the OSI Basic Reference Model
b) The IETF (IP) Network Layer

The RTE-TCnet Data Link Service and the different Types of Data Link services defined in IEC 61158-3 are each presumed self-consistent, but in general are unrelated to the other Types of service.

6.2.2 Specifications

The principal objective of this section is to specify the characteristics of conceptual Data Link Services suitable for time-critical communications, and thus supplement the OSI Basic Reference Model in guiding the development of Data Link protocols for time-critical communications.

This PAS may be used as the basis for formal DL-Programming-Interfaces. Nevertheless, it is not a formal programming interface, and any such interface will need to address implementation issues not covered by this PAS, including

a) The sizes and octet ordering of various multi-octet service parameters, and
b) The correlation of paired request and confirm, or indication and response, primitives.

6.2.3 Conformance

This PAS does not specify individual implementations or products, nor does it constrain the implementations of Data Link entities within industrial automation systems.

There is no conformance of equipment to this Data Link Service definition specification. Instead, conformance is achieved through implementation of conforming Data Link protocols that fulfill any given Type of Data Link Services as defined in this PAS.

6.3 Void

6.4 Terms and definitions

6.4.1 node

Single DL-entity as it appears on one local link

6.4.2 node-id

Two-octet primary identifier for the DLE on the local link, whose values are constrained. A permissible value is from 1 to 255. A value 0 is specifically used for SYN node, which emits SYN frame.

6.4.3 DLSAP

Distinctive point at which DL-services are provided by a single DL-entity to a single higher-layer entity

6.4.4 DLCEP-address

DL-address which designates either

a) One peer DL-connection-end-point

b) One multi-peer publisher DL-connection-end-point and implicitly the corresponding set of subscriber DL-connection-end-points where each DL-connection-end-point exists within a distinct DLSAP and is associated with a corresponding distinct DLSAP-address

6.4.5 multi-peer DLC

Centralized multi-end-point DL-connection offering DL-duplex-transmission between a single distinguished DLS-user known as the publisher or publishing DLS-user, and a set of peer but undistinguished DLS-users known collectively as the subscribers or subscribing DLS-users. The publishing DLS-user can send to the subscribing DLS-users as a group (but not individually), and the subscribing DLS-users can send to the publishing DLS-user (but not to each other).

6.4.6 multipoint connection

Connection from one node to many nodes. Multipoint connection allows data transfer from a single publisher to many subscriber nodes.

6.5 Symbols and abbreviations

6.5.1 Common symbols and abbreviations

6.5.1.1 CSMA/CD Carrier sense multiple access with collision detection

6.5.1.2 DA Destination address

6.5.1.3 DL- Data Link Layer (as a prefix)

6.5.2 DLC DL-connection

6.5.2.1 DLCEP DL-connection-end-point

6.5.2.2 DLE DL-entity (the local active instance of the Data Link Layer)

6.5.2.3 DLL DL-layer

6.5.2.4 DLPDU DL-protocol-data-unit

6.5.2.5 DLM DL-management

6.5.2.6 DLME DL-management Entity (the local active instance of DL-management

6.5.2.7 DLMS DL-management Service

6.5.2.8 DLS Dl-service

6.5.2.9 DLSAP DL-service-access-point

6.5.2.10 DLSDU DL-service-data-unit

6.5.2.11 MSDU MAC service data unit

6.5.2.12 OSI Open systems interconnection

6.5.2.13 Ph- Physical layer (as a prefix)

6.5.2.14 PhL Ph-layer

6.5.2.15 PhS Ph-service

6.5.2.16 PhSAP Ph-service-access-point

6.5.2.17 QoS Quality of service

6.5.2.18 SA Source address

6.5.3 RTE-TCnet: Additional symbols and abbreviations

6.5.3.1 CM Common memory

6.5.3.2 GCM Global common memory

6.5.3.3 RTE Real Time Ethernet

6.6 Conventions

6.6.1 General conventions

This PAS uses the descriptive conventions given in ISO/IEC 10731. The service model, service primitives, and time-sequence diagrams used are entirely abstract descriptions; they do not represent a specification for implementation.

6.6.1.1 Parameters

Service primitives, used to represent service user/service provider interactions (see ISO/IEC 10731), convey parameters that indicate information available in the user/provider interaction.

This PAS uses a tabular format to describe the component parameters of the DLS primitives. The parameters that apply to each group of DLS primitives are set out in tables throughout the remainder of specification. Each table consists of up to six columns, containing the name of the service parameter, and a column each for those primitives and parameter-transfer directions used by the DLS:

The request primitive's input parameters;

The request primitive's output parameters;

The indication primitive's output parameters;

The response primitive's input parameters; and

The confirm primitive's output parameters

NOTE The request, indication, response and confirm primitives are also known as requestor.submit, acceptor. deliver, acceptor, submit, and requestor. deliver primitives, respectively (see ISO/IEC 10731).

One parameter (or part of it) is listed in each row of each table. Under the appropriate service primitive columns, a code is used to specify the type of usage of the parameter on the primitive and parameter direction specified in the column:

M – Parameter is mandatory for the primitive.

U – Parameter is a User option, and may or may not be provided depending on the dynamic usage of the DLS-user. When not provided, a default value for the parameter is assumed.

C – Parameter is conditional upon other parameters or upon the environment of the DLS-user.

(Blank) – Parameter is never present.

Some entries are further qualified by items in brackets. These may be

a) A parameter-specific constraint

(=) indicates that the parameter is semantically equivalent to the parameter in the service primitive to its immediate left in the table.

b) An indication that some note applies to the entry

(n) indicates that the following note n contains additional information pertaining to the parameter and its use.

In any particular interface, not all parameters need be explicitly stated. Some may be implicitly associated with the DLSAP at which the primitive is issued.

In the diagrams which illustrate these interfaces, dashed lines indicate cause-and-effect or time-sequence relationships, and wavy lines indicate that events are roughly contemporaneous.

6.6.1.2 Identifiers

Many of the DLS primitives specify one or more identifier parameters that are drawn from either a local DL-identifier space or a local DLS-user-identifier space. The existence and use of such identifiers in an implementation of the services specified in this PAS is a purely local issue. Nevertheless, these identifiers are specified explicitly in these primitives to provide a descriptive means

a) of canceling (aborting) an outstanding request primitive before receiving its corresponding confirm primitive;

b) for referring within a request or response primitive to persistent DL-objects, such as buffers and queues, which were created as the result of a previous DLS primitive; and

c) for referring within an indication or confirm primitive to persistent DL-objects which were created as the result of a previous DLS primitive.

Adherence to the OSI principle of architectural layering necessitates the presumption of distinct non-intersecting identifier spaces for the DLS-provider and each separate DLS-user, because they may have non-overlapping local views. Consequently, DLS-user identifiers are required for a) and b); while DL-identifiers are required for c).

6.6.2 Additional convention

In the diagrams which illustrate the DLS and DLM interfaces, dashed lines indicate cause-and-effect or time-sequence relationships between actions at different stations, while solid lines with arrows indicate cause-and-effect time-sequence relationships which occur within the DLE-provider at a single station.

6.7 Data link service and concept

6.7.1 General description of services

6.7.1.1 General

The DLS provides for transparent and reliable transfer of data between DLS-users over RTE-TCnet. The DLS is base on services provided by the physical layer of ISO/IEC 8802-3 to the conceptual interface between the Physical and Data Link layers.

Two types of data transmission services are provided:

a) **Time-critical Cyclic data service**: The connection-oriented buffer transfers between pre-established point-to-multipoint DLCEPs on the same local link.

b) **Sporadic message data service**: The unacknowledged connectionless message transfers between single DLSAPs, or unacknowledged connectionless message transfer from a single DLSAP to a group of DLSAPs on the extended link.

NOTE For the purpose of clarity, the expressions "buffer transfer" and "message transfer" are used to distinguish between the two types of communications services, connection-oriented and connectionless, respectively, that are offered by this DLS,

The buffer transfer service or the Time-critical Cyclic data service is based on cyclic data transfers of three kinds of transmission period. The transmission periods and the total volumes of each level of the buffer transfer by the multiple distributed DLS-providers on the same local link are defined when the system is configured, and are based on application needs. Cyclic data transfers are automatically triggered by the communications system without the user requesting them.

There are also three types of Time-critical Cyclic data service according to the transmission period:

a) **High-speed Cyclic data transmission**

b) **Medium-speed Cyclic data transmission**

c) **Low-speed Cyclic data transmission**

The message transfer service or the Sporadic message data service is based on aperiodic data transfer, sporadically occurred upon DLS-user requesting one or more message to transfer. The priority level and the target token rotation time, corresponding to the target time to obtain the transmission right to send out the message data on the medium, is defined when the system is configured, and are based on application needs. The transmit delay is depending on the priority level and the target token rotation time. Regular ISO/IEC 8802-3 Ethernet message frame is transferred by means of this message transfer service.

The DLS provides DLS-users with a means to set up a quality of service for the transfer of data. QoS is specified by means of QoS parameters representing aspects such as data transmit delay, priority level, DLCEP data length, and so on. A DL-management Service (DLMS) is defined in 6.8.

6.7.1.2 Overview of the Data frames flow on the medium

Overview of the data frames flow on the medium is shown in

Figure 4 and Figure 5.

The DLL provides the opportunity of transferring data to each node in a sequential order and within a predetermined time period. At the start time of every High-speed-Transmission-period (Tsyn), the SYN frame is broadcasted to all RTE-TCnet nodes. Received the SYN frame, the node with sequential number 1 (Node 1) starts sending its data frames, and after the completion of its data frames transmission Node 2 can send out its data frames. The Nth node (Node N) can obtain the transmission right after the (N-1) th node completes its data frames transmission. The sequential number is assigned to each node at the time approval to join the RTE-TCnet is granted, and is up to 255.

Each node can hold the transmission right for a preset time and must transfer the transmission right to the next node within the preset time. The data to be sent and the data to be held over are determined by priority.

Data transmission includes cyclic data and sporadic message transmission. Cyclic data transmission is divided into High, Medium and Low-speed Cyclic data transmission. Each node sends the High-speed Cyclic data frames on each occasion when it obtains the transmission right. The data of lower priorities, that is the Medium-speed Cyclic data, the sporadic message data and the Low-speed Cyclic data respectively, is sent or not sent depending on the circumstances.

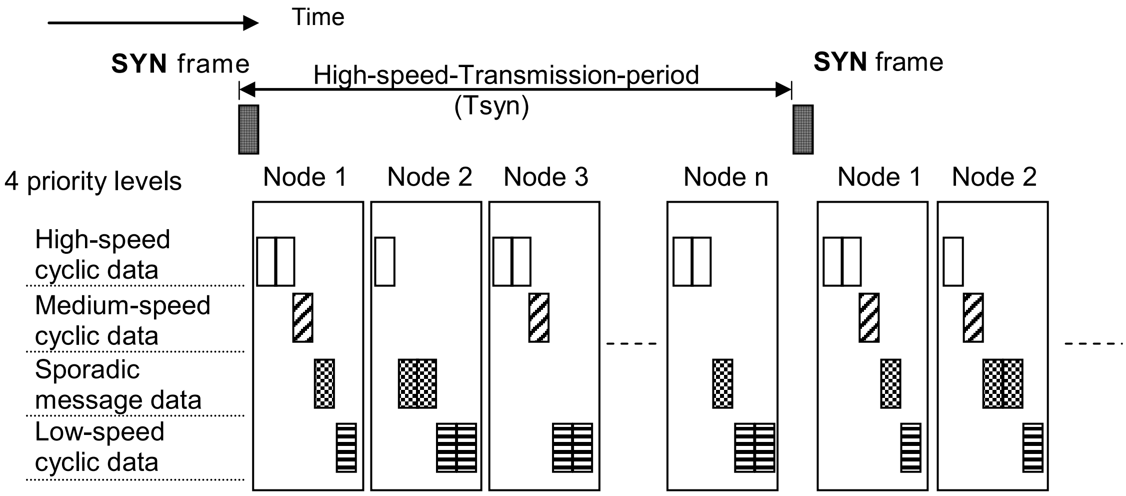
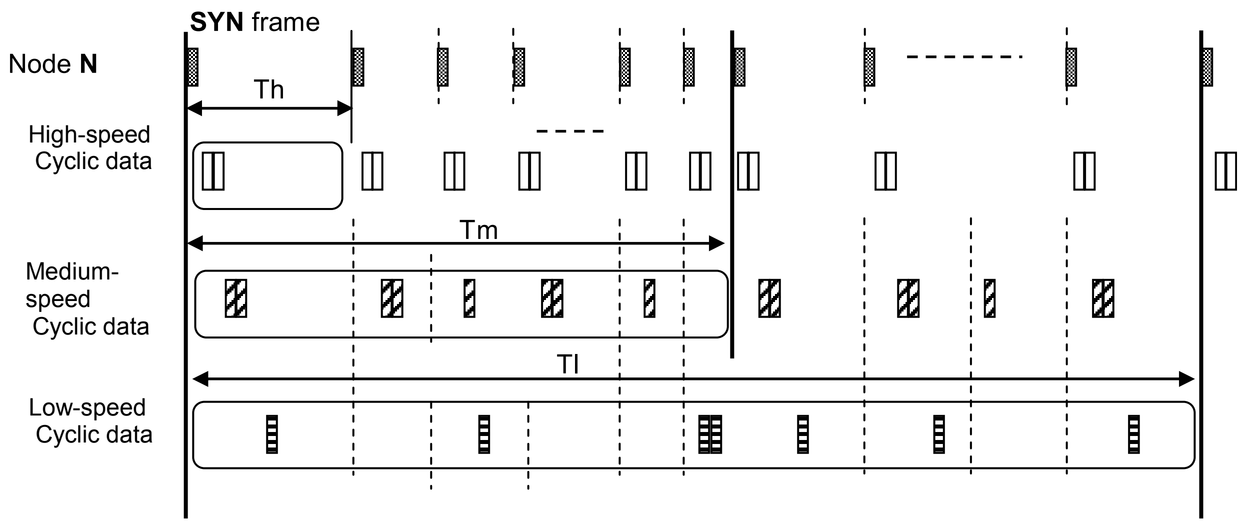


Figure 4 – Overall flow of Data frames during 1 Minimum Cycle period (High-speed Transmission-period)

The holding time of the transmission right of each node is determined by the settings of the High-speed Cyclic, the Medium-speed Cyclic, the sporadic message and the Low-speed Cyclic data transmission periods and by the volume of transmission data for each node. After sending all the High-speed Cyclic data, the node sends the Medium-speed Cyclic data. If the holding time of the transmission right ends during sending the Medium-speed Cyclic data, the transmission of the Medium-speed Cyclic data is interrupted. The Node N obtains the transmission right again during the next High-speed transmission period, during which time all the High-speed Cyclic data and the remainder of the previous Medium-speed Cyclic data is sent. As to the Low-speed Cyclic data is sent out in this fashion.



Th: High-speed transmission period (1 Cycle period)
Tm: Medium-speed transmission period (1 Cycle period)
Tl: Low-speed transmission period (1 Cycle period)

Figure 5 – Overall flow of Cyclic Data frames over 1 Maximum cycle period (Low-speed-Transmission-period)

6.7.1.3 Sequence of primitives

A request primitive is used by a DLS-user to request a service. A confirm primitive is returned to the DLS-user upon completion of the service. Moreover, an indication primitive is used to report to the DLS-user the receipt of new DLS-User Data or the receipt of a new message.

6.7.1.3.1 Primitives on Time-critical Cyclic Data service

The sequence of primitives on Time-critical Cyclic Data service is shown in Figure 6.

DL_DATA_REQ indication primitive informs the DLS-user initiates the data transfer using DL_PUT service from the corresponding DLS-user buffer associated with a specified DLCEP to the send buffer of local DLE for publishing the data.

DL_Put request primitive which responds to DL_DATA_REQ indication primitive allows the DLS-user to transfer data of the corresponding DLS-user buffer associated with a specified DLCEP by the DL_DATA_REQ indication primitive to the Send_buffer of the local DLE where the DLE is the publisher.

DL_Buffer_Received indication primitive informs the DLS-user that a subscribed data if a DLCEP has just correctly received. The data in the Receive_buffer of the local DLE is updated and is available, and is read by the DLS-user using the DL_Get request primitive.

DL_Get request primitive which responds to DL_Buffer_Received indication primitive allows the DLS-user to get the data in the Receive_buffer of the local DLE and to transfer the data from the Receive_buffer to the corresponding DLS-user buffer associated with the specified DLCEP by DL_Buffer_Received indication primitive by the DLE where the DLE is the subscriber.

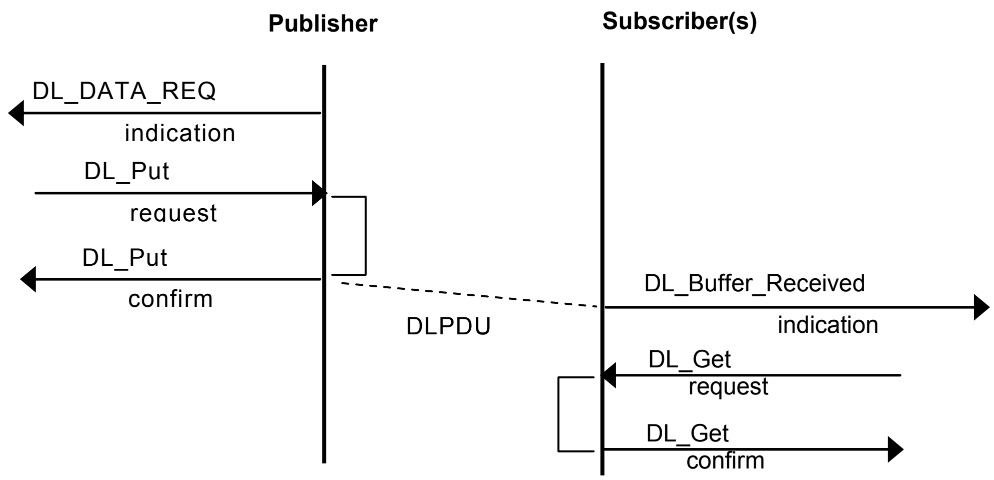


Figure 6 – Sequence Diagram of Time-critical Cyclic Data service

6.7.1.3.2 Sporadic Message Data service

The sequence of primitives on Sporadic Message Data service is shown in Figure 7.

DL_SPDATA request and DL_SPDATA indication correspond to the MA_DATA request and MA_DATA indication defined by ISO/IEC 8802-3 respectively.

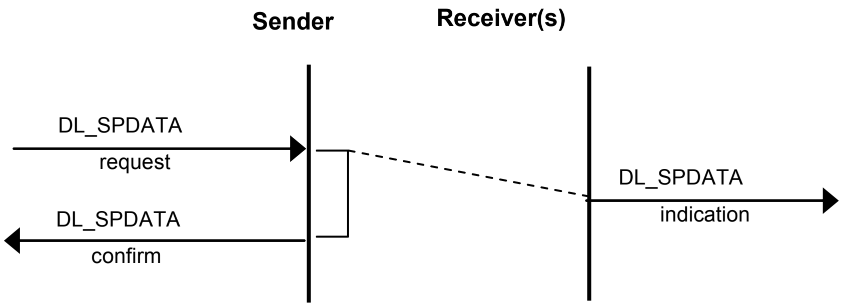


Figure 7 – Sequence diagram of Sporadic Message service

6.7.1.4 Addressing

Two different types of addressing are for the DL-addressing: one for Time-critical Cyclic Data service and the other for Sporadic Message Data service.

For buffer transfers of Time-critical Cyclic Data service: Each variable in the system is associated with a DLCEP-identifier that characterizes it within the system in a unique manner. Entities participating in a buffer transfer are not identified explicitly. Rather, they are identified indirectly as subscribers or publisher of the identified variable. Each variable has only one publisher.

For message transfer of Sporadic Message Data service: One or more DLSAP-addresses are defined within each DLE. These DLSAP-addresses give access to a message transfer service. Each DLSAP-address identifies an access point to a message service linked to a DLS-user entity. During the message transaction two DLSAP-addresses are indicated in order to establish contact between the communicating entities. Each DLSAP-address specifies a DLS-user of the message service (for both emission and reception). This DL-address is unique within the extended link.

Variable addressing is restricted to the local link. The addressing mechanism makes it possible to identify variables and exchanges independent of the producing and consuming DLEs. For buffer transfers all relationships between the various DLS-users are known and defined when the system is configured. Each DLCEP-identifier characterizes a single system variable and thus establishes a relationship between the unique publisher of the variable and the subscribers of the variable.

Buffer transfers use the local broadcast medium and are restricted to the local link: the DLCEP-identifier and the value of a variable are made available to all DLEs on the local link. The DLCEP-identifier associated with the variable allows each DLE to recognize whether or not it is the publisher or a subscriber of the value associated with the identified variable.

The relationship of peer to peer and multi-peer DLCs and their DLCEPs is shown in Figure 8. DLSAPs and PhSAPs are depicted as ovals spanning the boundary between two adjacent layers. DL-addresses are depicted as designating small gaps (points of access) in the DLL portion of a DLSAP. A DLCEP-address also designates a specific point of information flow (its DLCEP) within the DLSAP. A single DL-entity may have multiple DLSAP-addresses and group DL-addresses associated with a single DLSAP. This figure also shows the relationships of DL-paths and PhSAPs.

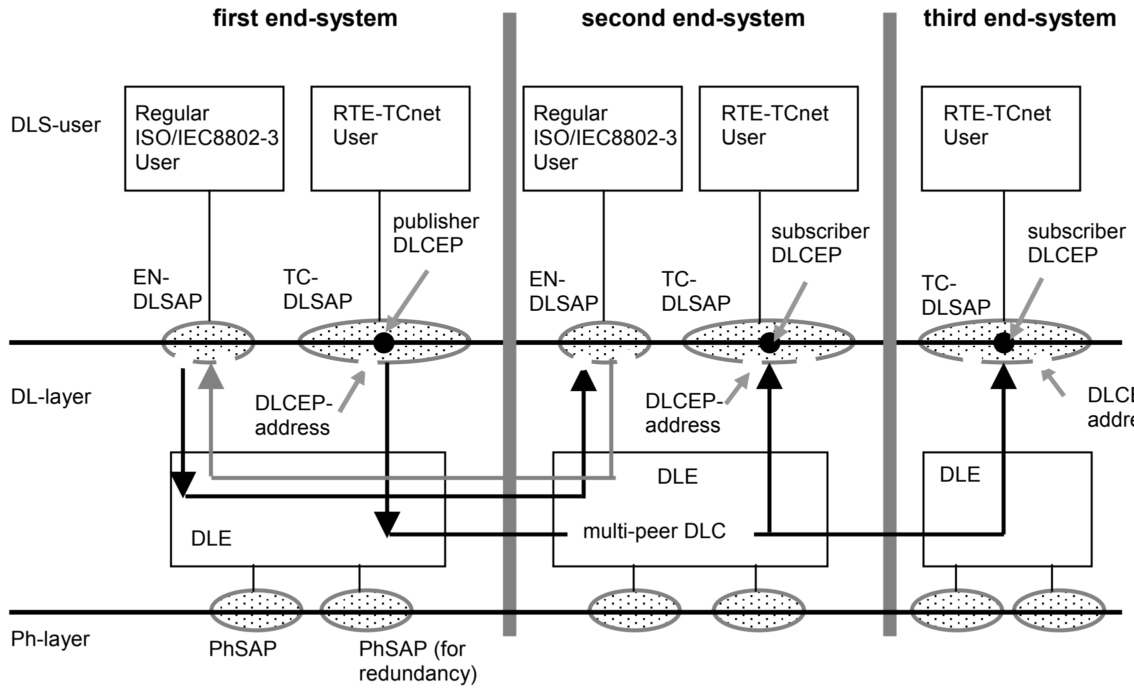


Figure 8 – Relationship of DLSAP, DLCEP and DLCEP-address

6.7.2 Time critical cyclic data service

6.7.2.1 General

This DLS provides connection-oriented buffer transfers between pre-established point-to-multipoint DLCEPs on the same local link.

6.7.2.2 Sequence of primitives

The Time-critical Cyclic Data service primitives and the parameters are summarized in Table 1, and the primitive sequence is shown in Figure 6.

Table 1 – Primitives and parameters used on time critical cyclic data service

Function	Location	Primitive	Direction	Parameters
Notify publisher DLS-user	Publisher	DL_DATA_REQ indication	from DLE	DLCEP-identifier DLSDU-length
Put buffer	Publisher	DL_Put request	to DLE	DLCEP-identifier DLSDU-length DLSDU
Put confirm to publisher DLS-user	Publisher	DL_Put confirm	from DLE	DLCEP-identifier Status
Notify subscriber DLS-user	Subscriber(s)	DL_Buffer_Received indication	from DLE	DLCEP-identifier DLSDU-length
Get buffer	Subscriber(s)	DL_Get request	to DLE	DLCEP-identifier DLSDU-length
Confirm to publisher DLS-user	Subscriber(s)	DL_Get confirm	from DLE	DLCEP-identifier DLSDU-length DLSDU Status
NOTE In this table, time increases from top to bottom.				

6.7.2.3 Data request

6.7.2.3.1 Function

DL_DATA_REQ service informs the DLS-user that initiates the transfer using the DL-Put service from the corresponding DLS-user buffer associated with a specified DLCEP to the Send_buffer of local DLE for publishing the data.

6.7.2.3.2 Types of primitives and the parameters of Data request

Table 2 indicates the parameters of Data request service.

Table 2 – Data request primitives and the parameters

DL-DATA_REQ	Indication
Parameter name	output
DLCEP-identifier	M
DLSDU-length	M

6.7.2.3.2.1 DLCEP-identifier

This parameter specifies from which buffer the DLS-user reads and transfers the data into the Send_buffer of the local DLE for publishing.

6.7.2.3.2.2 DLSDU-length

This parameter indicates the length of the identified buffer, that is the actual length of the contents in a DLSDU.

6.7.2.4 Put Buffer

6.7.2.4.1 Function

DL_Put request, which responds to DL_DATA_REQ indication, allows the DLS-user to transfer data of the corresponding DLS-user buffer associated with the specified DLCEP by the DL_DATA_REQ indication to the Send_buffer of the local DLE where the DLE is the publisher.

6.7.2.4.2 Types of primitives and parameters of Put buffer

Table 3 indicates the types of primitives and parameters for Put buffer.

Table 3 – Put buffer primitives and parameters

DL-Put	**Request**	**Confirm**
Parameter name	input	output
DLCEP-identifier	M	M(=)
DLSDU-length	M	M(=)
DLSDU	M	
Status		M

6.7.2.4.2.1 DLCEP-identifier

This parameter indicates from which buffer DLS-user reads and writes data into the Send_buffer of the local DLE.

6.7.2.4.2.2 DLSDU-length

This parameter indicates the length of the identified buffer, that is the actual length of the contents in a DLSDU.

6.7.2.4.2.3 DLSDU

This parameter specifies the information that is transferred by buffer transfer from the local DLE as a publisher to the remote multi-peer DLEs as subscribers.

6.7.2.4.2.4 Status

This parameter allows the DLS-user to determine whether the requested DLS was provided successfully, or failed for the reason specified. The value conveyed in this parameter is as follows:

a) "success"
b) "failure – invalid requested parameter"

6.7.2.5 Get Buffer

6.7.2.5.1 Function

DL_Get request which responds to DL_Buffer_Received indication allows the DLS-user to get and transfer the data from the Receive_buffer of the local DLE to the corresponding DLS-user buffer associated with the specified DLCEP by DL_Buffer_Received indication by the DLE where the DLE is the subscriber.

6.7.2.5.2 Type of primitives and parameters of the Get buffer

Table 4 indicates the types of primitives and parameters for Get buffer.

Table 4 – Get buffer primitives and parameters

DL-Get	**Request**	**Confirm**
Parameter name	input	output
DLCEP-identifier	M	M(=)
DLSDU-length	M	M(=)
DLSDU		M
Status		M

6.7.2.5.2.1 DLCEP-identifier

This parameter unambiguously designates to which buffer DLS-user reads and transfers data from the Receive_buffer of the local DLE.

6.7.2.5.2.2 DLSDU-length

This parameter indicates the length of the data to be transferred from the Receive_buffer of the local DLE to the identified buffer of the DLS-user.

6.7.2.5.2.3 DLSDU

This parameter specifies the information that is transferred by buffer transfer by the remote DLE as a publisher to the local DLE as subscribers.

6.7.2.5.2.4 Status

This parameter allows the DLS-user to determine whether the requested DLS was provided successfully, or failed for the reason specified. The value conveyed in this parameter is as follows:

a) "success"

b) "failure -- invalid requested parameter"

6.7.2.6 Notify Buffer Received

6.7.2.6.1 Function

DL_Buffer_Received indication informs the DLS-user that a subscribed identified data has just correctly received. The data in Receive_buffer of the local DLE is updated and is available, and is read by the DLS-user using the DL_Get request primitive.

6.7.2.6.2 Types of primitives and parameters of the Notify buffer received

Table 5 indicates the types of primitives and parameters for Notify buffer received.

Table 5 – Notify buffer received primitives and parameters

DL-Buffer_Received	Indication
Parameter name	output
DLCEP-identifier	M
DLSDU-length	M

6.7.2.6.2.1 DLCEP-identifier

This parameter unambiguously designates to which buffer DLS-user reads and transfers the data from the receive_buffer of the local DLE.

6.7.2.6.2.2 DLSDU-length

This parameter indicates the length of the identified buffer that is the actual length of the contents in a DLSDU.

6.7.3 Detail description of Sporadic Message Data service

6.7.3.1 General

This DLS provides unacknowledged connectionless message transfers between single DLSAPs, or unacknowledged connectionless message transfer from a single DLSAP to a group of DLSAPs on the extended link.

A pair set of DL_SPDATA request and DL_SPDATA indication corresponds to that of the MA_DATA request and MA_DATA indication defined by ISO/IEC 8802-3 respectively.

6.7.3.2 Sequence of primitives

The Sporadic Message Data service primitives and the parameters are summarized in Table 6, and the primitive sequence is shown in Figure 7.

Table 6 – Primitives and parameters used on Sporadic message data service

Function	Location	Primitive	Direction	Parameters
Send DLSDU	Sender	DL_SPDATA request	to DLE	DA(destination_address) SA(source_address) MSDU(m_sdu) Service_class
Send confirmation to calling DLS-user	Sender	DL_SPDATA confirm	from DLE	Status
Notify and provide MA_SDU received to peer called DLS-user	Receiver(s)	DL-SPDATA indication	from DLE	DA(destination_address) SA(source_address) MSDU(m_sdu) Rec_Status(reception_status)

NOTE 1 In this table, time increases from top to bottom.

NOTE 2 DL_SPDATA service primitives respond to the MA_DATA service primitives defined by ISO/IEC 8023:2003.

6.7.3.3 Send Sporadic message

6.7.3.3.1 Function

DL_SPDATA service primitives allow the DLS-user to transfer message data to a single peer DLS-user or multi-peer DLS-users in remote nodes.

6.7.3.3.2 Types of primitives and parameters of the Send sporadic message

Table 7 indicates the types of primitives and parameters for Send sporadic message.

Table 7 – Submit sporadic message primitives and parameters

DL-SPDATA	Request	Indication	Confirm
Parameter name	input	output	output
DA(destination address)	M	M(=)	M(=)
SA(source address)	U	M(=)	
Service_class	U		
MSDU（m_sdu）	M	M(=)	
Rec-Status（reception status）		M	
Status			M

6.7.3.3.2.1 DA (destination address)

This parameter indicates the DLE(s) for which the DLPDU is intended. It may be an individual or multicast (including broadcast) address.

6.7.3.3.2.2 SA (source address)

This parameter indicates an individual DLE from which the DLPDU is sending.

6.7.3.3.2.3 MSDU (MAC service data unit)

This parameter specifies the MAC service data unit to be transmitted by MAC sublayer entity.

6.7.3.3.2.4 Service_Class

This parameter indicates a quality of service requested by DLS-user.

NOTE The RTE-TCnet DL protocol as well as the CSMA/CD (ISO/IEC 8802-3) MAC protocol provides a single quality of service regardless of the service_class.

6.7.3.3.2.5 Rec_Status

This parameter is used to pass status information upon the DLPDU received to the DLS-user.

a) "success – received without error "

b) "failure – received but with error "

6.7.3.3.2.6 Status

This parameter allows the DLS-user to determine whether the requested DLS was provided successfully, or failed for the reason specified. The value conveyed in this parameter is as follows:

a) "success – successfully completed "

b) "failure – invalid requested parameter"

6.8 DL-management services

6.8.1 General

The interface between a DLE and a DL-management user (DLMS-user) is described. The services of this interface are needed for the protocol which implements the DLS specified in 6.7 Data link service and concept.

6.8.2 Facilities of the DL-Management service

DL-management service provides the facilities for the initialization, configuration, event and error handling between the DLMS-user and the logical functions in the DLE. The following functions are provided to the DLMS-user.

a) Reset of the local DLE

b) Request for and modification of the actual operating parameters and of the counters of the local DLE

c) Notification of unexpected events, errors, and status changes, both local and remote

d) Request for identification and for the Common Memory configuration of the local DLE

e) Activation and deactivation of the Cyclic Data transmission of the local DLE

6.8.3 Service of the DL-management

6.8.3.1 Overview

DL-management provides the following services to DLMS-user.

a) Reset

b) Set Value

c) Get Value

d) Event

e) Set Publisher configuration

f) Get Publisher configuration

g) Activate Publisher

h) Deactivate Publisher

The services Reset, Set Value, Event, Set Publisher configuration and Activate Publisher are considered mandatory. Other services are considered optional.

6.8.3.2 Reset

The DLMS-user employs this service to cause DL-management to reset the DLE. A reset is equivalent to power on. The DLMS-user receives a confirmation thereof.

6.8.3.3 Set value

The DLMS-user employs this service to assign new values to the variables of the DLE. The DLMS-user receives a confirmation whether the specified variables have been set to the new values.

6.8.3.4 Get value

This service allows the DLMS-user to read variables of the DLE. The response of the DL-management returns the actual value of the specified variables.

6.8.3.5 Event

DL-management employs this service to inform the DLMS-user about certain events or error in the DLL.

6.8.3.6 Set Publisher configuration

The DLMS-user employs this service to assign new Publisher configuration parameters to the Time-critical cyclic data of the DLE. DLS-user receives a confirmation whether the specified parameters have been set to the new one.

6.8.3.7 Get Publisher configuration

This service enables DL-management to read Publisher configuration parameters of the Time-critical cyclic data. The response of the DL-management returns the actual value of Publisher configuration parameters.

6.8.3.8 Activate Time-critical Cyclic data service

This service allows the DLMS-user to activate the Time-critical Cyclic data transmission service. The DLMS-user receives the confirmation whether the designated Time-critical Cyclic data transmission has been started or this request has been completed unsuccessfully.

6.8.3.9 Deactivate Time-critical Cyclic data service

This service allows the DLMS-user to deactivate the Time-critical Cyclic data transmission service. The DLMS-user receives the confirmation whether the designated Time-critical Cyclic data transmission has been terminated or this request has been completed unsuccessfully.

Overview of interactions

The DL-management services and their primitives are summarized in Table 8.

Table 8 – Summary of DL-management primitives and parameters

Service	Primitive	Parameter
Reset	DLM-Reset request	(<none>)
	DLM-Reset confirm	(out Status)
Set value	DLM-Set-Value request	(in Variable-name, Desired-value)
	DLM-Set-Value confirm	(out Status)
Get value	DLM-Get-Value request	(in Variable-name)
	DLM-Get-Value confirm	(out Status, Current-value)
Event	DLM-Event indication	(out Event-identifier, Additional-information)
Set Publisher configuration parameters	DLM-Set-Publisher-Configuration request	(in Desired-speed-class, Desire-configuration)
	DLM-Set-Publisher-Configuration confirm	(out Status)
Get Publisher configuration parameters	DLM-Get-Publisher-Configuration request	(in Desire-speed-class)
	DLM-Get-Publisher-Configuration confirm	(out Status, Current-configuration)
Activate Time-Critical Cyclic service	DLM-Activate-TCC request	(in Desired-speed-class)
	DLM-Activate-TCC confirm	(out Status)
Deactivate Time-Critical Cyclic service	DLM-Deactivate-TCC request	(in Desired-speed-class)
	DLM-Deactivate-TCC confirm	(out Status)

The sequence of the DL-management primitives are shown in Figure 9 and Figure 10.

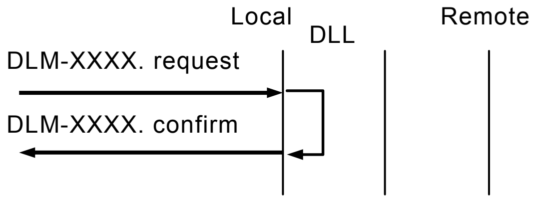


Figure 9 – Sequence diagram of Reset, Set value, Get value, Set CM configuration, Get CM configuration, Activate Time-critical cyclic data and Deactivate Time-critical Cyclic data service primitives

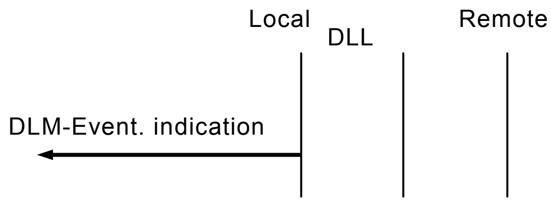


Figure 10 – Sequence diagram of Sporadic Message service

6.8.4 Detail specification of service and interactions

6.8.4.1 Reset

6.8.4.1.1 Function

The DLM-Reset request primitive causes DLMS to reset the DLE. The DLE assumes the "Offline" status after carried out in the same manner as at a power on, and all DLE variables are cleared. The DLMS-user receives DLM-Reset confirmation primitive with the status of the result in success or failure.

6.8.4.1.2 Type of primitives and parameters of DLM-Reset

Table 9 indicates the primitive and parameters of the Reset service.

Table 9 – DLM-Reset primitives and parameters

DLM-Reset	Request	Confirm
Parameter name	input	output
Status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

6.8.4.1.2.1 Status

This parameter allows the DLMS-user to determine whether the requested DLMS was provided successfully, or failed for the reason specified. The value conveyed in this parameter is as follows:

a) "OK – successfully completed"

b) "Failure – terminated before completion".

6.8.4.2 Set value

6.8.4.2.1 Function

This service is used to assign new values to the variables of the DLE. The DLMS-user receives the confirmation whether the specified variables has been set to the new values.

6.8.4.2.2 Type of primitives and parameters of DLM-Set-value

Table 10 indicates the primitive and parameters of the set service.

Table 10 – DLM-Set-value primitives and parameters

DLM-Set value	Request	Confirm
Parameter name	input	output
Variable-name	M	
Desired-value	M	
Status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

6.8.4.2.2.1 Variable-name

This parameter specifies the variable within the DLE whose value is to be set. The selectable variables are defined in the corresponding part of clause 7 Data Link Layer Protocol specification.

Table 11 – Mandatory DLE-variables and the permissible values

Operating parameters		
Variable-name	**Description**	**Permissible values**
Th	High-speed transmission period	value 1 to 160 in 0,1millisecond)
Tm	Medium-speed transmission period	value 10 to 1000 (in 1 millisecond)
Tl	Low-speed transmission period	value 100 to 10000 (in 1 millisecond)
ST	Slot-time is the fundamentally observational time unit, using in DLME for observing to initiate action such for re-initialisation in sending out CLM packet, for sending out the CMP packet specifically by the current SYN node in substitute for the node missed to send out the CMP packet.	Value 1 to 255 (in 512 bits physical symbol time identical to ISO/IEC 8802-3. Default value is 20 (= 100 μs).)
MTHT	Maximum-token-hold-time for high-speed cyclic data transmission	Value 1 to 2^{16}-1 (in octet time. Default value is 0x30B4 (= 900 μs).)
TISPD	Time-interval cyclically processed for Sporadic message data service	Value is 1 to 1000 (in 1 millisecond. Default value is 100.)
MD	Maximum distance on the connection path between any 2 nodes	Value 1 to 100 (in kilometre. Default value is 8.)
MDD	Maximum difference of the distance of two redundant physical mediums on the connection path between any 2 nodes	Value 0 to 2000 (in metre. Default value is 500.)
MPD	Maximum difference of the signal delay time propagating over two redundant physical mediums on the connection path between any 2 nodes	Value 1 to 0x7c (in 0,04 μs. Default value is 0x7c (≈ 5 μs).)
MN	Maximum node number	254 (Default value)
IA	Individual address of this node	Individual address in 48 bits length identical to ISO/IEC 8802-3.
MGA	Multicast group address	Multicast address in 48 bits length identical to ISO/IEC 8802-3, for logically associated TCnet nodes. Default value is 0x01-0x00-0x5e-0x50-0x00-0x01.
MRT	Maximum number of the repeater units on the connection path between any 2 nodes	Value 0 to N (Default value is 3.)
BW	Length of Time-critical Cyclic Data word in a DLPDU.	Value is 32 to 128. (in octet. Default value is 128.)
SCMPL	Permissible repetitive count of the CMP packet sent by SYN node in substitution for the corresponding node in order to detect a node out of service.	Value 1 to 16. (Default value is 3)

Table 11 – *(continued)*

Operating parameters		
Variable-name	**Description**	**Permissible values**
RCS	Receive- channel switching control for receiving packets	Designates the switching control for receiving packets out of one of two receive-channel A and B corresponding to each of the redundant medium A and B. “Automatic”: Automatically switch to the proper receive-channel. “Force A”, “Force B”: Force to switch Receive-channel A or B respectively. Default is “Automatic”.
RMGP	Maximum time-interval for one receive-channel, which has already completed one packet received and has waited for IGP time, to wait the completion of a packet to be received on the other receive-channel in order to detect the other receive-channel disrupted.	Value is 250. (In 0,04 μs. Value 250 is equal to 10 μs)
PBh	List of the data buffers using for sending out High-speed time-critical cyclic data	Max. number of the buffers is 2048.
PBm	List of the data buffers using for sending out Medium-speed time-critical cycle data	Max. number of the buffers is 2048.
PBl	List of the data buffers using for sending out Low-speed time-critical cyclic data	Max. number of the buffers is 2048.

6.8.4.2.2.2 Desired-value

This parameter specifies the desired value for the selected variable. The permitted values or value ranges are defined particularly in clause 7. Though a part of the mandatory DLE-variables and the permissible vales are shown for the information in Table 11.

6.8.4.2.2.3 Status

This parameter allows the DLMS-user to determine whether the requested DLMS was provided successfully, or failed for the reason specified. The value conveyed in this parameter is as follows:

a) “OK – success – the variable could be update"

b) “Failure – the variable does not exist or could not assume the new value”

c) "Failure – invalid parameters in the request"

6.8.4.3 Get value

6.8.4.3.1.1 Function

This service can be used to read the value of a DLE variable. The response of the DLMS returns the actual value of the specified variable.

6.8.4.3.2 Type of primitives and parameters of DLM-Get-value

Table 12 indicates the primitive and parameters of the DLM-Get-value service.

Table 12 – DLM-Get-value primitives and parameters

DLM-Get value	Request	Confirm
Parameter name	input	output
Variable-name	M	
Desired-value		M
Status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

6.8.4.3.2.1 Variable-name

This parameter specifies the variable within the DLE whose value is being requested. The selectable variables are defined in the corresponding part of clause 7.

6.8.4.3.2.2 Status

This parameter allows the DLMS-user to determine whether the requested service was provided successfully, or failed for the reason specified. The value conveyed in this parameter is as follows:

a) “OK – success – the variable could be read"

b) “Failure – the variable does not exist or could not be read”

c) "Failure – invalid parameters in the request"

6.8.4.3.2.3 Current-value

This parameter is present when the status parameter indicates that the requested service was performed successfully. This parameter specifies the current value of the selected variable. A part of the observational DLE-variables and the range of vales are shown for the information in Table 13.

Table 13 – Observational variables and the range of values

Operating parameter		
Variable-name	**Description**	**Range of value**
ATSYN	The observed time period from SYN packet arrival to SYN packet arrival	Value is indicated in octet time.
ATTRT2	The observed time period of the TTRT2	Value is decremented in the range of 2^{16}-1 to 1.
ATTRT1	The observed time period of the TTRT1	Value is decremented in the range of 2^{16}-1 to 1.
ATTRT0	The observed time period of the TTRT0	Value is decremented in the range of 2^{16}-1 to 1.
ATHT	The observed time period of the MTHT	Value is decremented in the range of MTHT to 0.
ASL	The observed time period of the SL	Value is decremented in the range of V(SL) *1 to 0.

Table 13 – *(continued)*

Operating parameter		
Variable-name	**Description**	**Range of value**
ARMGP	The observed time period for one receive-channel, which has already completed one packet received and has waited for IGP time (0.96μsec), to wait the completion of a packet received on the other receive-channel in order to detect the other receive-channel disrupted.	Value is decremented in the range of 250 to 0. Value 250 is equal to 10 μs.
ASCMP	Observed repetitive count of the CMP packet sent out by SYN node in substitution for the corresponding node which has missed to send out the CMP packet. The number counted is indicated and is incremented coincidentally at each DLME.	Value 1 to 16. Default value is 3 and is equal to the value of SCMPL.
NONC	Permissible repetitive count of no CMP packet received by the SYN node within the corresponding consecutive Tsyn cycles, that is 256 times by SCMPL, in order to detect no other node except the SYN node in the TCnet domain.	Value 1 to 16. Default value is 3.
ROK_A, ROK_B	Cumulative count of DLPDU received without error on the receive-channel A or B.	Value is in the range of 2^{32}-1 to 0.
NCD_A, NCD_B	Cumulative count of Non-Carrier detected on the receive-channel A or B. As to NCD_A it is incremented at the time Non-Carrier happens on the receive-channel A while Carrier occurs on the receive-channel B. NCD_B is as well.	Value is in the range of 2^{32}-1 to 0.
RE_A, RE_B	Cumulative count of DLPDU received error on the receive-channel A or B. In the case of RE_A, it is incremented at the time there happens error on the receive-channel A while on the receive-channel B there is no error happens.	Value is in the range of 2^{32}-1 to 0.
CDh	Cumulative count of High-speed-cyclic data packet sent out on the medium.	Value is in the range of 2^{32}-1 to 0.
CDm	Cumulative count of Medium-speed-cyclic data packet sent out on the medium.	Value is in the range of 2^{32}-1 to 0.
CDl	Cumulative count of Low-speed-cyclic data packet sent out on the medium.	Value is in the range of 2^{32}-1 to 0.
SD	Cumulative count of Sporadic data packet sent out on the medium.	Value is in the range of 2^{32}-1 to 0.
LL	Live list indicating whether or not the corresponding node, at this moment, is connecting to and running in the TCnet domain in the received SYN frame.	True: Connecting and working. False: not connecting and working. Live list is a collection of 8 words of 32 bits length, of which bit is corresponding to each node in the TCnet domain and is indicating the current status in operation or not.
LN	Extracted number from LL at each node and used by each node to decide whether the node is able to send the data frame out over the medium.	Value 1 to 255.
CDH_{blk}	Information data block of "$_{blk}$" indicating the corresponding time-critical cyclic data being active(health) or inactive(un-healthy)	True: Active False: inactive

NOTE 1 SL(Silence time) for detecting the current SYN node out of order, following to initiate sending the CLM packet out from the nodes which are designated to be a SYN node.

6.8.4.4 Event

6.8.4.4.1.1 Function

This service is used to inform the DLMS-user about certain events or errors in the DLL.

6.8.4.4.1.2 Type of primitives and parameters of DLM-Event

Table 14 indicates the primitive and parameters of the DLM-Event service.

Table 14 – Event primitives and parameters

DLM-Event	**Indication**
Parameter name	output
DLM-event-identifier	M
Additional-information	C

6.8.4.4.1.2.1 DLM-event-identifier

This parameter specifies the primitive or composite event within the DLE whose occurrence is being announced. The possible values are defined in the corresponding part of clause 7 Data Link Layer Protocol specification. Though a part of the state change parameters related to various events are shown for information in Table 15.

Table 15 – Event related state change variables

Operating parameters		
Variable name	**Description**	**Additional-information**
RCS_A, RCS_B	Receive-channel A, Receive-channel B selected for receiving packets	True: the receive-channel A or B selected for receiving packets respectively. False: Not selected for receiving packets on channel A or B respectively.
SEN_A, SEN_B	Transmitter A, Transmitter B enabled to send out packets	True: Enabled to send out packets from Transmitter A or B respectively. False: Disabled to send out packets from Transmitter A or B respectively. Default value is "True".
IS	IN or out of service of total node operation	True: In service False: Out of service
FE	Fatal error occurred	True: Fatal error occurred and system entered into out of service. False: under normal operation
TCIS	IN or out of service of Time-critical Cyclic data transmission	True: In service of Time-critical cyclic data transmission False: out of Time-critical cyclic data transmission service
CRCC	Receive-channel changed	True: Receive-channel changed. Receiving channel can be checked out using RCS_A and RCS_B. Is this indicator momentary action? If not when / how is it changed to False state?

Table 15 – *(continued)*

Operating parameters		
Variable name	**Description**	**Additional-information**
$LINK_A$, $LINK_B$	Ethernet link Channel A or B is running or not.	True: Ethernet link is running. False: Ethernet link is out of service.
CTCC	Transmit-channel changed	True: Transmit-channel changed. Transmitting channel can be checked out using SENA and SENB. Is this indicator momentary action? If not when / how is it changed to False state?
CLLC	Live-list content changed	True: LL content changed.
DDLE	Duplicate DLEs detected with same node identifier number of TN.	True: Duplicate DLEs detected with same node identifier number of TN. False: Only one DLE with this number of TN.
CCDHS	State, indicating active (healthy) or inactive (unhealthy) of the corresponding time-critical cyclic data, is changed. State is reflected on the corresponding information data block "CDH_{blk}"	True: State changed active from/to inactive. False: No state changed.

6.8.4.4.1.2.1.1 Additional-information

This optional parameter provides event-specific additional information.

6.8.4.5 Set Publisher Configuration

6.8.4.5.1 Function

This service allows the DLMS-user to set up the Publisher configuration information related to the DLCEP in the DLL. The DLMS-user receives the confirmation when this service has been completed.

6.8.4.5.2 Type of primitives and parameters of Set Publisher Configuration

Table 16 indicates the primitive and parameters of the Set Publisher configuration service.

Table 16 –Set Publisher Configuration primitives and parameters

DLM-Set-Publisher-Configuration	**Request**	**Confirm**
Parameter name	input	output
Desired-Speed-class	M	
Desired-configuration	M	
Status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

6.8.4.5.2.1 Desired-Speed-class

This parameter specifies the speed class for the Time-critical Cyclic data transmission. The DMLS-user can specify High-speed, Medium-speed or Low-speed for the speed class.

a) High-speed

b) Medium-speed

c) Low-speed

6.8.4.5.2.2 Desired-configuration

This parameter specifies a set of the DLCEP-identifiers as a publisher for the Time-critical Cyclic data transmission with the desired–Speed-class. The DLCEP-identifiers may be specified within the range of 0 to 2047, though the maximum number can be specified in the limitation of $V(PB_h)$, $V(PB_m)$ or $V(PB_l)$ for the corresponding speed-class of High-speed, Medium-speed or Low-speed respectively.

6.8.4.5.2.3 Status

This parameter allows the DLMS-user to determine whether the requested service was provided successfully, or failed for the reason specified. The value conveyed in this parameter is as follows:

a) “OK –success – the Publisher configuration is updated"

b) “Failure – terminated unsuccessfully"

c) "Failure – invalid parameters in the request".

6.8.4.6 Get Publisher Configuration

6.8.4.6.1.1 Function

This option service allows the DLMS-user to read the Publisher configuration information in the DLL. The DLMS-user receives the confirmation with the current Publisher configuration when this service has been completed.

6.8.4.6.2 Type of primitives and parameters of Get Publisher Configuration

Table 17 indicates the primitive and parameters of the Get Publisher configuration service.

Table 17 – DLM-Get-Publisher-Configuration primitives and parameters

DLM-Get-Publisher-Configuration	**Request**	**Confirm**
Parameter name	input	output
Desire-speed-Class	M	
Status		M
Current-configuration		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

6.8.4.6.2.1 Desire-speed-Class

This parameter specifies the speed class for the Time-critical Cyclic data transmission. The DMLS-user can specify High-speed, Medium-speed or Low-speed for the speed class.

a) High-speed

b) Medium-speed

c) Low-Speed

6.8.4.6.2.2 Status

This parameter allows the DLMS-user to determine whether the requested service was provided successfully, or failed for the reason specified. The value conveyed in this parameter is as follows:

a) “Success – the current configuration could be read"

b) “Failure – the current configuration does not exist or terminated unsuccessfully”

c) "Failure – invalid parameters in the request"

6.8.4.6.2.3 Current-Configuration

This parameter is present when the status parameter indicates that the requested service was performed successfully. This parameter indicates a set of the current values of the DLCEP-identifiers for the Time-critical Cyclic data transmission of the corresponding desired–Speed-class.

6.8.4.7 Activate Time-critical Cyclic data transmission

This service allows the DLMS-user to activate the Time-critical Cyclic data transmission service. The DLMS-user receives the confirmation whether the designated Time-critical Cyclic data transmission has been started or this request has been terminated unsuccessfully.

6.8.4.7.1 Type of primitives and parameters of DLM-Activate-TCC

Table 18 indicates the primitive and parameters of the DLM-Activate-TCC service.

Table 18 – DLM-Activate-TCC primitives and the parameters

DLM-Activate-TCC	Request	Confirm
Parameter name	input	output
Desired-speed-class	M	
Status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

6.8.4.7.2 Desire-speed-Class

This parameter specifies the speed class for the Time-critical Cyclic data transmission. The DMLS-user can specify High-speed, Medium-speed or Low-speed for the speed class.

a) High-speed

b) Medium-speed

c) Low-Speed

6.8.4.7.3 Status

This parameter allows the DLMS-user to determine whether the requested service was provided successfully, or failed for the reason specified. The value conveyed in this parameter is as follows:

a) "Success – the Time-critical Cyclic data transmission is activated"

b) "Failure – terminated unsuccessfully"

6.8.4.8 Deactivate Time-critical Cyclic data transmission

This service allows the DLMS-user to deactivate Time-critical Cyclic data transmission service. The DLMS-user receives the confirmation whether the designated Time-critical Cyclic data transmission has been terminated or this request has been completed unsuccessfully.

6.8.4.8.1 Type of primitives and parameters of DLM-Deactivate-TCC

Table 19 indicates the primitive and parameters of the DLM-Deactivate-Publisher service.

Table 19 – DLM-Deactivate-TCC primitives and the parameters

DLM-Deactivate-TCC	**Request**	**Confirm**
Parameter name	input	output
Desired-speed-class	M	
Status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

6.8.4.8.2 Desire-speed-Class

This parameter specifies the speed class for the Time-critical Cyclic data transmission. The DMLS-user can specify High-speed, Medium-speed or Low-speed for the speed class.

a) High-speed

b) Medium-speed

c) Low-Speed

6.8.4.8.3 Status

This parameter allows the DLMS-user to determine whether the requested service was provided successfully, or failed for the reason specified. The value conveyed in this parameter is as follows:

a) "Success – the Time-critical Cyclic data transmission is deactivated "

b) "Failure – terminated unsuccessfully"

7 Data Link Layer Protocol specification

7.1 Introduction

The Data Link Protocol of RTE-TCnet is specified in this section. This section of the Data Link Protocol is to be one of type of IEC 61158-4 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, which is based in part on the Basic Reference Model for Open Systems Interconnection. Both Reference Models subdivide the area of standardization for interconnection into a series of layers of specification, each of manageable size.

This section of the Data Link Protocol provides the Data Link Service specified in section 6 by making use of the services available from Fieldbus Physical Layer. This section defines the Data Link Protocol functions and the characteristics for the interactions between the peer nodes. The relationship between the International Standards for Fieldbus Data Link Service, Fieldbus Data Link Protocol, Fieldbus Physical Service and Systems Management is illustrated in Figure 11.

NOTE Systems Management, as used in this PAS, is a local mechanism for managing the layer protocols.

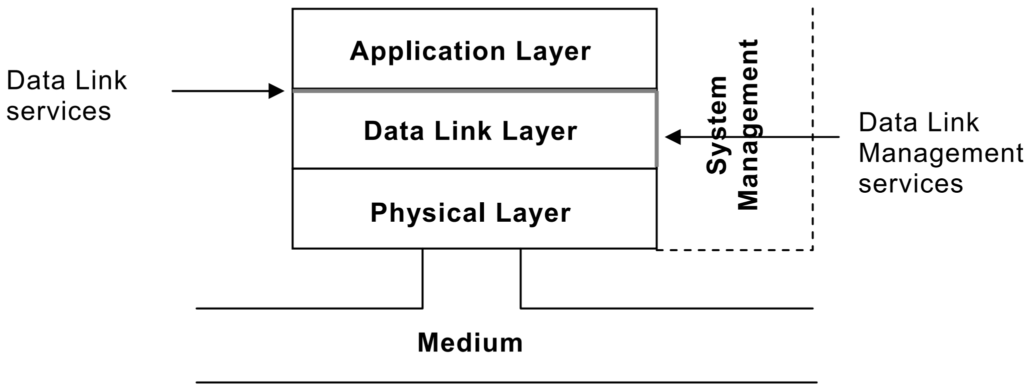


Figure 11 – Relationship of RTE-TCnet Data Link Protocol to other RTE layers and to users of RTE Data link service

The primary aim of this section is to provide a set of rules for communication expressed in terms of the procedures to be carried out by peer Data Link entities (DLEs) of RTE-TCnet 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:

a) As a guide for implementers and designers;

b) For use in the testing and procurement of equipment;

c) As part of an agreement for the admittance of systems into the open systems environment;

d) As a refinement to the understanding of time-critical communications within OSI

This section is concerned, in particular, with the communication and interworking of sensors, effectors and other automation devices. By using the Data Link Protocol of RTE-TCnet together with other standards positioned within the OSI or Fieldbus Reference Models, otherwise incompatible systems may work together in any combination.

7.2 Scope

7.2.1 Overview

The Data Link Layer of RTE-TCnet provides basic time-critical messaging communications between devices in an automation environment. 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.

The purpose of this section is to define the Data Link Protocol that is to be one of distinct and non-interoperable Fieldbus Data Link Protocols of IEC 61158-4. This Data Link Protocol of RTE-TCnet is related to, and lies within the field of application of, the corresponding Data Link Service Definition defined in section 6.

The maximum system size RTE-TCnet is an unlimited number of links of 255 nodes, each with 2 DLSAP-addresses and related peer and publisher DLCEPs, another 4 group DL-addresses per link, and 2048 peer and publisher DLCEPs per link which can be allocated among the link’s nodes.

7.2.2 Specifications

This PAS specifies:

a) The procedures for the timely transfer of data from one data-link user entity to a peer user entity and multi-peer user-entities, and for the transfer of control information among the data-link entities forming the distributed data-link service provider
b) The procedures for giving communications opportunities to all participating data-link entities, sequentially and in a cyclic manner for deterministic and synchronized transfer at the cyclic intervals up to 1millisecond
c) The procedure for giving communication opportunities available for time-critical data transmission together with non-time-critical data transmission without prejudice to the time-critical data transmission
d) The procedure for giving communication opportunities for time-critical data transmission in a cyclic and acyclic with prioritized access
e) The procedure for giving communication opportunities based on ISO/IEC 8802-3 medium access control. There are provisions for node to be added or removed during normal operation.
f) The structure of the Data Link Protocol Data Units used for the transfer of data and control information, and their representation as Physical Interface Data Units

7.2.3 Procedures

The procedures are defined in terms of

a) The interactions between peer DL-entities through the exchange of the Data Link Protocol Data Units
b) The interactions between a DLS-provider and a DLS-user in the same system through the exchange of DLS primitives
c) The interactions between a DLS-provider and a Physical Service provider in the same system through the exchange of Ph-service primitives

These procedures are applicable to instances of communication between systems that support time-critical communications services within the Data Link layer of the OSI or Fieldbus Reference Models, and which require the ability to interconnect in an open systems interconnection environment.

7.2.4 Applicability

These procedures are applicable to instances of communication between systems that support time-critical communications services within the Data Link layer of the OSI or Fieldbus Reference Models, and which require the ability to interconnect in an open systems interconnection environment.

7.2.5 Conformance

This PAS also specifies conformance requirements for systems implementing these procedures. There are not contained tests to demonstrate compliance with such requirements.

7.3 Void

7.4 Terms and definitions

7.4.1 DLSAP
Distinctive point at which DL-services are provided by a single DL-entity to a single higher-layer entity

7.4.2 DL(SAP)-address
Either an individual DLSAP-address, designating a single DLSAP of a single DLS-user, or a group DL-address potentially designating multiple DLSAPs, each of a single DLS-user

7.4.3 node
Single DL-entity as it appears on one local link

7.4.4 SYN node
Node transmitting SYN frame

7.4.5 receiving DLS-user
DL-service user that acts as a recipient of DL-user-data

NOTE A DL-service user can be concurrently both a sending and receiving DLS-user

7.4.6 sending DLS-user
DL-service user that acts as a source of DL-user-data

7.4.7 DLCEP-address
DL-address which designates either

a) One peer DL-connection-end-point
b) One multi-peer publisher DL-connection-end-point, and implicitly the corresponding set of subscriber DL-connection-end-points where each DL-connection-end-point exists within a distinct DLSAP and is associated with a corresponding distinct DLSAP-address

7.4.8 multi-peer DLC
Centralized multi-end-point DL-connection offering DL-duplex-transmission between a single distinguished DLS-user, known as the publisher or publishing DLS-user, and a set of peer but undistinguished DLS-users, known collectively as the subscribers or subscribing DLS-users, where the publishing DLS-user can send to the subscribing DLS-users as a group (but not individually), and the subscribing DLS-users can send to the publishing DLS-user (but not to each other)

7.4.9 node DL-address
DL-address which designates the (single) DL-entity associated with a single node on a specific local link

7.4.10 node-id
Two-octet primary identifier for the DLE on the local link, whose values are constrained. A permissible value is from 1 to 255. A value 0 is specifically used for SYN node, which emits SYN frame.

7.4.11 slot-time
The 512 bit time of the physical signalling symbol specified in ISO/IEC 8802-3 MAC specification.

7.4.12 token
Right to transmit on the local link

7.4.13 FCS error
Error that occurs when the computed frame check sequence value after reception of all the octets in a DLPDU does not match the expected residual

7.4.14 implicit token
Mechanism that governs the right to transmit, moreover no actual token message is transmitted on the medium. Each node keeps track of the node that it believes currently holds the right to transmit. The right to transmit is passed from node to node by keeping the node that last transmitted. A slot time is used to allow a missing node to be skipped in the rotation

7.4.15 multipoint connection
Connection from one node to many nodes. Multipoint connection allows data transfer from a single publisher to be received by many subscriber nodes.

7.4.16 data DLPDU
DLPDU that carries a DLSDU from a local DLS-user to a remote DLS-user

7.5 Symbols and abbreviations

7.5.1 Common symbols and abbreviations

7.5.1.1 Data units

7.5.1.1.1 DLPDU DL-protocol data unit

7.5.1.1.2 DLSDU DL-service data unit

7.5.1.1.3 PhIDU Ph-interface data unit

7.5.1.2 FCS Frame Check Sequence

7.5.1.3 ind indication primitive

7.5.1.4 MAC medium access control

7.5.1.5 FIFO First-in first-out (queuing method)

7.5.1.6 PLS Physical Signalling

7.5.1.7 req request primitive

7.5.1.8 RX receive

7.5.1.9 SFD Start Frame Delimiter

7.5.1.10 TX transmit

7.5.2 RTE-TCnet: Additional symbols and abbreviations

7.5.2.1 ACM Access Control Machine

7.5.2.2 CLM Claim frame

7.5.2.3 CTRC Cyclic-transmission TX/RX Control

7.5.2.4 FC Frame Control field

7.5.2.5 F-Type Frame type

7.5.2.5.1 CLM Claim frame

7.5.2.5.2 CMP Transmission Complete frame

7.5.2.5.3 COM Command frame

7.5.2.5.4 DT Cyclic Data frame

7.5.2.5.5 DT-CMP DT with transmission complete frame

7.5.2.5.6 RAS RAS frame

7.5.2.5.7 REQ In-ring Request frame

7.5.2.5.8 SYN Synchronization frame

7.5.2.6 Parameters of SYN DLPDU

7.5.2.6.1 CW Control word

7.5.2.6.2 LL Live list

7.5.2.6.3 PM Periodic Mode

7.5.2.6.4 RMSEL Redundant medium selection

7.5.2.6.5 PN Transmission Permits Node Number

7.5.2.6.6 ST Slot time

7.5.2.7 Pri Priority field

7.5.2.8 RMC Redundancy Medium Control

7.5.2.9 SN Source Node number field

7.5.2.10 Transmission period

7.5.2.10.1 Th High-speed transmission period

7.5.2.10.2 Tm Medium-speed transmission period

7.5.2.10.3 Ts Sporadic message transmission target-token-rotation-time period

7.5.2.10.4 Tl Low-speed transmission period

7.6 Overview of the DL-protocol

7.6.1 Overview of the media access control

The RTE-TCnet has a deterministic media access control in order to avoid collisions that occur when a number of the nodes send data frames simultaneously, and to provide the opportunity of sending data to each node in a sequential order and within a predetermined time period. Figure 12 shows the basic principle of media access control of the RTE-TCnet.

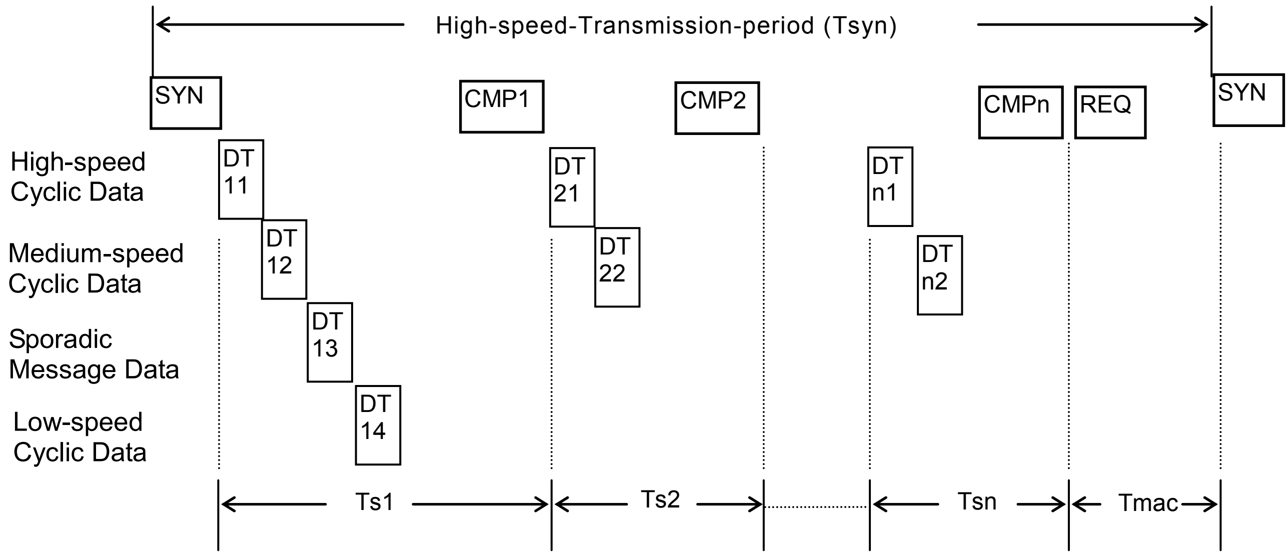


Figure 12 – Basic principle of Media Access Control

At the start time of every High-speed-Transmission-period (Tsyn), the SYN frame is broadcasted to all nodes. Receiving the SYN frame, the node with sequential number 1 starts sending its data frames, and after that broadcasts its CMP frame in order to indicate the completion of its data frames transmission. The Nth node can send out its data frames after receiving the CMP frame from the (N-1) node. After all the nodes send their data frames, the time period to solicit new nodes begins. The REQ frame is used for a new node requesting inclusion to the RTE-TCnet network. The sequential number is assigned to a new node at the time approval to join is granted.

Each node can hold the transmission right for a preset time and must send its CMP frame to transfer the transmission right to the next node within the preset time. The data to be sent and the data to be held over are determined by priority.

Transmission includes Time-critical Cyclic data and sporadic Ethernet message transmission. Cyclic data transmission is divided into High, Medium and Low-speed data transmission. Each node sends the High-speed Cyclic data frames on each occasion when it obtains the transmission right. The data of lower priorities, that is the Medium-speed Cyclic data, the sporadic Ethernet message data and the Low-speed Cyclic data respectively, is sent or not sent depending on the circumstances. The holding time of the transmission right of each node is determined by the settings of the High-speed Cyclic, the Medium-speed Cyclic, the sporadic Ethernet message and the Low-speed Cyclic data transmission periods and by the volume of transmission data for each node. After sending all the High-speed Cyclic data, the node sends the Medium-speed Cyclic data. If the holding time of the transmission right ends during sending the Medium-speed Cyclic data, the transmission of the Medium-speed Cyclic data is interrupted, and the CMP frame is sent out. The n^{th} node obtains the transmission right again during the next High-speed data transmission period, during which time all the High-speed Cyclic data and the remainder of the previous Medium-speed Cyclic data is sent. Tmac is the period for a new node sending out REQ frame to enter the network.

7.6.2 Service assumed from the PhL

This subclause describes the assumed Physical Service (PhS) and the constraints on use by the RTE-TCnet DLE. The Physical Service is assumed to provide the following service primitives specified by ISO/IEC 8802-3 "Physical Signaling (PLS) service specifications".

7.6.2.1 Assumed primitives of PhS

The PhS is assumed to provide the following two categories of primitives to the DLL of the RTE-TCnet.

1) Service primitives used for transmitting and receiving frames to / from all other peer DLE.
2) Service primitives provide information needed by the local DLE to perform the media access functions.

The assumed primitives of PhS are grouped into these two categories:

1) Transfer of Data to all other peer DLE
    a) PLS_DATA request
    b) PLS_DATA indication
2) Media access management by local DLE
    a) PLS_CARRIER indication
    b) PLS_SIGNAL indication
    c) PLS_DATA_VALID indication

The interaction of the PhS primitives to the DLE is shown in Figure 13.

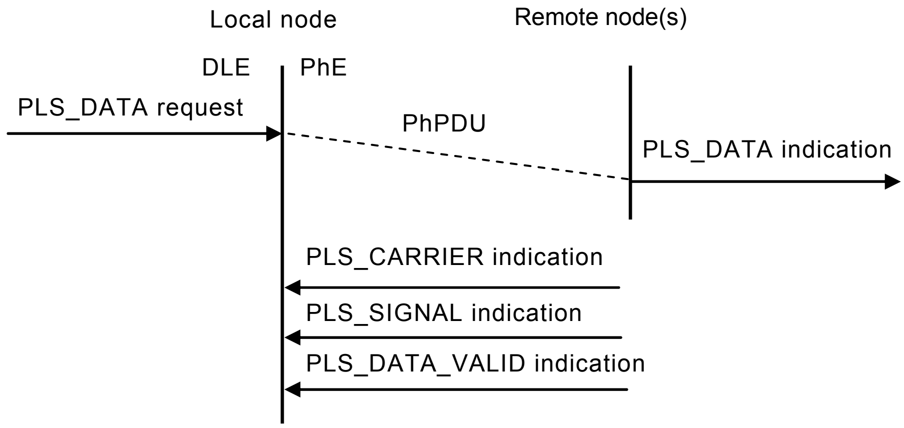


Figure 13 – Interaction of PhS primitives to DLE.

7.6.3 DLL architecture

The RTE-TCnet DLL is modeled as a combination of control components of Access Control Machine (ACM), Cyclic transmission TX/RX Control(CTRC), Sporadic TX/RX Control(STRC), Redundancy Medium Control (RMC), Serializer, Deserializer and DLL management interface.

The Access Control Machine as the primary control component provides the function for deterministic media access control cooperating with the Cyclic-transmission TX/RX Control, the Sporadic TX/RX Control and the Redundancy Medium Control for reliable and efficient support of higher-level connection-mode and connectionless data transfer services. Specifically the Access Control Machine has the primary responsibility for;

a) Assuring that the local node detects and fully utilizes its assigned access time period.
b) Assuring that the local node does not interfere with the transmissions of other nodes, especially of the node transmitting the SYN frame.
c) Detecting network disruption, and initiating the SYN frame transmission for restoration of the network disruption from after prescribed time duration in which the SYN frame is not heard.
d) Assuring a new node adding to and removing from the network

The DLL management interface provides DLL management functions. PhL framing and delimiters are managed by DLL functions for serializing and deserializing M_symbol requests and indications.

The Data Link Layer is comprised of the components listed in Table 20.

Table 20 – Data Link Layer components

Components	Description
Access Control Machine (ACM)	Deterministic media access control and scheduling the opportunities to sending out the DLPDUs, control for adding and removing nodes, restoration from disruption. Assembles and transmits the DLPDUs to the TX framer in the RMC, receives and disassembles the DLPDUs with the control information from the RX framer in the RMC, and determines the timing and duration of the transmissions.
Cyclic transmission TX/RX Control (CTRC)	Buffers and dispatches in time DLSDU received for the Time-critical Cyclic data transfer between the DLS-user and the ACM.
Sporadic TX/RX Control (STRC)	Buffers and dispatches in time DLSDU received for the Sporadic message data between DLS-user and the ACM.
Redundancy Medium Control (RMC)	Receives the DLPDUs from the ACM and breaks them down into octet symbol requests to the Octet Serializer, assembles received octets from Octet Deserializer into DLPDUs and submits them to the ACM. Select one of two outputs of the Octet Deserializers for medium redundancy.
TX/RX Framer	Receives the octet symbols from RMC, detects and indicates the start timing of a DLPDU to Octet Serializer, passes received octets from Octet Deserializer and indicates the start timing of a DLPDU to RMC.
Octet Serializer	Receives octet symbols, encodes and serializes them, and sends them as M_symbols to the PhL. It is also responsible for generating the FCS.
Octet Deserializer	Receives M_symbols from the PhL, converts M_symbols into octets and sends them to the receive machine. It is also responsible for checking the FCS.
DLL Management Interface	Holds the Station Management variables that belong to the DLL, and manage synchronized changes of the link parameters.

The internal arrangement of these components, and their interfaces, are shown in Figure 14. The arrowheads illustrate the primary direction of flow of data and control.

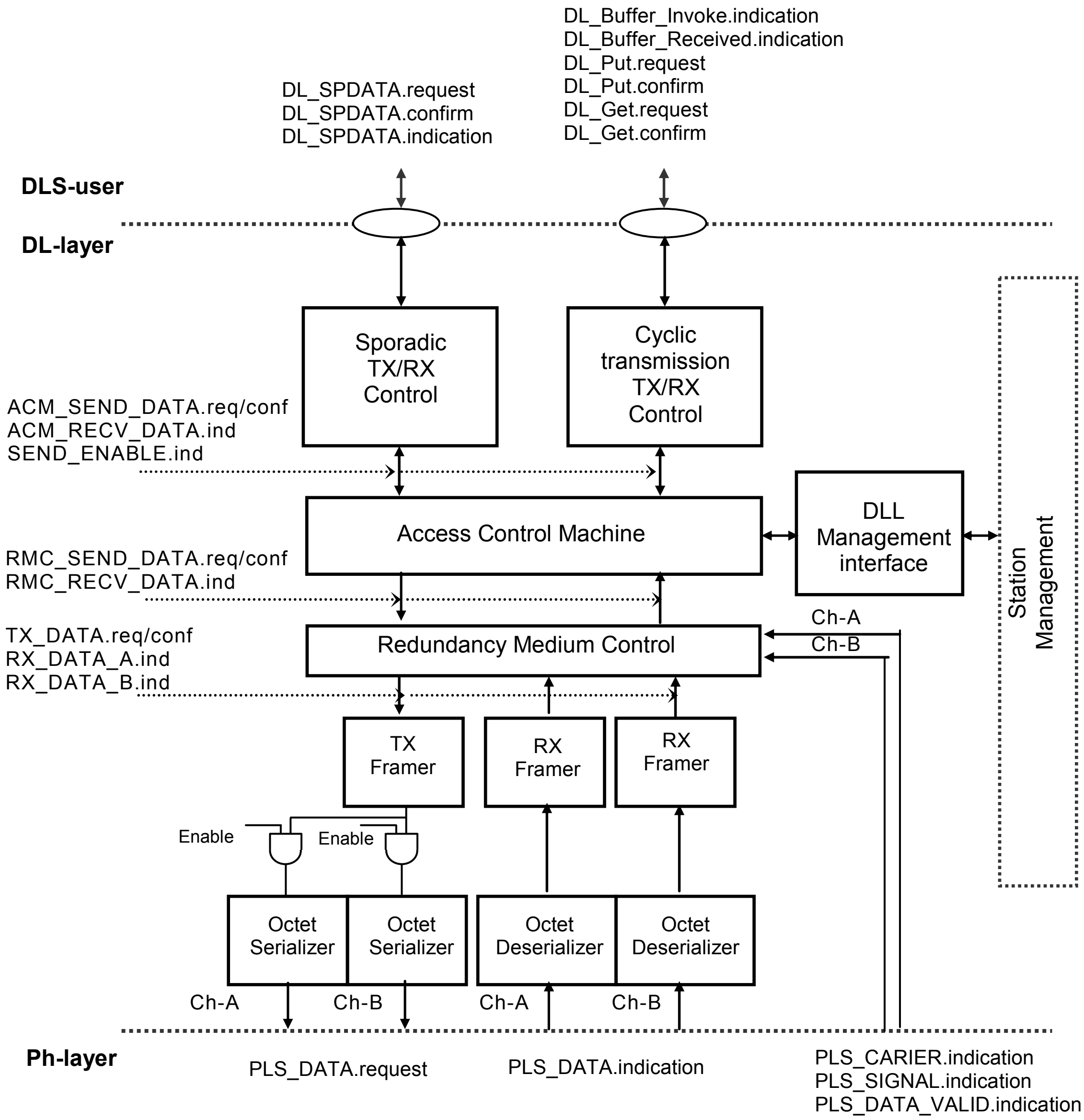


Figure 14 – Data Link layer internal architecture

7.6.4 Access Control Machine and schedule support functions

The ACM functions schedule all communications between the DLEs participating in the RTE-TCnet, and the timing of this communications is controlled as to;

1) fulfill the specific media access control to give all the DLEs the opportunities to send out 2 kinds of class of Time-critical Cyclic data and Sporadic message data in timely, prioritized and deterministic fashion, and to detect network disruption and to initiate the restoration in appropriate time, further to add and remove nodes on line.

2) provide 3 levels of Time-critical data transfer opportunities of sending data to node in sequential order and within each pre-specified time period, and that the data transfer of each level is performed within the pre-specified time duration (token holding time) and whether the data transfer of lower levels to be carried out or to be held over to

later cyclic time period depends on the level and the occasion though the top level of the data transfer is always carried out at every occasion, on the other hand the whole volume of the data transfer of lower levels is transferred within each pre-specified time period.

3) provides sporadic message data transfer opportunities of sending out to node that the request to transfer is happened sporadically by the DLS-user, and the data transfer is performed in pre-specified time period of the corresponding level of priority and is based on regular ISO/IEC 8802-3 applications.

These accurate scheduling timing is very important to support many control and data collection tasks in the applications domain of this protocol.

7.6.5 Local parameters, variable, counters, timers

This PAS uses DLS-user request parameters P(…) and local variables V(…) as a means of clarifying the effect of certain actions and the conditions under which those actions are valid, local timers T(…) as a means of monitoring actions of the distributed DLS-provider and of ensuring a local DLE response to the absence of those actions, and local counters C(…) for performing rate measurement functions. It also uses local queues Q(…) as a means of ordering certain activities, of clarifying the effects of certain actions, and of clarifying the conditions under which those activities are valid.

Unless otherwise specified, at the moment of their creation or of DLE activation,

a) All variables shall be initialized to their default value, or to their minimum permitted value if no default is specified;

b) All counters shall be initialized to zero;

c) All timers shall be initialized to inactive;

d) All queues shall be initialized to empty.

DL-management may change the values of configuration variables.

7.6.5.1 Variables, Parameter, Counter, Timer and Queues to support DLE function

7.6.5.1.1 V(DR): Data-rate

The value of this variable indicates the data signalling rate in Mbps. The value is 100.

7.6.5.1.2 V(TN), P(TN): This-node

This variable holds and designates node identifier number of this node. The initial value is 0 on power-up or reset of this node. The range of this value is 1 to 255.

7.6.5.1.3 V(Tsl): DLE number able to send out next slot

This variable is used by the DLE to hold and indicate the node identifier number of V(TN) of which node can send out its data next in the sequential order in succession to a node currently having completed to send out its all data and CMP or DT-CMP DLPDU. The value of this variable is updated and set to the value of V(TN) of the node next to send out, every time on the reception of CMP or DT-CMP DLPDU. The range of this variable is 1 to 255.

7.6.5.1.4 V(MN), P(MN): Maximum-node-number

This variable holds the maximum node number, and is set by DLMS. Default value of this variable is 254.

7.6.5.1.5 V(IA): Individual-address-of-this-node

This variable holds individual address of this node in 48 bits length specified by ISO/IEC 8802-3. The vale is set by DLMS.

7.6.5.1.6 V(MD), P(MD): Maximum-distance

The value of this variable holds, and is set by DLMS, the maximum distance in kilo meter on the connection path between any 2 nodes. The range of this value is 1 to 100, and the default value is 8.

7.6.5.1.7 V(MDD), P(MDD): Maximum-difference-of-the-distance

This variable holds, and is set by DLMS, maximum difference in meter of the distance of two redundant physical mediums on the connection path between any 2 nodes. The range of this value is 0 to 2000, and the default value is 500.

7.6.5.1.8 V(MPD), P(MPD): Maximum-distance-of-the-signal-propagate-delay

This variable holds, and is set by DLMS, maximum difference in microsecond of the signal delay time propagating over two redundant physical mediums on the connection path between any 2 nodes. The range of this value is 1 to 0x7c, and the default value is 0x7c which is equivalent to approx. 5 microsecond, i.e. 0x7c (=124) x 0,04 microsecond.

7.6.5.1.9 V(MRT), P(MRT): Maximum-number-of-the-repeater-unit

This variable holds, and is set by the DLMS, maximum number of the repeater units on the connection path between any 2 nodes. The possible value is 0 to 7, and the default vale is 3.

7.6.5.1.10 V(RCS), P(RCS): Receive- channel switching control for receiving packets

This variable holds and designates the switching control for receiving packets out of one of two receive-channel A and B corresponding to each of the redundant medium A and B.

The possible value is “Automatic”, “Force A” or “Force B”. “Automatic” indicates that the switching control for receiving packets is automatically to switch to the proper receive-channel. “Force A” or “Force B” indicates that the switching control is forced to switch to Receive-channel A or B respectively. The default value is “Automatic”.

7.6.5.1.11 V(BW), P(BW): Length of Time-critical Cyclic Data word in a DLPDU

This variable holds and designates the length of Time-critical cyclic data ward in a DLPDU. The possible value of the length in octet is 32 to 128, and the default value is 128.

7.6.5.1.12 V(IP), P(IP): IP address of this node

This variable holds and designates DLSAP value assigned for TCnet Sporadic message data service.

7.6.5.1.13 V(SM), P(SM) : Subnet address mask

This variable holds and designates the value of subnet address mask for V(IP) for TCnet Sporadic message data service.

7.6.5.1.14 V(ST), P(ST), T(ST): RTE-TCnet Slot-time

Slot-time is the fundamentally observational time unit, using in the DLME for observing to initiate action such for re-initialisation in sending out CLM packet, for sending out the CMP packet specifically by the current SYN node in substitute for the node missed to send out the CMP packet.

This variable holds and designates the fundamentally observational time value of slot time. The rage of the value is 1 to 255, of which time is in 512 bits physical symbol time specified by and identical to ISO/IEC 8802-3. The default value is 20, which is equivalent to approx.100 microsecond, i.e. 20 X 512 x 1/100 microsecond.

NOTE Slot time value is calculated from the following equation.

V(ST)= round up [2 x V(MD) + V(MRT)/2 + 2]

7.6.5.1.15 V(TTRT2), P(TTRT2) : Target-token-rotation-time for access class 2

This variable holds and designates the value of target-token-rotation-time for access class 2, especially for medium-speed cyclic data transmission. The range of this value is 1 to 2^{16}-1 of which unit is in octet time by the data signalling rate. The default value is equal to P(TSYN). This value is used by the DLE, and should be equal and common to all DLEs of the RTE-TCnet.

P(TTR2) is used by the DLME, and the value is the expected time period in which all nodes with medium-speed cyclic data can obtain the transmission right and send out all of the medium-speed cyclic data.

7.6.5.1.16 V(TTRT1), P(TTRT1): Target-token-rotation-time for access class 1

This variable holds and designates the value of target-token-rotation-time for access class 1, especially for sporadic message data transmission. The range of this value is 1 to 2^{16}-1 of which unit is in octet time by the data signalling rate. The default value is equal to P(TSYN). This value is used by the DLE, and should be equal and common to all DLEs of the RTE-TCnet.

P(TTR1) is used by the DLME, and the value is the time period in which a node with sporadic message data can expect to obtain the transmission right to send out.

7.6.5.1.17 V(TTRT0), P(TTRT0): Target-token-rotation-time for access class 0

This variable holds and designates the value of target-token-rotation-time for access class 0, especially for low-speed cyclic data transmission. The range of this value is 1 to 2^{16}-1 of which unit is in octet time by the data signalling rate. The default value is equal to P(TSYN). This value is used by the DLE, and should be equal and common to all DLEs of the RTE-TCnet.

P(TTR0) is used by the DLME, and the value is the expected time period in which all nodes with low-speed cyclic data can obtain the transmission right and send out all of the low-speed cyclic data.

7.6.5.1.18 V(TSYN): Target-periodic-time of synchronization

This variable indicates the target time period from SYN packet arrival to SYN packet arrival. The value is equal to V(Th), and the value of P(Th) is used by the DLMS and is set to V(Th). The possible range of this time period is 0,1 to 160 in millisecond. V(SYN) and V(Th) indicate the cyclic time period in which all the node with high-speed cyclic data can obtain the transmission right to send out all of the high-speed cyclic data.

7.6.5.1.19 T(ATSYN): TSYN monitor

T(ATSYN) is used by the DLE to monitor the time period from SYN packet arrival to SYN packet arrival. The value is indicated in octet time by the data signalling rate.

7.6.5.1.20 T(ATTRT2): TTRT2 monitor

T(ATTRT2) is used by the DLE to monitor the time period of TTRT2. The value is decremented in the range of 2^{16}-1 to 1, of which unit is in octet time by the data signalling rate.

7.6.5.1.21 T(ATTRT1): TTRT1 monitor

T(ATTRT1) is used by the DLE to monitor the time period of TTRT1. The value is decremented in the range of 2^{16}-1 to 1, of which unit is in octet time by the data signalling rate.

7.6.5.1.22 T(ATTRT0): TTRT0 monitor

T(ATTRT0) is used by the DLE to monitor the time period of TTRT0. The value is decremented in the range of 2^{16}-1 to 1, of which unit is in octet time by the data signalling rate.

7.6.5.1.23 V(MTHT), P(MTHT): Maximum-token-hold-time

This variable holds and designates the value of maximum-token-hold-time for high-speed cyclic data transmission. The range of this value is 1 to 2^{16}-1 of which unit is in octet time by the data signalling rate. The default value is 0x30B4, that is approx. 900 µs.

7.6.5.1.24 T(ATHT): MTHT monitor

T(ATHT) is used by the DLE to measure the time elapsed of MTHT. The value is decremented in the range of P(MTHT) to 0.

7.6.5.1.25 V(IGP): Inter-packet-time-gap

This variable indicates the value of the time interval from the end of a previous packet to the start of the consecutive packet sent out of a node. The value is equivalent to 0,96 µs and is identical to the specification of ISO/IEC 8802-3.

7.6.5.1.26 T(AIGP$_A$), T(AIGP$_B$): IGP monitor over the receive-channel A and B

T(AIGP$_A$) and T(AIGP$_B$) are used by the DLE to measure the time elapsed of IGP over the receive-channel A and B respectively. The value is decremented in the range of 24 to 0, and value 24 is equivalent to 0,96 µs.

7.6.5.1.27 V(SL), P(SL): Silence time

This variable holds and designates the value of the silence time or inactivity time period for detecting the current SYN node out of order, following to initiate sending the CLM packet out from this node in order to claim SYN node if this node is designated and permitted to be a SYN node. The value is set by the DLE to the T(SL) timer as follows;

T(SL)=V(TSYN) + V(ST) x V(TN) x 2

7.6.5.1.28 T(ASL): SL monitor

T(ASL) is used by the DLE to measure the time elapsed of SL. The value is decremented in the range of V(SL) to 0.

7.6.5.1.29 V(RMGP), P(RMGP): IGP over other redundant media

This variable holds and designates the value of maximum time-interval for one receive-channel, which has already completed one packet received and has waited for IGP time, to wait the completion of a packet to be received on the other receive-channel in order to detect the other receive-channel disrupted. The default value is 250, of which unit is in 0,04 μs, and is equivalent to 10μsec.

7.6.5.1.30 T(ARMGP): RMGP monitor

T(ARMGP) is used by the DLE to monitor the time elapsed of RMGP. The value is decremented in the range of 250 to 0, and value 250 is equivalent to 10 μs.

7.6.5.1.31 V(TISPD), P(TISPD): Time-interval for Sporadic-message-data-service

This variable holds and designates the value of time interval cyclically processed for Sporadic-message-data service. The range of this value is 1 to 1000, of which unit is in 1 millisecond. Default value is 100.

7.6.5.1.32 T(TISPD): TISPD monitor

T(TISPD) is used by the DLE to measure the time interval of TISPD.

7.6.5.1.33 V(LL): Live-list

This variable indicates the current operational status, whether a corresponding node is connecting to and is running normal in the RTE-TCnet. The possible value is “True” or “False”, “True” means the node is connecting to and working normal and “False” is not. V(LL) is used by the DLE and is generated from the information conveyed by SYN frame. Live-list is a collection of 8 words of 32 bits length, each of which bit is corresponding to the node in the RTE-TCnet and indicates the current operational status. Each bit corresponds to the node number V(TN) in a sequential order from 0 to 255 in little endian format.

7.6.5.1.34 V(LN): Live-node-number

This variable indicates the TN number of node, extracted from V(LL), which is connecting to and are running normal in the TCnet at this point. V(LN) is used by the DLE to decide whether the node is able to send the data frame out over the medium. The range of this value is 1 to 255.

7.6.5.1.35 V(TMAC), P(TMAC): Medium-access-control-time to solicit new nodes

This variable holds and designates the value of maximum observational time period for the SYN node to solicit new nodes into RTE-TCnet. During the time period new nodes attempt to send out REQ packet to the SYN node. The default value is 100, and of which unit is in 0.1 millisecond.

7.6.5.1.36 T(TMAC): TMAC monitor

T(TMAC) is used by the DLE of SYN node to measure the time elapsed of TMAC in order to wait REQ packets sent out from node which attempts to enter RTE-TCnet. The value is decremented in the range of V(TMAC) to 0.

7.6.5.1.37 V(SCMP), P(SCMP): Medium-access-control-time for substitute CMP

This variable holds and designates the value of maximum observational time period for SYN node to initiate sending out CMP packet in substitution for the node which has missed to send out the CMP packet.

7.6.5.1.38 T(SCMP): SCMP monitor

T(SCMP) is used by DLE of SYN node to measure the time elapsed of SCMP to initiate sending out CMP packet in substitution for the node which has missed to send out the CMP packet. The value is decremented in the range of V(SCMP) to 0.

7.6.5.2 Variables, counter, timer and queues to support Time-critical cyclic data service

7.6.5.2.1 V(MGA), P(MGA): Multicast group address

This variable holds and designates the multicast group address, used by DLE to build logically associated group of RTE-TCnet nodes, in 48 bits length specified by and identical to ISO/IEC8802-3. The default value is 0x01-0x00-0x5e-0x50-0x00-0x01.

7.6.5.2.2 V(Th), P(Th): High-speed transmission period

This variable holds and designates the cyclic time period of High-speed time-critical data transmission. P(Th) is used by the DLE and the value is set to V(SYN). The range of this value is 1 to 160, of which unit is in 0.1millisecond. The default value is 100.

7.6.5.2.3 V(Tm), P(Tm): Medium-speed transmission period

This variable holds and designates the cyclic time period of Medium-speed time-critical data transmission. P(Tm) is used by the DLE. The range of this value is 10 to 1000, of which unit is in 1millisecond. The default value is 100.

7.6.5.2.4 V(Tl), P(Tl): Low-speed transmission period

This variable holds and designates the cyclic time period of Low-speed time-critical data transmission. P(Tl) is used by the DLE. The range of this value is 100 to 1000, of which unit is in 1 millisecond, and the default value is 100.

7.6.5.2.5 V(HTh), P(HTh): Maximum observational time period for detecting the continuous High-speed transmission cycle disrupted

This variable holds and designates the value of maximum observational time period for detecting the continuous High-speed transmission cycle disrupted. The value of this variable is in the range of 2 to 2^{16}-1, of which unit is in 1msec. The default value is 3 x V(Th).

7.6.5.2.6 T(HTh): HTh monitor

T(HTh) is used by the DLE to measure the time elapsed of HTh to detect the continuous High-speed transmission cycle disrupted. The value is decremented in the range of V(HTh) to 1.

7.6.5.2.7 V(HTm), P(HTm): Maximum observational time period for detecting the continuous Medium-speed transmission cycle disrupted

This variable holds and designates the value of maximum observational time period for detecting the continuous Medium-speed transmission cycle disrupted. The value of this variable is in the range of 2 to 2^{16}-1, of which unit is in 1msec. The default value is 3 x V(Th).

7.6.5.2.8 T(HTm): HTm monitor

T(HTm) is used by the DLE to measure the time elapsed of HTm to detect the continuous Medium-speed transmission cycle disrupted. The value is decremented in the range of V(HTm) to 1.

7.6.5.2.9 V(HTI), P(HTI): Maximum observational time period for detecting the continuous Low-speed transmission cycle

This variable holds and designates the value of maximum observational time period for detecting the continuous Low-speed transmission cycle disrupted. The value is in the range of 2 to 2^{16}-1, of which unit is in 1 ms. The default value is 3 x V(Th).

7.6.5.2.10 T(HTI): HTI monitor

T(HTI) is used by the DLE to measure the time elapsed of HTI to detect the continuous Low-speed transmission cycle disrupted. The value is decremented in the range of V(HTI) to 1.

7.6.5.2.11 V(RCS_A), V(RCS_B): Receive-channel A or B selected for receiving packets

This variable indicates the corresponding receive-channel is selected for receiving packets from other node. The value “True” indicates the corresponding receive-channel is selected, and “False” is not.

7.6.5.2.12 V(SEN_A) , P(SEN_A); V(SEN_B), P(SEN_B): Transmitter A or B enabled to send out packets

This variable designates the transmitter A or B enable to send out packets from, and indicates the state of the transmitter A or B being enabled or disabled. The value “True” is to be enabled and “False” is not for the corresponding Transmitter A or B. The default value is “True”.

7.6.5.2.13 V(SCMPL), P(SCMP): Permissible repetitive count of substitute CMP

This variable holds and designates the value of permissible repetitive count for CMP packet sent out from SYN node in substitution for the corresponding node which has missed to send out the CMP packet due to somewhat being malfunctioned. The range of this value is 1 to 16 and the default value is 3.

7.6.5.2.14 C(ASCMP): ASCMP count

C(ASCMP) indicates the number of observed repetitive count for CMP packet sent out by SYN node in substitution for the corresponding node which has missed to send out the CMP packet. The number of C(ASCMP) is incremented coincidentally at each node in RTE-TCnet. The value is in the range of 1 to V(SCMPL) and the default value is 3.

7.6.5.2.15 V(NONC), P(NONC): Permissible repetitive count of no CMP

This variable holds and designates the value of permissible repetitive count of no CMP packet received by the SYN node within the corresponding consecutive Tsyn cycles, that is 256 times by V(SCMPL), in order to detect no other node except the SYN node in the RTE-TCnet. The range of this value is 1 to 16, and the default value is 3.

7.6.5.2.16 C(NONC): NONC count

C(NONC) indicates the number of repetitive count of no CMP packet received by the SYN node within 256 x V(SCMPL) of the consecutive Tsyn cycles. C(NONC) reached to the number of V(NONC), the SYN node detects and recognizes no other node is hooking up to the RTE-TCnet.

7.6.5.2.17 C(ROK_A), C(ROK_B): Cumulative count of DLPDU received without error on the receive-channel A or B

C(ROK_A) or C(ROK_B) indicates the number of the cumulative count of DLPDU received without error on the receive-channel A or B respectively. C(ROK_A) and C(ROK_B) are roll-over binary counters of 32 bits length.

7.6.5.2.18 C(NCD_A), C(NCD_B): Cumulative count of Non-Carrier detected on the receive-channel A or B

C(NCD_A) or C(NCD_B) indicates the number of cumulative count of non-carrier detected on one receive-channel while the carrier occurs on the other side of receive-channel. C(NCD_A) and C(NCD_B) are roll-over binary counters of 32 bits length.

7.6.5.2.19 C(RE_A), C(RE_B): Cumulative count of DLPDU received with error on the receive-channel A or B

C(RE_A) or C(RE_B) indicates the number of cumulative count of DLPDU received with error while on the other side of receive-channel there is no error occurs. C(RE_A) and C(RE_B) are roll-over binary counters of 32 bits length.

7.6.5.2.20 C(CDh): Cumulative count of High-speed-cyclic data packet sent out

C(CDh) indicates the number of cumulative count of high-speed-cyclic data packet sent out on the medium. C(CDh) is a roll-over binary counter of 32 bits length.

7.6.5.2.21 C(CDm): Cumulative count of Medium-speed-cyclic data packet sent out

C(CDm) indicates the number of cumulative count of medium-speed-cyclic data packet sent out on the medium. C(CDm) is a roll-over binary counter of 32 bits length.

7.6.5.2.22 C(CDl): Cumulative count of Low-speed-cyclic data packet sent out

C(CDl) indicates the number of cumulative count of low-speed-cyclic data packet sent out on the medium. C(CDl) is a roll-over binary counter of 32 bits length.

7.6.5.2.23 C(SD): Cumulative count of Sporadic data packet sent out

C(SD) indicates the number of cumulative count of Sporadic data packet sent out on the medium. C(SD) is a roll-over binary counter of 32 bits length.

7.6.5.2.24 Q(QBF_{SPD}): Queue buffer for Sporadic data

Queue buffer for the Sporadic message data service holds the DLPDU by DL-user to be sent out. The Queue buffer size to transfer as well as to receive the Sporadic message data is implementation matter.

7.6.5.2.25 Q(BF_{blk}): Data buffer used for sending and receiving a DLPDU for time-critical cyclic data DLPDU

Total number of buffers is equivalent of the total number of DLCEP. Each buffer with the identifier "$_{blk}$" number is corresponding to a DLCEP with the identifier number "identifier".

7.6.5.2.26 Q(PBh): List of the data buffers using for sending out High-speed time-critical cyclic data

Max. number of the buffers is 2048.

7.6.5.2.27 Q(PBm): List of the data buffers using for sending out Medium-speed time-critical cyclic data

Max. number of the buffers is 2048.

7.6.5.2.28 Q(PBI): List of the data buffers using for sending out Low-speed time-critical cyclic data

Max. number of the buffers is 2048.

7.6.5.2.29 Q(CDH$_{blk}$): Information data block of "$_{blk}$" indicating the corresponding time-critical cyclic data being active(health) or inactive(un-healthy)

7.7 General structure and encoding of PhIDUs and DLPDU and related elements of procedure

7.7.1 Overview

The DLL and its procedures are those necessary to provide the services offered to DLS user by using the services available from the PhL. This clause describes the structure and semantics of PhIDU, MA_PDU, DLPDU and the procedure, commonly used in this PAS nevertheless this portion is identical to and fully compliant with ISO/IEC 8802-3.

NOTE Within this clause, any reference to bit K of an octet is reference to the bit whose weight in a one-octet unsigned is 2K, and this is sometimes referred to as "little endian" bit numbering.

7.7.2 PhIDU structure and encoding

The local MAC sublayer uses the service primitives provided by the PLS sublayer specified by ISO/IEC 8802.3 Clause 6 "Physical Signalling (PLS) service specification". All of the service primitives provided by the PLS sublayer is as follows and are considered mandatory.

a) PLS_DATA request

b) PLS_DATA indication

c) PLS_CARRIER indication

d) PLS_SIGNAL indication

e) PLS_DATA_VALID indication

NOTE In a case of 100BASE-X specification applied, the reconciliation sublayer maps the signals provided at the MII to the PLS service primitives as specified by ISO/IEC 8802.3 Clause 22 "Reconciliation Sublayer (RS) and Media Independent Interface (MII)". The PLS service primitives provided by the reconciliation sublayer behave in exactly same manner as defined in Clause 6.

7.7.3 Common MAC frame structure, encoding and elements of procedure

7.7.3.1 MAC frame structure

7.7.3.1.1 Common MAC frame format for RTE-TCnet DLPDU

The structure of MAC frame for RTE-TCnet DLPDU is encapsulated in the frame format of Ethernet V2.0 specified by ISO/IEC 8802.3 Clause 3:"Media access control frame structure". The value of the Length/Type field is designated to 0x888B, which is authorized and registered as protocol identification number by IEEE Registration Authority, to be identified as RTE-TCnet frame.

Figure 15 shows the MAC frame format used common to RTE-TCnet DLPDU except for sporadic data transmission of RTE-TCnet.

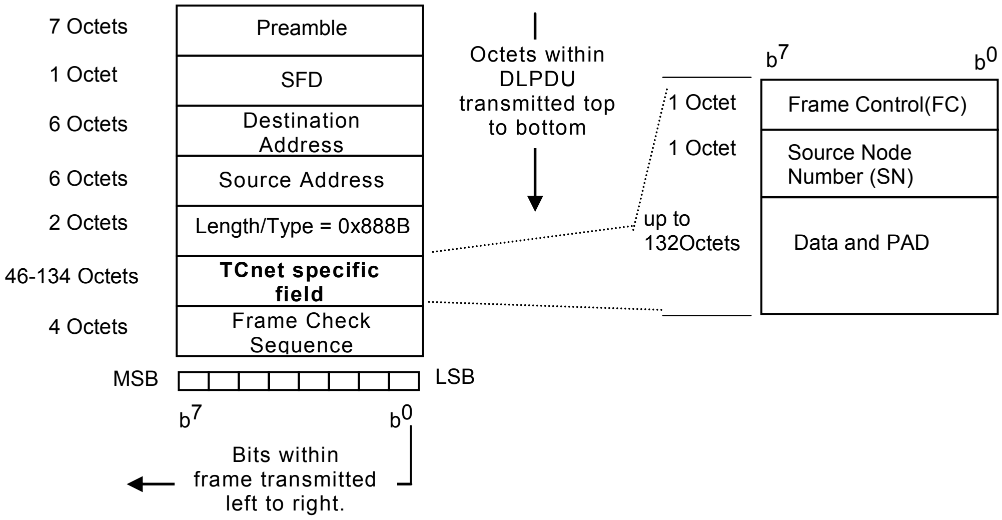


Figure 15 – Common MAC frame format for RTE-TCnet DLPDU

7.7.3.1.2 MAC frame format for RTE-TCnet Sporadic DLPDU

The MAC frame format used for Sporadic data transmission of RTE-TCnet is fully identical to the frame format of Ethernet V2.0 specified by ISO/IEC 8802.3 Clause 3, and the value of the Length/Type field is all but 0x888B. Figure 16 shows the frame format for sporadic DLPDU of RTE-TCnet.

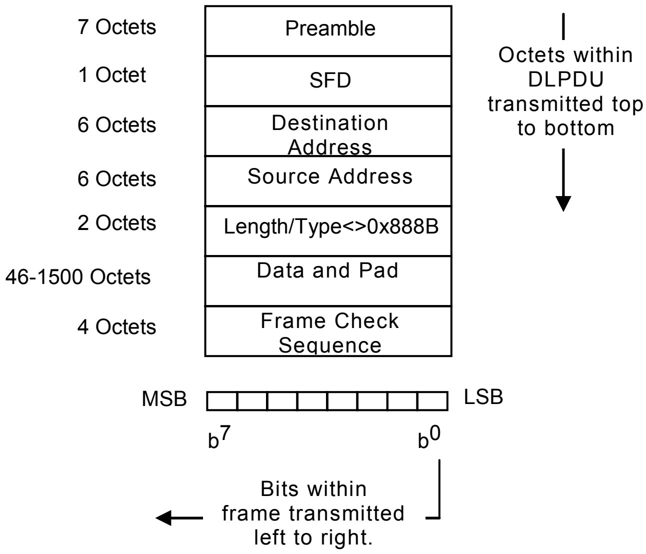


Figure 16 – Sporadic DLPDU format

7.7.4 Elements of the MAC frame

The elements of MAC frame are the preamble, the start frame delimiter, the destination address, the source address, the length/type code, and the frame check sequence specified by ISO/IEC 8802.3 Clause 3.

7.7.4.1 Preamble field

The preamble of MAC frame is identical to ISO/IEC 8802.3 Clause 3. The preamble field is a 7-octet field that is used to allow the physical signalling part circuitry to reach its steady-state synchronization with the receiving frame timing.

The preamble pattern is:

“10101010 10101010 10101010 10101010 10101010 10101010”

7.7.4.2 Start Frame Delimiter（SFD）

The Start Frame Delimiter (SFD) is identical to ISO/IEC 8802.3 Clause 3 The SFD field is the sequence of bit pattern “10101011”. It immediately follows the preamble pattern and indicates the start of a frame.

7.7.4.3 Address field

The address fields (both Destination address and Source address) are identical in structure and semantics to the address field of the basic MAC frame, described in ISO/IEC 8802.3 Clause 3 "Media access control frame structure". Each address field shall be 48 bits in length.

7.7.4.3.1 Destination Address field (DA)

The Destination Address (DA) filed is identical to ISO/IEC 8802.3 Clause 3 The Destination Address field specifies the station(s) for which the frame is intended. It may be an individual or multicast (including broadcast) address. The value of DA for RTE-TCnet DLPDU except for RTE-TCnet sporadic data transmission is always set to V(MGA).

7.7.4.3.2 Source Address field (SA)

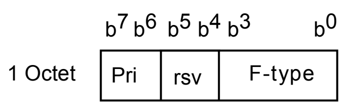
The Source Address (SA) filed is identical to ISO/IEC 8802.3 Clause 3. The Source Address field specifies the station sending the frame. The Source Address field is not interpreted by the DLE as well as CSMA/CD MAC sublayer. The value of SA is always set to V(IA).

7.7.4.4 Length/Type field

The Length/Type field is identical to ISO/IEC 8802.3 Clause 3 "Media access control frame structure". In order to be identified as RTR-TCnet frame, the value of the Length/Type field is set to 0x888B, which is authorized and registered as protocol identification number for RTE-TCnet by IEEE Registration Authority. Every frame with the value all but 0x888B is identical to the frame by ISO/IEC 8802.3 “Media access control frame” and is processed as RTE-TCnet sporadic data frame.

7.7.4.5 Frame Control field (FC)

The structure of the Frame Control field is shown in Figure 17.



NOTE

Pri: Priority
F-Type: Frame Type
rsv: Reserved （0x0）

Figure 17 – Structure of FC field

7.7.4.5.1 Frame Type（F-Type） field

F-type field is used by DLE to identify and designate the frame type of RTE-TCnet for medium access control. Table 21 shows the list of F-type and the corresponding RTE-TCnet frame.

Table 21 – F-Type and TCnet DLPDU

F-type Value	Frame type	Frame name
0x00	CLM	Claim frame
0x01	SYN	Synchronization frame
0x02	REQ	In-ring Request frame
0x04	COM	Command frame
0x05	RAS	RAS frame
0x07	DT	Cyclic Data frame
0x08	CMP	Transmission Complete frame
0x0F	DT-CMP	DT with transmission complete frame
NOTE All but above is reserved for future use.		

7.7.4.5.2 Priority field (Pri)

The priority field is used by DLE to designate and identify the service class of time-critical cyclic data frame, which is applied to both DT and DT-CMP. 3 levels of service class are provided and the top to lowest level is level 3 to 0. High-speed time-critical data transmission is assigned to level 3 of the service class, Medium-speed time-critical data transmission is to level 2 and Low-speed time-critical data transmission is to level 0. Other type of frame except for DT and DT-CMP is assigned to level 3 of the service class.

7.7.4.6 Source Node number field (SN)

The value of SN is the node identifier number and is equal to the value of V(TN) of the node which has sent out RTE-TCnet DLPDU.

7.7.4.7 Data and PAD field

The data field contains a sequence of N octets which provides full data transparency in the sense that any arbitrary sequence of octet values may appear in the data field up to a maximum number specified by ISO/IEC 8802-3. A minimum frame size is required, that is mimFrameSize by ISO/IEC 8802-3, and if a frame size is less than mimFrameSize, then the data filed is extended by appending extra bits in units of octets. The frame, from the DA field through the FCS field inclusive, is at least mimFrameSize bits.

The structure of this field for RTE-TCnet DLPDU is detailed in 7.8.

7.7.4.8 Frame Check Sequence (FCS)

The Frame Check Sequence (FCS) is identical to ISO/IEC 8802-3 Clause 3.

A cyclic redundancy check (CRC) is used by the transmit and receive algorithms to generate a CRC value for the FCS field. The FCS field contains a 4 octet (32-bit) cyclic redundancy check (CRC) value. This value is computed as a function of the contents of SA, DA, Length/Type and the RTE-TCnet specific field (that is, all fields except the Preamble, SFD and FCS).The encoding is defined by the following generating polynomial.

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$

Mathematically, the CRC value corresponding to a given frame is defined by the following procedure:

a) The first 32 bits of the frame are complemented.

b) The n bits of the frame are then considered to be the coefficients of a polynomial M(x) of degree n-1. (The first bit of the Destination Address field corresponds to the $x^{(n-1)}$ term and the last bit of the data field corresponds to the x^0 term.)

c) M(x) is multiplied by x32 and divided by G(x), producing a remainder R(x) of degree <= 31.

d) The coefficients of R(x) are considered to be a 32-bits sequence.

e) The bit sequence is complemented and the result is the CRC.

The 32 bits of the CRC value are placed in the field of frame check sequence so that the x^{31} term is the leftmost bit of the first octet, and the x^0 term is the right most bit of the last octet. (The bits of the CRC are thus transmitted in the order x^{31}, x^{30},⋯, x^1, x^0.)

7.7.5 Order of bit transmission

The order of bit transmission is identical to ISO/IEC 8802.3,Clause 3. Each octet of the DLPDU, with the exception of the FCS, is transmitted low-order bit first.

7.7.6 Invalid DLPDU

An invalid DLPDU shall be defined as one that meets at least one of the following conditions, and is almost identical to the ISO/IEC 8802-3, Clause 3:

a) The frame length is inconsistent with a length value specified in the Length/Type field. If the Length/Type field contains a type value as defined by ISO/IEC 8802-3, 3.2.6 “Length/Type field”, then the frame length is assumed consistent with this field and should not be considered an invalid DLPDU on this basis.

b) It is not an integral number of octets in length.

c) The bits of the incoming DLPDU (exclusive of the FCS field itself) do not generate a CRC value identical to the one received.

d) It is inconsistent with a F-type value of RTE-TCnet DLPDU.

The contents of invalid DLPDU shall not be passed to the DL-user or DLE. The occurrence of invalid DLPDU may be communicated to network management.

NOTE Invalid DLPDU may be ignored, discard, or used in private manner by DL-user other than RTE DL-user. The use of such DLPDUs is beyond the scope of this PAS.

7.8 DLPDU-specific structure, encoding and elements of procedure

7.8.1 General

This clause defines the structure, contents and encoding of each type and format of DLPDU except for Sporadic DLPDU of RTE-TCnet, and specifies elements of procedure for the DLPDU.

Within each subclause, the structure, contents, parameters and encoding of DLPDU are described, and the RTE-TCnet specific part of DLPDU structure, which is shown in Figure 15, is specified. The aspects relating to the sending and receiving DLS-users and their DLEs are further described. All data format and encoding is described in little endian format throughout this clause.

NOTE Within this clause, any reference to bit K of an octet is reference to the bit whose weight in a one-octet unsigned is 2K, and this is sometimes referred to as "little endian" bit numbering.

Whenever a conditional action is specified and the specific enabling condition does not occur, then the corresponding action also does not occur. The effect eventually is that events which do not meet any of the enabling conditions specified in a service procedure will have no consequential actions with respect to that specific service procedure.

7.8.2 Synchronization DLPDU (SYN)

A DLE which is as SYN node sends SYN DLPDU periodically at the specific time intervals of V(Th) for the synchronization of time-critical cyclic data transmission by all related DLEs in RTE-TCnet.

A DLE which attempts to be on line, sends REQ DLPDU for claiming to SYN node to be added in RTE-TCnet under the condition when the DLE has received a SYN DLPDU with the value of PN filed equal to the V(TN). The time frame, which REQ DLPDU is sent out, is during the MAC control period of Tmac. The MAC control period of Tmac starts immediately after the DLE has completely received a CMP DLPDU or DT-CMP DLPDU by the node with the biggest number of the value of V(TN).

A DLE which is on line and is functioning as SYN node, controls the time-critical cyclic data transmission of RTE-TCnet DLPDUs, solicits new node to join and manages node dropped out, controls and manages all of the parameters related to keep cyclic and stable data transmission. A DLE operating not as SYN node shall attempt to become new SYN node to keep cyclic and stable data transmission in succession to the current SYN node when current SYN node happens to malfunction. A DLE not operating as SYN node decides the order and the timing to send out its time-critical cyclic data transmission using the value of V(LL) in SYN DLPDU received.

7.8.2.1 Structure of SYN DLPDU

The structure of SYN DLPDU is shown in Figure 18 .

1	1	1	1	1	3	2	2	2	32	Octets
FC	SN	PN	CW	ST	Th	Tm	Ts	Tl	LL	

Where
FC: Frame control (11000001)
SN: Source node number
PN: Transmission permits node number
ST: Slot time
CW: Control word
Tm: Medium-speed periodic time
Th: High-speed periodic time
LL: Live list
Ts: Sporadic message rotation time
Tl: Low-speed periodic time

Figure 18 – Structure of SYN DLPDU

7.8.2.1.1 Parameters of SYN DLPDU

7.8.2.1.1.1 Transmission Permits Node Number (PN)

A DLE attempting to be on line is permitted to send REQ DLPDU out for claiming to enter into RTE-TCnet to SYN node when the value of PN in SYN DLPDU becomes equal to the value of V(TN). The DLE can send REQ DLPDU out to SYN node in the time frame of the MAC control period during the time slot of V(SYN) in which the DLE has received the SYN DLPDU with equal value of PN to that of the V(TN).

The range of this value is 0x1 to 0xFF. The value is incremented by each SYN DLPDU and rolled over from 0xFF to 0x1.

Table 22 shows the structure of PN parameter.

Table 22– PN -parameter: 3rd Octet

Transmission permits node number							
PN (1 to 255)							
7	6	5	4	3	2	1	0

7.8.2.1.1.2 Control word (CW)

The structure of CW is shown in Table 23.

Table 23– CW -parameters: 4th Octet

Control word							
Periodic mode	**reserved**					**Redundant medium selection**	
PM	0					**RMSEL**	
7	6	5	4	3	2	1	0

7.8.2.1.1.2.1 Periodic Mode (PM)

"PM" parameter indicates the mode of synchronization for Time-critical Cyclic data transmission.

"Free-run" mode is based on simple periodic data transmission, that is SYN node sends out SYN frame immediately after SYN node detects all node has sent out their time-critical cyclic data within the time period of V(SYN). On the other hand, "Constant period" mode is based on time-synchronized data transmission with a signal from a clock source of constant time-period.

Table 24 shows the mode of PM parameter.

Table 24 – PM parameter

PM mode	Value	Description
Free-run	0	Cyclic data transmission based on free run
Constant period	1	Cyclic data transmission based on time-synchronization

7.8.2.1.1.2.2 Redundant medium selection (RMSEL)

This parameter indicates and designates the mode of redundant medium selection to all nodes. The value of RMSEL is used and set to V (RCS) by the DLE for switching control to receive packets out of one of two receive-channel A and B corresponding to each of the redundant medium A and B. The possible value is "Automatic", "Force A" and "Force B".

"Automatic" designates that the switching control for receiving packets is automatically to switch to the proper receive-channel. "Force A" or "Force B" designates that the switching control is forced to switch to Receive-channel A or B respectively. The initial value is "Automatic".

The mode of RMSEL parameter is summarized in Table 25.

Table 25 – RMSEL parameter

RMSEL mode	value	Description
Automatic	00	Automatically switch to the proper receive-channel
	01	Reserved for future use
Force A	10	Force to switch receive-channel A
Force B	11	Force to switch receive-channel B

7.8.2.1.1.3 Slot time (ST)

This parameter is the fundamentally observational time unit, using in the DLE for observing to initiate action such for re-initialisation in sending out CLM packet, for sending out the CMP packet specifically by the current SYN node in substitute for the node missed to send out the CMP packet.

V(ST) holds the value of ST. The rage of this value is 1 to 255, of which time is in 512 bits physical symbol time specified by and identical to ISO/IEC 8802-3. The default value is 20, which is equivalent to Approx.100 microsecond, i.e. 20 X 512 x 1/100 microsecond.

Table 26 shows the structure of ST parameter.

Table 26 – ST-parameter: 5th Octet

V(ST): Slot time							
7	6	5	4	3	2	1	0

7.8.2.1.1.4 High-speed transmission period (Th)

This parameter designates the cyclic time period of High-speed time-critical data transmission. The value of Th is used by the DLE, and ultimately set to V(Th), V(SYN) equally of each node on-line. The range of this value is 1250 to 2500000, of which unit is in 80 nanosecond.

Table 27 shows the structure of Th parameter.

Table 27 – Th-parameter: 6th, 7th and 8th Octets

High-speed transmission period																							
Th (2^7 – 2^0)								Th (2^8 – 2^{15})								Th (2^{16} – 2^{31})							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

7.8.2.1.2 Medium-speed transmission period (Tm)

This parameter designates the cyclic time period of Medium-speed time-critical data transmission. The value of Tm is used by the DLE, and is set to V(Tm) equally of each node on-line. The range of this value is 10 to 1000, of which unit is in 1millisecond.

Table 28 shows the structure of Tm parameter.

Table 28 – Tm-parameter: 9th and 10th Octets

Medium-speed transmission period															
Tm (2^7 – 2^0)								Tm (2^8 – 2^{15})							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

7.8.2.1.2.1 Sporadic message transmission target-token-rotation-time period (Ts)

This parameter designates the value of target-token-rotation-time for sporadic message data transmission. This value is used by the DLE, and is set to V(TTRT1) equally of each node on-line. The range of this value is 10 to 10000, of which unit is in 1 millisecond.

Table 29 shows the structure of Ts parameter.

Table 29 – Ts-parameter: 11th and 12th Octets

Sporadic message transmission target-rotation-time period															
Ts (2^7 – 2^0)								Ts (2^8 – 2^{15})							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

7.8.2.1.2.2 Low-speed transmission period (Tl)

This parameter designates the cyclic time period of Low-speed time-critical data transmission. The value of Tl is used by the DLE, and is set to V(Tl) equally of each node on-line. The range of this value is 10 to 10000, of which unit is in 1 millisecond.

Table 30 shows the structure of Tl parameter.

Table 30 – Tl-parameter: 13th and 14th Octets

Low-speed transmission period															
Tl (2^7 – 2^0)								Tl (2^8 – 2^{15})							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

7.8.2.1.2.3 Live list (LL)

This parameter indicates the current operational status, whether a corresponding node is on line and is running normal in RTE-TCnet. Each bit of value “1” indicates the corresponding node on line and normal, and a node of value “0” is off line.

V(LL) is used by the DLE and is generated from the information conveyed by SYN frame. Live-list is a collection of 8 words of 32 bits length, each of which bit is corresponding to each node in the RTE-TCnet and indicates the current operational status. Each bit in the LL word corresponds to the node number V(TN) in a sequential order from 0 to 255 in little endian format.

Table 31 shows the structure of LL parameter.

Table 31 – LL parameters: 15th to 46th Octets

	Live-list																
	LL (2^7 – 2^0)								LL (2^8 – 2^{15})								
15th	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	16th
17th	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24	18th
19th	39	38	37	36	35	34	33	32	47	46	45	44	43	42	41	40	20th
21th	55	54	53	52	51	50	49	48	63	62	61	60	59	58	57	56	22th
23th	71	70	69	68	67	66	65	64	79	78	77	76	75	74	73	72	24th
:																	:
:																	:
41th	215	214	213	212	211	210	209	208	223	222	221	220	219	218	217	216	42th
43th	231	230	229	228	227	226	225	224	239	238	237	236	235	234	233	232	44th
45th	247	246	245	244	243	242	241	240	255	254	253	252	251	250	249	248	46th

7.8.2.1.3 User data

No user data is conveyed by SYN DLPDU.

7.8.2.2 Sending SYN DLPDU

A DLE of current SYN node sends out SYN DLPDU with a set of parameters, as stated above, which are managed, maintained and set to SYN DLPDU by the DLE of current SYN node.

7.8.2.3 Receiving the SYN DLPDU

Each DLE of the node, which is on line or is to be on line, except SYN node takes the following actions on reception of SYN DLPDU.

1) The value of RMSEL is used to determine the switching control for receiving packets out of one of two receive-channel A and B corresponding to each of the redundant medium A and B, and is reflected in V(RCS).
2) Each value of ST, Th, Ts and Tl received in SYN DLPDU and the corresponding value of V(ST),V(Th),V(Tm) and V(Tl) in a node is compared respectively. If the value is different, then each value of ST, Th, Ts and Tl is to be a new value of each variable V(ST), V(Th), V(Tm) and V(TL) respectively.
3) The value of LL is to be a new value of V(LL).
4) The DLE, which has sent REQ DLPDU out, shall confirm that the corresponding bit to the V(TN) in LL becomes “true”. If the value of the corresponding bit of LL is “true”, then the node of the V(TN) is on line. The DLE of the V(TN) shall start to send out its Time-critical Cyclic data every time immediately after the DLE has received CMP or DT-CMP DLPDU from a node that is on line and of which node identifier number can be obtained from the LL and is lower in sequential order in the LL.

7.8.3 Transmission Complete DLPDU (CMP)

CMP DLPDU is sent out to indicate all of the time-critical cyclic data transmission completed at the end of data transmission. Two types of DLPDU for indicating the data transmission completed is specified, one is CMP DLPDU and the other is DT DLPDU.

7.8.3.1 Structure of CMP DLPDU

Figure 19 shows the structure of CMP DLPDU.

1	1	1	1	Octet
FC	SN	rsv	SYN	

Where
FC: Frame Control (11001000) SN: Source Node number
rsv: Reserved (0x0)
SYN: SYN node number

Figure 19 – Structure of CMP DLPDU

7.8.3.1.1 Parameters of the CMP DLPDU

7.8.3.1.1.1 SYN node number (SYN)

This parameter indicates the number of the V(TN) of current SYN node.

7.8.3.1.2 User data of CMP DLPDU

No user data is conveyed by CMP DLPDU.

7.8.3.2 Sending the CMP DLPDU

CMP DLPDU is sent out by the DLE at the timing and condition specified as follows.

1) The DLE which is in the assigned time frame to send out its data and is just sending out its data in the V(TSYN) time period, shall send out CMP DLPDU at the time when the DLE has completed to send out all of the data both of Time-critical Cyclic data and Sporadic message data. CMP DLPDU is sent out by the last frame out of the node to notify its data transmission completed to all RT-TCnet nodes.

   NOTE When a DLE sends out time-critical cyclic data as the last DLPDU, the DLE shall send out DT-CMP DLPDU in substitution of CMP DLPDU.

2) A DLE of SYN node shall wait for CMP or DT-CMP DLPDU from the DLE, which is sending its data out, in the duration of V(SCMP). When the DLE of SYN node had missed to receive CMP or DT-CMP DLPDU, the DLE of SYN node shall initiate sending out CMP DLPDU in substitution for the DLE which has not send out its CMP or DT-CMP DLPDU.

7.8.3.3 Receiving CMP DLPDU

Each DLE of the node, which is on line, shall activate the followings on reception of CMP or DT-CMP DLPDU.

1) The value of V(Tsl) is updated to the number of V(TN) of which node is to send out its date, after searched next node out of V(LL), translated into and get actual node identifier number of V(TN). In searching a next node, the DLE checks and extracts first bit of V(LL) being “true” in the greater order than the corresponding bit position of the node which has completed to send out. Found the first bit of being “true” and according to the bit position found in the V(LL), the DLE shall translate the bit position into the actual node identifier number of V(TN).

2) At the condition that the value of the node identifier number, which is translated from the bit position in the V(LL), is equal to the number of V(TN) of this node, the DLE shall begin to send out it data of every high-speed time-critical cyclic data, Medium-speed time-critical cyclic data, Sporadic message data and Low-speed time-critical cyclic data in this order.

The Medium-speed Time-critical Cyclic data, Sporadic message data and low-speed time-critical cyclic data is able to send out from the node if the condition is met respectively, and each data transmission shall terminate at one of the following condition met.

a) All the data with each priority level has completed to send out.

b) The corresponding observational timer is expired.

c) T(MTHT) for High-speed time-critical cyclic data transmission has been expired.

7.8.4 In-Ring Request DLPDU (REQ)

REQ DLPDU is sent out by the DLE to SYN node to claim to be added in RTE-TCnet.

7.8.4.1 Structure of REQ DLPDU

Figure 20 shows the structure of REQ DLPDU.

1	1	1	Octet
FC	SN	rsv	

Where
FC: Frame Control (11000010) SN: Source Node number
rsv: Reserved (0x0)

Figure 20 – Structure of the REQ DLPDU

7.8.4.1.1 Parameter of REQ DLPDU

No user parameter is conveyed by REQ DLPDU.

7.8.4.1.2 User data of REQ DLPDU

No user data is conveyed by REQ DLPDU.

7.8.4.2 Sending the REQ DLPDU

REQ DLPDU is sent out by the DLE of a node which attempts to be a member in RTE-TCnet, claiming to be added and to be on-line, under the condition when the DLE has received a SYN DLPDU with the value of PN field equal to the V(TN) and further in the time frame of the MAC control period of Tmac. The start of MAC control period of Tmac is immediately after the DLE has completely received a CMP DLPDU or a DT-CMP DLPDU out of a node with the biggest number of the value of V(TN) in RTE-TCnet.

7.8.4.3 Receiving the REQ DLPDU

REQ DLPDU is received by the DLE of current SYN node in operation. Received REQ DLPDU, the DLE extracts the number of V(TN) of sending DLE out of the value of SN field in the REQ DLPDU, and sets to "1" (being on-line or live) the corresponding bit of the V(TN) in V(LL). Immediately after the MAC control period of Tmac over during which the DLE received a REQ DLPDU, the DLE sends out SYN DLPDU with the value of extracted number of V(TN) in the LL field.

7.8.5 Claim DLPDU (CLM)

CLM DLPDU is sent out by the DLE claiming for re-synchronization of RTE-TCnet under the condition that current SYN node happens into malfunction and T(SL) timer to monitor SYN node inactive is expired. CLM DLPDU can be sent out from the node which is designated and permitted to be SYN node. The value of T(SL) of each node is different and depends on the value of the V(ST) and the V(TN).

7.8.5.1 Structure of CLM DLPDU

Figure 21 shows the structure of CLM DLPDU.

1	1	1	1	1	Octet
FC	SN	rsv	RC	ST	

Where
FC: Frame Control(11000000) SN: Source Node number
rsv: Reserved (0x0)
RC: Residual Counts of CLM DLPDU able to send out

Figure 21 – Structure of CLM DLPDU

7.8.5.1.1 Parameter of the CLM DLPDU

7.8.5.1.1.1 Residual Counts of CLM DLPDU parameter (RC)

This parameter indicates the residual counts for the DLE to send out CLM DLPDU. The value is decremented in the range of V(ST) to 0. The default value is 20. Reached to count “0”, the DLE becomes SYN DLE.

Table 32 shows the structure of CLM parameter.

Table 32 – CLM-parameter: 4th Octet

Residual Counts of CLM DLPDU							
V(RC) (2^7 – 2^0)							
7	6	5	4	3	2	1	0

7.8.5.1.1.2 Slot time (ST) parameter

The value of this parameter is set to that defined in 7.8.2.1.1.3.

7.8.5.1.2 User data of the CLM DLPDU

No user data is conveyed by CLM DLPDU.

7.8.5.2 Sending and Receiving CLM DLPDU

CLM DLPDU is sent out by the DLE, which is provided with SYN node function, claiming for re-synchronization of RTE-TCnet.

After the DLE has been powered and initialized, once T(SL) timer happens to be expired, T(ST) timer is successively triggered. When T(ST) timer is expired, CLM DLPDU is sent out by the DLE.

The DLE shall send out CLM DLPDU in predefined number of times, which depends on the value of V(ST). The number of times is calculated by the following equation, in which “N” is the number of times.

$$N=(2 \text{ x } V(MD))+(1/2 \text{ x } V(MRT)$$

During a CLM DLPDU being sent out, when the collision has happened, the DLE immediately stops to send out CLM DLPDU and starts to monitor and check the condition over the common medium.

When the DLE has successfully received a CLM DLPDU from other DLE, the DLE shall extract the value of SN within the CLM DLPDU and compare the value of SN with its V(TN). When the value of SN is lower than the V(TN), the DLE shall not persist in sending out another CLM DLPDU. On the contrary when the V(TN) is lower, the DLE shall again attempt to send out CLM DLPDU after both of T(SL) and consecutive T(ST) expired.

When the DLE has successfully sent out CLM DLPDU in total counts of V(ST), the DLE becomes SYNM DLE. On the other hand received SYN DLPDU from other DLE, the DLE shall cease to send out CLM DLPDU for re-synchronization.

The DLE that is not provided with SYN node function or is not designated to be a SYN node, shall not send out CLM DLPDU, furthermore shall not respond to CLM DLPDU received from other node.

7.8.6 Command (COM) DLPDU

COM DLPDU is sent out by the DLE to ask the DLE of SYN node to equalize and change the value of RTE-TCnet parameters of all member nodes, and the change shall be reflected in the SYN DLPDU to be sent in the next V(Th) time period . The RTE-TCnet parameters to be equalized are as follows.

1) V(ST): Slot time(ST)

2) V(Th), V(Tm), V(Ts), V(Tl): The cyclic time periods, the target-token-rotation time period

3) V(RCS): Redundant medium selection for receiving data packets

7.8.6.1 Structure of the COM DLPDU

Figure 22 shows the structure of COM DLPDU.

1	1	1	1	1	3	2	2	2	32	Octet
FC	SN	rsv	CW	ST	Th	Tm	Ts	Tl	rsv	

where
FC: Frame control(11000100)
SN: Source node number
rsv: Reserved (00000000)
CW: Control word
ST: Slot time
Th: High-speed periodic time
Tm: Medium-speed periodic time
Ts: Sporadic message rotation time
Tl: Low-speed periodic time

Figure 22 – Structure of COM DLPDU

7.8.6.1.1 Parameters of COM DLPDU

The parameters of COM DLPDU are equal to that of SYN DLPDU specified in 7.8.2.1.1Parameters of SYN DLPDU.

7.8.6.1.2 User data of the COM DLPDU

No user data is conveyed by COM DLPDU.

7.8.6.2 Sending and receiving COM DLPDU

The DLE is activated to send COM DLPDU by receiving DLM_SET_Value primitive, especially requiring change of each value of V(ST), V(Th), V(Tm), V(Ts), V(Tl) and V(RCS).

When the DLE of SYN node has successfully received COM DLPDU from the DLE of other node, the DLE reflects the requirement by COM DLPDU to SYN DLPDU to be sent in the next V(Th) time period.

7.8.7 Cyclic data and Cyclic data with transmission complete DLPDU (DT) and (DT-CMP)

Either DT DLPDU or DT-CMP DLPDU is used to convey the Time-critical Cyclic data to all RTE-TCnet member nodes. The difference of DT DLPDU and DT-CMP DLPDU is indication of transmit completion or not.

7.8.7.1 Structure of the DT DLPDU

Figure 23 shows the structure of DT DLPDU.

1	1	2	Up to 130	Octet
FC	SN	DLCEP-address	User data	

where
FC: Frame Control (DT: pp000111, DT-CMP: pp001111)
SN: Source node number

Figure 23 – Structure of DT DLPDU

7.8.7.1.1 Parameters of DT DLPDU

7.8.7.1.1.1 DLCEP-address parameter

This parameter indicates DLCEP for predefined multi-point publisher DL-connection.

Table 33 shows the structure of DT parameter.

Table 33 – DT parameter: 3rd and 4th Octet

DLCEP-address															
DLCEP-address (2^7 – 2^0)								DLCEP-address (2^8 – 2^{15})							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

7.8.7.1.2 User data

DT DLPDU can send user data up to 132 octets.

7.8.7.2 Sending DT or DT-CMP DLPDU

The CTRC(Cyclic-transmission TX/RC control) manages to send out DT DLPDU or DT-CMP DLPDU on receipt of DL-PUT primitive from DL-user.

The type of High-speed Cyclic data transmission or Medium-speed Cyclic data transmission or Low-speed Cyclic data transmission shall be designated to the pp sub-field of FC of DT DLPDU and DT-CMP DLPDU.

7.8.7.3 Receiving DT or DT-CMP DLPDU

The DLE that has received DT DLPDU or DT-CMP DLPDU shall update the corresponding Receive_buffer associated with DLCEP address field in DT DLPDU or DT-CMP DLPDU, and notify the DLS-user using DL-Buffer-Received indication primitive that the data in the specified receive-buffer is updated and is available to read out.

7.8.8 RAS DLPDU (RAS)

RAS DLPDU is used for transfer the RAS(Reliability, Availability, Serviceability) related information to all member node.

7.8.8.1 Structure of RAS DLPDU

Figure 24 shows the structure of RAS DLPDU.

1	1	2	Up to 34	Octet
FC	SN	DLCEP-address	User data	

where
FC: Frame control(11000101) SN: Source Node number

Figure 24 – Structure of RAS DLPDU

7.8.8.2 Parameters of RAS DLPDU

7.8.8.2.1 DCEP-address parameter

This parameter indicates DLCEP for predefined multi-point publisher DL-connection.

Table 34 shows the structure of RAS parameter.

Table 34 — RAS parameter: 3rd and 4th byte

DLCEP-address															
DLCEP-address (2^7 – 2^0)								DLCEP-address (2^8 – 2^{15})							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

7.8.8.2.2 User data

RAS DLPDU conveys the user data, which is specifically associated with RAS related information of local DLE, up to 34 octets one time. The DLE of each local node may handle RAS related information over 34 octets, and then the total number of RAS(related information of each node is broken down into several fragments and each fragment is conveyed on RAS DLPDU.

7.8.8.2.3 Sending and receiving RAS DLPDU

The DLE of local node can broadcast the RAS related information to all RTE-TCnet member node using RAS DLPDU. RAS DLPDU is sent out during TMAC time duration by the DLE of each local node each by each in order of V(TN), therefore the transmission of RAS DLPDU or broadcasting of RAS related information of local node is carried out on background basis.

Received RAS DLPDU, the DLE shall update the corresponding Receive_buffer associated with DLCEP address field in RAS DLPDU ,and notify the DLS-user using DL-Buffer-Receive indication primitive that the data in specific receive-buffer is updated and is available to read out.

7.9 DLE elements of procedure

7.9.1 Overall structure

The DLL is composed of control elements of Cyclic transmission TX/RX Control(CTRC), Sporadic TX/RX Control(STRC), Access Control Machine(ACM), Redundancy Medium Control(RMC), TX/RX Framer, Octet Serializer, Octet Deserializer and DLL Management Interface.

The ACM as the primary control element provides the function for deterministic media access control cooperating with the CTRC, the STRC and the RMC for reliable and efficient support both of higher-level connection-mode and connection-less mode data transfer services.

The DLL management interface provides DLL management functions.

Figure 25 depicts the overall structure of DLL.

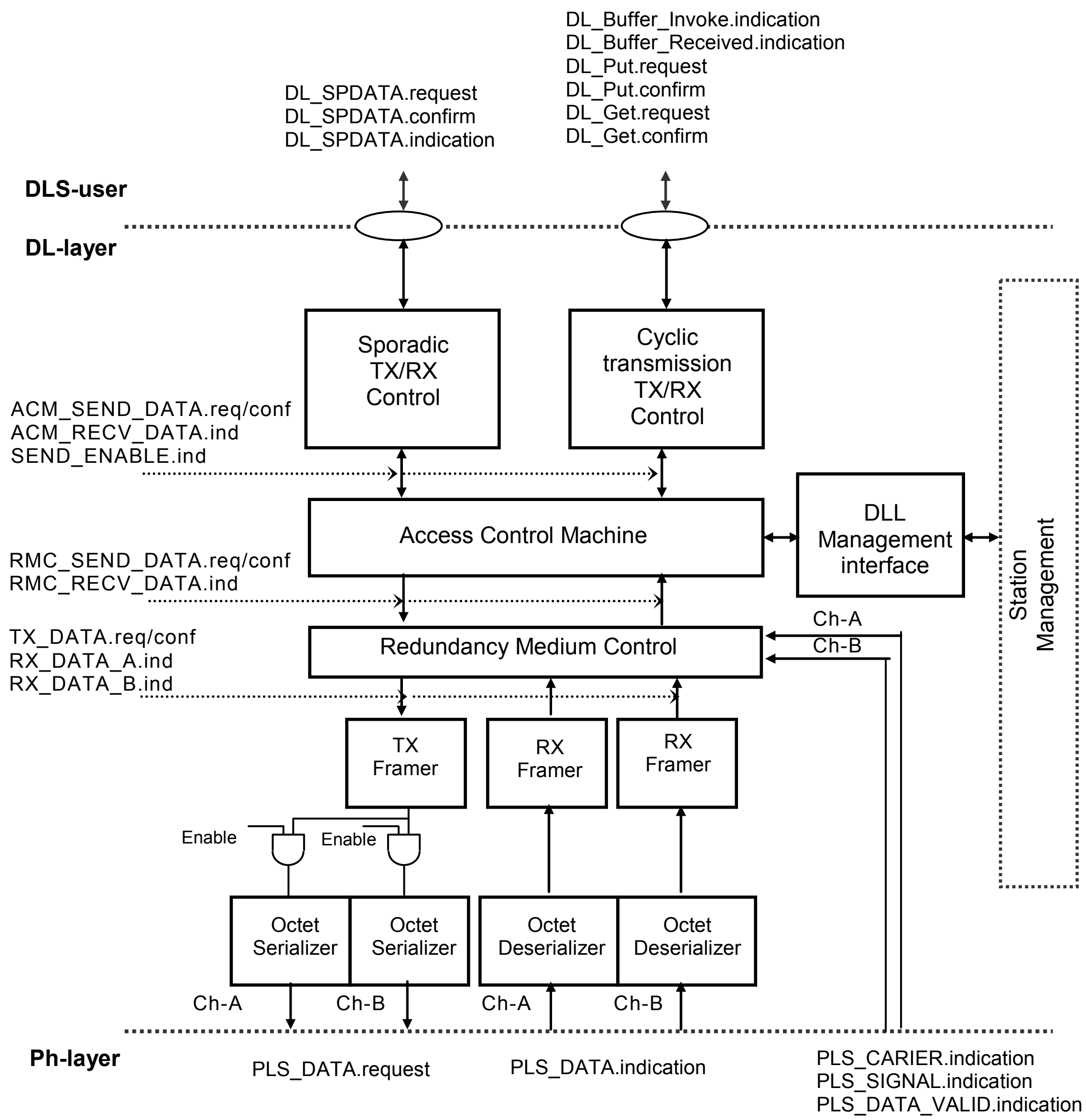


Figure 25 – Overall structure of DLL

7.9.2 Initialization

Upon power-up or after received DLM-RESET request primitive, the DLE shall go into the OFF-LINE state. When in OFF-LINE, the DLE shall not transmit and shall ignore any DLPDU received.

When all of the variables for normal DLE operation are set up by DLM-Set-Value request primitive, the state is changed from OFF-LINE to STATION MODE CONTROL in which the DLE shall start normal DL operation.

Figure 26 depicts the DLE state transition.

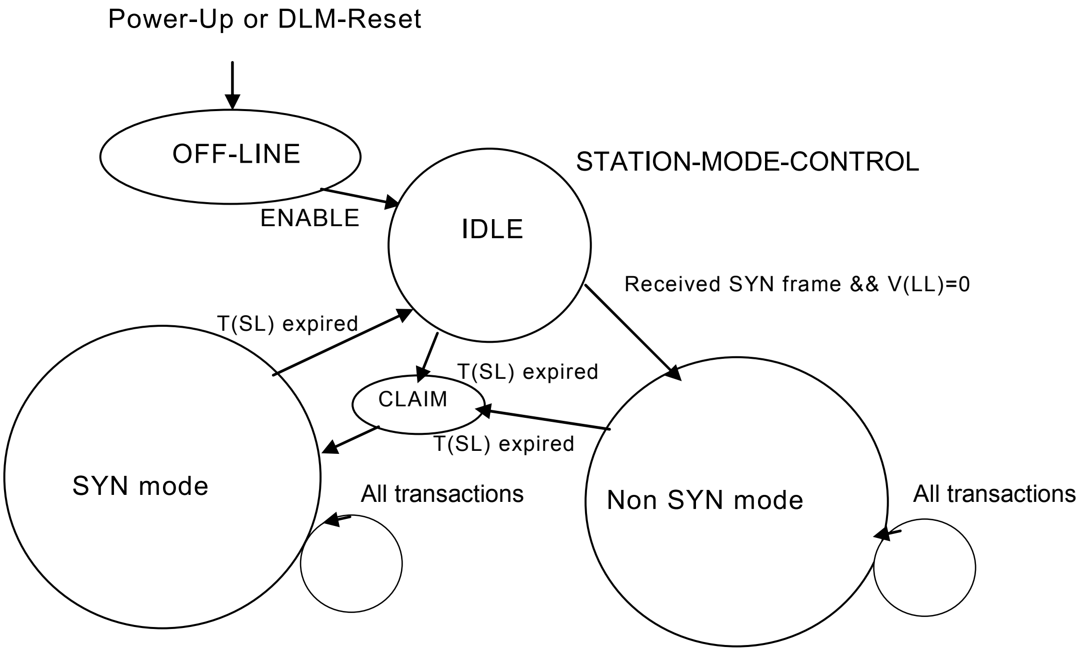


Figure 26 – DLE state transition

7.9.3 Cyclic transmission TX/RX Control (CTRC)

7.9.3.1 Overview

The Cyclic transmission TX/RX Control (CTRC) is responsible for buffering and dispatching DLSDU received for Time-critical Cyclic data transfer between the DLS-user and the ACM.

DL_DATA_REQ indication primitive by the CTRC informs the DLS-user to initiate the data transfer from each DLS-user buffer to the send buffer of the DLE using DL_PUT request primitive. The DLS-user buffer is addressed with the DLCEP-identifier parameter in the DL_DATA_REQ indication primitive from the CTRC.

Each DLS-user buffer contains data for Time-critical Cyclic data transmission. After the DLS-user data received, the CTRC issues ACM_SEND_DATA request primitive to initiate ACM sending out the DLS-user data as DT or DT-CMP DLPDU.

DL_DATA_REQ indication primitive is issued by the CTRC in response to SEND_ENABLE indication primitive from the ACM. The SPEED Class parameter in SEND_ENABLE indication primitive allows the CTRC to activate transfer of corresponding class of data–buffer by the DLS-user. The classes designated by the SPEED Class parameter correspond to the classes of High-speed Cyclic data, Medium-speed Cyclic data and Low-speed Cyclic data transfer.

The data transfer of each Speed Class designated is dependant on the level of transfer priority.

1) As for Priority class 3, which corresponds to High-speed Cyclic data transfer, when received SEND_ENABLE indication primitive, the CTRC shall handle all of the data of every DLS-user buffer corresponding to High-speed Cyclic data transfer or High-Speed

class is to be delivered to ACM and to be sent out on the transmission medium all at once on each occasion.

2) As for Priority class 2 and 0, which corresponds to Medium-Speed Cyclic data transfer or Medium-Speed class and Low-Speed Cyclic data transfer or Low-Speed class respectively, the CTRC shall handle on each occasion the data of every DLS-user buffer corresponding to Medium-Speed and Low-Speed class is sent out or not depending on the condition. The token-holding time of each class 2 and 0 governs the condition. After sent out all the High-Speed Cyclic data, the Medium-Speed Cyclic data is sent out. If the token-holding time ends during sending the Medium-Speed Cyclic data, the data transfer is interrupted and the CMP DLPDU or DT-CMP DLPDU is sent out at the occasion. On the next occasion happens, the remainder of the previous Medium-Speed Cyclic data is to be sent out depending on same condition. As for the Low-Speed Cyclic data, same control for the Medium-Speed Cyclic data transfer is to be done. The occasion happens every V(TSYN) time period.

When received ACM_RECV_DATA indication primitive from ACM, the CTRC stores the DLPDU by ACM into each receive-buffer, which is addressed with the DLCEP-identifier parameter in ACM_RECV_DATA indication primitive. Furthermore the CTRC issues DL_Buffer_Received indication primitive to inform the DLS-user the receive-buffer of the DLE is updated and is available to read out. The DLS-user can read out the data in the receive-buffer using DL_Get request primitive in response to DL_Buffer_Received indication primitive from CTRC.

7.9.3.2 Primitive definitions

7.9.3.2.1 Primitives exchanged between DLS-user and CTRC

The primitives exchanged between DLS-user and CTRC is shown in Table 35 and the primitives exchanged between CTRC and ACM is summarized in Table 36.

Table 35 – Primitives exchanged between DLS-user and CTRC

Primitive names	Source	Associated parameters
DL_DATA_REQ.indication	CTRC	DLCEP-identifier
DL_Put.request	DL-user	DLCEP-identifier, DLSDU-length, DLSDU
DL_Put.confirm	CTRC	DLCEP-identifier, Status
DL_Buffer_Received.indication	CTRC	DLCEP-identifier
DL_Get.request	DL-user	DLCEP-identifier
DL_Get.confirm	CTRC	DLCEP-identifier, DLSDU-length, DLSDU, Status

Table 36 – Primitives exchanged between CTRC and ACM

Primitive names	Source	Associated parameters
SEND_ENABLE.ind	ACM	Speed-Class
ACM_SEND_DATA.req	CTRC	DLPDU
ACM_SEND_DATA.conf	ACM	Status
ACM_RECV_DATA.ind	ACM	DLPDU

7.9.3.2.2 Parameters used with primitives exchanged between DLS-user and CTRC

The parameters used for interaction between DLS-user and CTRC are summarized in Table 37.

Table 37 – Parameters used with primitives exchanged between DLS-user and CTRC

Parameter names	Description
DLCEP-identifier	Identifier to designate Send-Buffer or Receive-Buffer
DLSDU	The contents of Send-Buffer or Receive-Buffer, which is Time-critical Cyclic data processed by CTRC
DLSDU-length	The length of DLSDU
Speed-Class	Speed-Class to designate the class of Time-critical Cyclic data transfer, that is for High-speed Cyclic data, Medium-speed Cyclic data and Low-speed Cyclic data.
Status	Status report whether the requested service is successfully provided or failed for the reason specified.

7.9.3.3 CTRC state table

The state transition diagram of CTRC is depicted in Figure 27, and the CTRC state table is shown in Table 38.

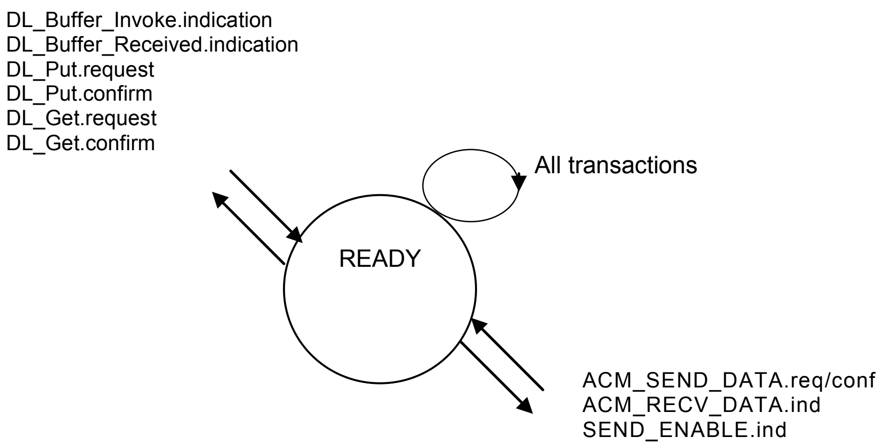


Figure 27 – State transition diagram of CTRC

Table 38 – CTRC state table

No.	Current state	Event /condition ⇒ actions	Next state
1	READY	DL_Put.request {DLCEP-identifier, DLSDU-length, DLSDU} / CHECK_PAR_PUT (DLCEP-identifier, DLSDU) = "True" ⇒ PUT_BUFFER (DLCEP-identifier, DLSDU) DL_Put.confirm { DLCEP-identifier, DLSDU-length, Status:= "success" }	READY
2	READY	DL_Put.request {DLCEP-identifier, DLSDU-length, DLSDU} / CHECK_PAR_PUT (DLCEP-identifier, DLSDU) = "False" ⇒ DL_Put.confirm { DLCEP-identifier, DLSDU-length, Status:= "failure" }	READY
3	READY	DL_Get.request {DLCEP-identifier } / CHECK_PAR_GET (DLCEP-identifier) = "True" ⇒ DL_Get.confirm { DLCEP-identifier, DLSDU-length, DLSDU:= GET_BUFFER(DLCEP-identifier), Status:= "success" }	READY
4	READY	DL_Get.request {DLCEP-identifier} / CHECK_PAR_GET (DLCEP-identifier) = "False" ⇒ DL_Get.confirm { DLCEP-identifier, DLSDU-length, DLSDU:= null, Status:= "failure" }	READY

5	READY	SEND_ENABLE.ind {Speed-Class}
		/ Speed-Class <> "SPORADIC" &&
		CHECK_NEXT_SEND(Speed-Class) = "True"
		⇒
		DL_DATA_REQ indication {
		DLCEP-identifier:= GET_NEXT_ID(Speed-Class)
		}
		DLSDU:= GET_BUFFER (DLCEP-identifier)
		DLPDU: = BUILD_DT (Speed-Class, DLCEP-identifier, DLSDU)
		ACM_DATA.req { DLPDU
		}
		ACM_DATA.conf { } -- immediate
		⇒
		NEXT(Speed-Class)
			READY

6	READY	SEND_ENABLE.ind {Speed-Class} / Speed-Class <> "SPORADIC" && CHECK_NEXT_SEND(Speed-Class) = "False" ⇒ ACM_DATA.req { null -- no data } ACM_DATA.conf { } -- immediate ⇒ NEXT(Speed-Class)	READY

7	READY	ACM_DATA.ind {DLPDU}
		/ Class <> "SCOPRADIC" &&
		CHECK_PAR_DT (DLPDU)
		⇒
		DL_Buffer_Received.indication {
		DLPDU.DLCEP-address
		}
			READY

7.9.3.4 Functions of CTRC

All functions of the CTRC are summarized in Table 39.

Table 39 – CTRC Functions table

Function Name	Input	Output	Operation
CHECK_PAR_PUT	DLCEP-identifier, DLSDU	True/False	Check that all parameters of DLCEP-identifier and DLSDU of DL_Put.request primitive are valid. If valid, “True” is returned and otherwise “False” is returned.
PUT_BUFFER	DLCEP-identifier, DLSDU	(none)	Store DLSDU into the Send-Buffer associated with DLCEP-identifier.
CHECK_PAR_GET	DLCEP-identifier	True/False	Check that DLCEP-identifier of DL_Get.request primitive is valid. If valid, “True” is returned.
GET_BUFFER	DLCEP-identifier	DLSDU	Get DLSDU in the Receive-Buffer associated with DLCEP-identifier.
CHECK_NEXT_SEND	Speed-Class	True/False	Check that the DLSDU specified by Speed-Class exists. If exists, “True” is returned and otherwise “False” is returned.
GET_NEXT_ID	Speed-Class	DLCEP-identifier	Get next DLCEP-identifier of the DLS User-Buffer specified by Speed-Class.
BUILD_DT	Speed-Class, DLCEP-identifier, DLSDU	DLPDU	Build into DLPDU of Time-critical Cyclic data out of the data in the Send-Buffer. DLPDU is assembled as follows. DLPDU.DA:= V(MGA) DLPDU.SA:= V(IA) DLPDU.Len/Type:= 0x888b DLPDU.FC:= Speed-Class + "DT" DLPDU.SN:= V(TN) DLPDU.DLCEP.address:= DLCEP-identifier DLPDU.DLSDU:= DLSDU
CHECK_PAR_DT	DLPDU	True/False	Check that the specified DLPDU is valid. If valid, “True” is returned.
NEXT	Speed-Class		Check that additional DLSDU specified by Speed-Class is remained. If remained, the control is returned to the top of current state.

7.9.4 Sporadic TX/RX Control （STRC）

7.9.4.1 Overview

The Sporadic TX/RX control (STRC) is responsible for buffering and dispatching in time DLSDU received for the Sporadic message data between DLS-user and the ACM.

7.9.4.2 Primitive definitions

7.9.4.2.1 Primitives exchanged between DLS-user and STRC

Table 40 shows all primitives exchanged between DLS-user and STRC and Table 41 shows all primitives exchanged between STRC and ACM.

Table 40 – Primitives exchanged between DLS-user and STRC

Primitive names	Source	Associated parameters
DL_SPDATA.request	DLS-user	DA, MSDU
DL_SPDATA.confirm	DLS-user	DA, Status
DL_SPDATA.indication	STRC	DA, SA, MSDU, Rec-Status

Table 41 – Primitives exchanged between STRC and ACM

Primitive names	Source	Associated parameters
SEND_ENABLE.ind	ACM	Speed-Class
ACM_SEND_DATA.req	STRC	DLPDU
ACM_SEND_DATA.conf	ACM	Status
ACM_SEND_DATA.ind	ACM	DLPDU

7.9.4.2.2 Parameters used with primitives exchanged between DLS-user and STRC

All parameters used with primitives exchanged between DLS-user and STRC is shown in Table 42.

Table 42 – Parameters used with primitives exchanged between DLS-user and STRC

Parameter names	Description
DA	Destination Address
SA	Source Address
MSDU	MAC service data unit
Status	Indicate whether the requested service of DL_SPDATA.request is successfully provided or failed for the reason specified.
Rec-Status	Indicates whether the DLPDU had received without error or not.
DLPDU	DLPDU for Sporadic message transfer
Speed-Class	Speed-Class requested by DLS-user. In Sporadic message transfer service, Speed-Class is specified to “Sporadic” or Class 1.

7.9.4.3 STRC state table

The state transition diagram of STRC is depicted in Figure 28, and the STC state table is shown in Table 43.

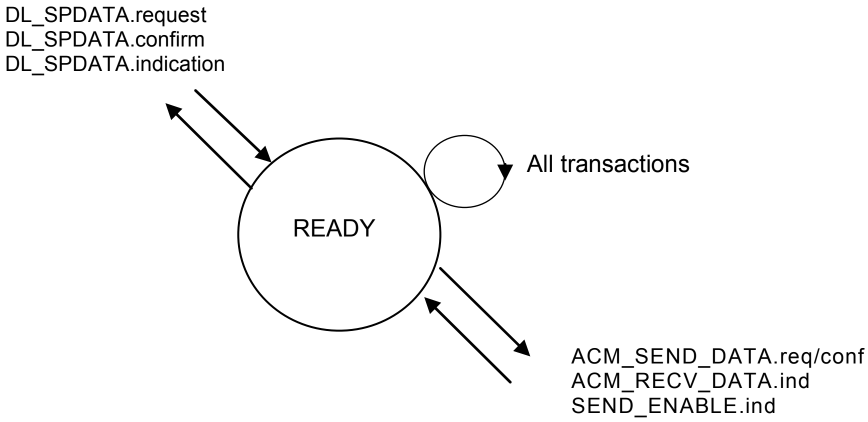


Figure 28 – State transition diagram of STRC

Table 43 – STRC state table

No.	Current state	Event /condition ⇒ actions	Next state
1	READY	DL_SPDATA.request {DA, MSDU} / CHECK_PAR_SPDATA (DA, MSDU) = "True" && CHECK_SPDATAQ ( ) <> "Full" ⇒ QUEUE_SPDATA (DA, MSDU) DL_SPDATA.confirm { DA, MSDU, Status:= "success" }	READY
2	READY	DL_SPDATA.request {DA, MSDU} / CHECK_PAR_SPDATA (DA, MSDU) = "True" && CHECK_SPDATAQ ( ) = "Full" ⇒ DL_SPDATA.confirm { DA, MSDU, Status:= "failure – The SPDATA Queue is full" }	READY
3	READY	DL_SPDATA.request {DA, MSDU} / CHECK_PAR_SPDATA (DA, MSDU) = "False" ⇒ DL_SPDATA.confirm { DA, MSDU, Status:= "Failure – Invalid parameter" }	READY
4	READY	SEND_ENABLE.ind {Speed-Class} / Speed-Class = "SPORADIC" && CHECK_SPDATAQ( ) <> "Empty" ⇒ DLPDU:= BUILD_SPDATA (DA, MSDU:= DEQUEUE_SPDATA ()) ACM_SEND_DATA.req { DLPDU } ACM_SEND_DATA.conf { } -- immediate response ⇒ (no action taken)	READY

5	READY	ACM_SEND_DATA.ind {DLPDU}
		/ Speed-Class == "SPORADIC" &&
		CHECK_SPDATA (DLPDU)
		⇒
		DL_Buffer_Received.indication {
		DLPDU.DLCEP-address
		}
			READY

7.9.4.4 Functions of STRC

All functions of the STRC are summarized in Table 44.

Table 44 – STRC Functions table

Function Name	Input	Output	Operation
CHECK_PAR_SPDATA	DA, MSDU	True/False	Check that all of parameters, DA and DSDU provided with DL_SPDATA.request are valid. If valid, "True" is returned.
CHECK_SPDATAQ	(none)	status	Check that the Queue condition for SPDATA is fully queued. The returned status is any one of "Full", "Empty" and "Queued".
QUEUE_SPDATA	DA, MSDU		Queues the input data into the SPDATA Queue on a FIFO basis.
DEQUEUE_SPDATA	(none)	DA, MSDU	Dequeue from the SPDATA queue on a FIFO basis.
BUILD_SPDATA	DA, MSDU	DLPDU	Build into DLPDU of Sporadic message data. DLPDU is assembled as follows. DLPDU.DA:= MSDU.DA DLPDU.SA:= V(IA) DLPDU.Len/Type:= MSDU.LEN/Type DLPDU.MSDU:= MSDU.DLSDU
CHECK_RCV_SPDATA	DLPDU	True/False	Check that the specified DLPDU is valid. If valid, “True” is returned.

7.9.5 Access Control Machine (ACM)

7.9.5.1 Overview

The Access Control Machine (ACM) is responsible for deterministic media access control and scheduling opportunities to send out the DLPDUs, for control to add and remove nodes and for restoration from network disruption to be system down.

The ACM has the primary responsibility for

a) Assuring that the local node detects and fully utilizes its assigned access time period.

b) Assuring that the local node does not interfere with the transmissions of other nodes, especially of the node transmitting the SYN DLPDU.

c) Detecting network disruption, and initiating the SYN DLPDU transmit for restoration of the network disruption from after prescribed time duration in which the SYN DLPDU is not heard.

d) Assuring a new node adding to and removing from the network.

7.9.5.2 ACM and the schedule support functions

The ACM functions schedule all communications between the DLEs participating in the RTE-TCnet, and the timing of this communications is controlled as to;

1) Fulfill the specific media access control to give all the DLEs the opportunities to send out 2 kinds of class of Time-critical Cyclic data and Sporadic message data in timely, prioritized and deterministic fashion, and to detect network disruption and to initiate the restoration in appropriate time, further to add and remove nodes on line.

2) provide 3 levels of Time-critical data transfer opportunities of sending data to node in sequential order and within each pre-specified time period, and that the data transfer of each level is performed within the pre-specified time duration (token holding time) and whether the data transfer of lower levels to be carried out or to be held over to later cyclic time period depends on the level and the occasion though the top level of the data transfer is always carried out at every occasion, on the other hand the whole volume of the data transfer of lower levels is transferred within each pre-specified time period.

3) provides sporadic message data transfer opportunities of sending out to node that the request to transfer is happened sporadically by the DLS-user, and the data transfer is performed in pre-specified time period of the corresponding level of priority and is based on regular ISO/IEC 8802-3 applications.

7.9.5.3 Primitive definitions

7.9.5.3.1 Primitives exchanged between ACM and RMC

Table 45 summarizes all primitives exchanged between ACM and RMC.

Table 45 – Primitives exchanged between ACM and RMC

Primitive names	Source	Associated parameters
RMC_SEND_DATA.req	ACM	RMSDU
RMC_SEND_DATA.conf	RMC	RMC_Status
RMC_RECV_DATA.ind	RMC	RMSDU

The parameters used with the primitives exchange the ACM and RMC are described in Table 46.

Table 46 – Parameters used with primitives exchanged between ACM and RMC

Parameter names	Description
RMSDU	SDU of RMC
RMC_Status	Status indicating the results of the request to RMC

7.9.5.3.2 Primitives exchanged between ACM and CTRC

Table 47 summarizes all primitives exchanged between ACM and CTRC.

Table 47 – Primitives exchanged between ACM and CTRC

Primitive names	Source	Associated parameters
SEND_ENABLE.ind	ACM	Speed-Class
ACM_SEND_DATA.req	CTRC	DLPDU
ACM_SEND_DATA.conf	ACM	ACM_status
ACM_RECV_DATA.ind	ACM	DLPDU

The parameters used with the primitives exchange the ACM and CTRC are described in Table 48.

Table 48 – Parameters used with primitives exchanged between ACM and CTRC

Parameter names	Description
ACM_Status	Status indicating the results
DLPDU	SDU of AMC
Speed-class	Speed class indicating Time-critical Cyclic data transmission

7.9.5.3.3 Primitives exchanged between ACM and STRC

Table 49 lists all primitives exchanged between ACM and STRC.

Table 49 – Primitives exchanged between ACM and STRC

Primitive names	Source	Associated parameters
SEND_ENABLE.ind	ACM	Speed-Class
ACM_SEND_DATA.req	STRC	DLPDU
ACM_SEND_DATA.conf	ACM	Status
ACM_SEND_DATA.ind	ACM	DLPDU

The parameters used with the primitives exchange the ACM and CTRC are described Table 50.

Table 50 – Parameters used with primitives exchanged between ACM and STRC

Parameter names	Description
ACM_Status	Status indicating the results
DLPDU	SDU of AMC
Speed-class	Speed class indicating Time-critical Cyclic data transmission

7.9.5.4 ACM state table

The state transition diagram is shown in Figure 29.

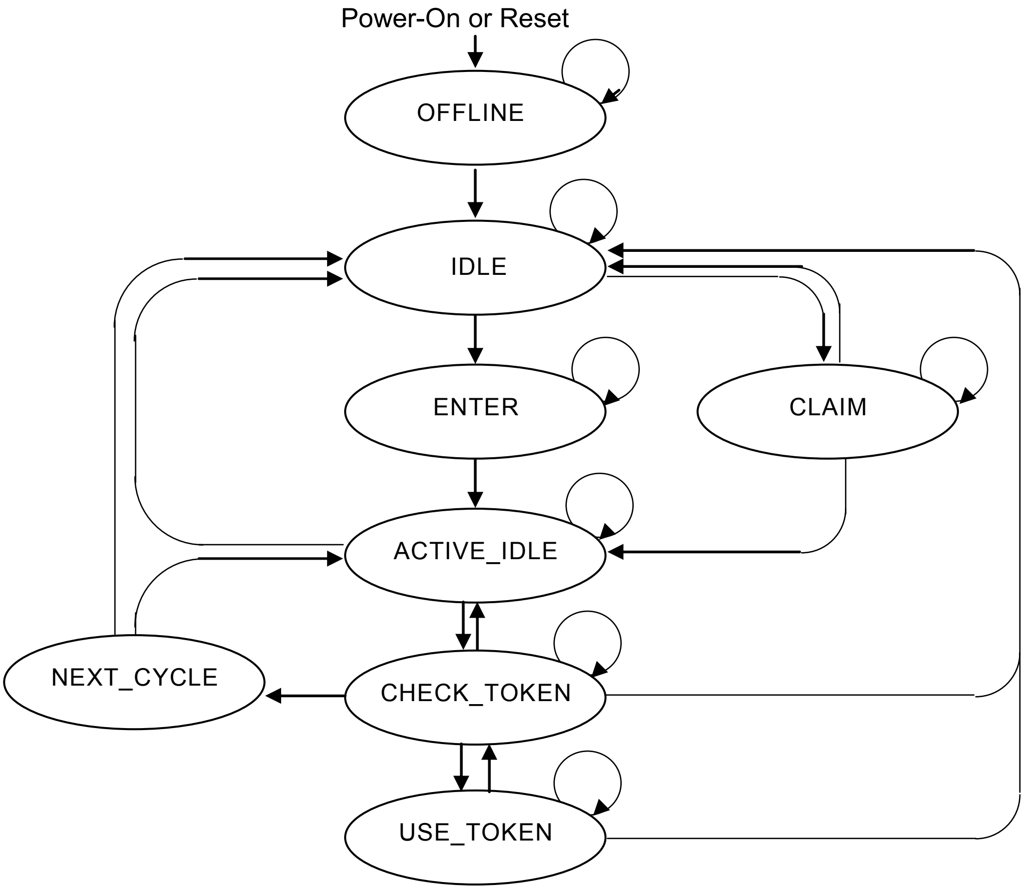


Figure 29 – State transition diagram of ACM

The state “OFFLINE” is entered on power up or when DLM_Reset.request primitive is issued. In this state, all of the variables, parameters and configuration information are set up by DLE. When "In_Ring_Desire" becomes true, the state changes to “IDLE".

The state "IDLE" is to wait and ready to join RTE-TCnet membership. When a SYN DLPDU has been received from other SYN node, the state changes to “ENTER” in which REQ DLPDU is sent out to claim adding into RTE-TCnet membership to current SYN node. Moreover in “IDLE” state, when the signals over the transmission media continues inactive during a specified time period, the state becomes to “CLAIM”, in which the node attempts to be new SYN node.

The state “ACTIVE_IDLE” is in RTE-TCnet membership, and the node manages to obtain the transmission right by RTE-TCnet media access control. Either When the node is operating as SYN mode by transmitting SYN DLPDU or as Non-SYN mode receiving SYN DLPDU by other SYN node, the state is in “CHECK_TOKEN”.

The state “CHECK_TOKEN” waits until V(Tsl) becomes equal to V(TN) in order to send out the data over the transmission media. When V(Tsl) becomes equal to V(TN), the state changes to “USE_TOKEN”.

The state “USE_TOKEN” is that the node is able to send out Time-critical Cyclic transmission data and Sporadic transmission data. When all data has been sent out or the token holding time is expired, the state changes to “NEXT_CYCLE”.

The state “NEXT_CYCLE” is for solicit new node as a primary function.

The ACM state table is shown in Table 51.

Table 51 – ACM state table

No.	Current State	Event /condition ⇒ actions	Next State
1	Any states	POWER-ON or RESET =>	OFFLINE
2	OFFLINE	In_RIng_Desired = "True" => START_TIMER (T(SL), V(SL))	IDLE
3	IDLE	RMC_RECV_DATA.ind { RMSDU } / RMSDU.FC = SYN && RMSDU.PN = V(TN) && V (LL)<V(TN)> = 0 => -- receiving SYN frame and LL this node not live -- SYN_frame:= RMSDU V(LN):= 1 START_TIMER (T(SL), V(SL))	ENTER
4	IDLE	EXPIRED_TIMER (T(SL)) = "True" => RMSDU:= BUILD_PDU (CLM) RMC_SEND_DATA.req { RMSDU } START_TIMER (T(SL), V(SL))	CLAIM
5	IDLE	RMC_RECV_DATA.ind { RMSDU } / => -- receiving a frame except the above -- START_TIMER (T(SL), V(SL))	IDLE
6	ENTER	V(LL)<V(LN)> = 1 && V(LN) <> V(TN) => START_TIMER (T(SCMP), V(Tsl))	ENTER
7	ENTER	V(LL)<V(LN)> = 0 && V(LN) <> V(MN) => V(LN)++	ENTER
8	ENTER	RMC_RECV_DATA.ind { RMSDU } / RMSDU.FC = CMP && V(LN) = V(MN) => -- Live Node = Maximum Node No. -- RMSDU:= BUILD_PDU (REQ) RMC_SEND_DATA.req { RMSDU } START_TIMER (T(SL), V(SL))	ENTER

Table 51 *(Continued)*

9	ENTER	RMC_RECV_DATA.ind { RMSDU } / RMSDU.FC = CMP && V(LN) <> V(MN) => -- Live Node less than Maximum Node No. -- V(LN)++ START_TIMER (T(SL), V(SL))	ENTER
10	ENTER	RMC_SEND_DATA.conf { } => (none)	ENTER
11	ENTER	RMC_RECV_DATA.ind { RMSDU } && FC = SYN / V (LL)<V(TN)> = 1 => SYN_frame:= RMSDU V(LN):= 1 START_TIMER (T(SL), V(SL))	ACTIVE_IDLE
12	ENTER	EXPIRED_TIMER (T(SCMP)) = "True" => V(LN)++	ENTER
13	ENTER	EXPIRED_TIMER (T(SL)) = "True" => -- Expired Silence Timer -- START_TIMER (T(SL), V(SL))	IDLE
14	CLAIM	RMC_SEND_DATA.conf { } / CHECK_COL () = "False" && All_Slot_Time <= V(MN) => All_Slot_Time++ START_TIMER (T(SL), V(SL))	CLAIM
15	CLAIM	RMC_SEND_DATA.conf { } / CHECK_COL ( ) = "False" && All_Slot_Time = V(MN) => V(LL) = 0 V(LL)<V(TN)>:= 1 RMSDU:= BUILD_PDU (SYN) SYN_frame:= RMSDU RMC_SEND_DATA.req { RMSDU } START_TIMER (T(SL), V(SL))	ACTIVE_IDLE
16	CLAIM	RMC_SEND_DATA.conf { } / CHECK_COL () = "True" => START_TIMER (T(SL), V(SL))	IDLE
17	CLAIM	RMC_RECV_DATA.ind { RMSDU } => -- receiving a frame except the above -- START_TIMER (T(SL), V(SL))	CLAIM
18	CLAIM	EXPIRED_TIMER (T(SL)) = "True" => -- Expired Silence Timer -- START_TIMER (T(SL), V(SL))	IDLE
19	ACTIVE_IDLE	RMC_RECV_DATA.ind { RMSDU } / RMSDU.FC = SYN => -- receiving SYN frame-- SYN_frame:= RMSDU V(LL):= SYN_frame.LL V(LN):= 1	CHECK_TOKEN
20	ACTIVE_IDLE	RMC_RECV_DATA.conf { RMSDU } / RMSDU.FC = SYN => -- sent SYN frame-- SYN_mode:= "True" V(LN):= 1	CHECK_TOKEN

Table 51 *(Continued)*

21	ACTIVE_IDLE	RMC_RECV_DATA.conf { RMSDU } / RMSDU.FC = DT => -- sending DT frame-- ACM_RECV_DATA.ind { RMSDU } START_TIMER (T(SL), V(SL))	ACTIVE_IDLE
22	ACTIVE_IDLE	RMC_RECV_DATA.conf { RMSDU } / RMSDU.FC <> RTE-TCnet FRAME => -- Receiving SPORADIC Frame -- ACM_RECV_DATA.ind { RMSDU } START_TIMER (T(SL), V(SL))	ACTIVE_IDLE
23	ACTIVE_IDLE	EXPIRED_TIMER (T(SL)) = "True" / => -- expired Silence Timer - SYN_mode:= "False" START_TIMER (T(SL), V(SL))	IDLE
24	CHECK_TOKEN	/ V(LL)<V(TN)> = 1 && V(LN) = V(TN) => -- Get token -- Speed-Class:= 3 – High Priority START_TOKEN_HOLD_TIMER () SEND_ENABLE.ind { Speed-Class } START_TIMER (T(MAC), V(MAC))	USE_TOKEN
25	CHECK_TOKEN	V(LL)<V(LN)> = 1 && V(LN) <> V(TN) => START_TIMER (T(SCMP), V(Tsl))	CHECK_TOKEN
26	CHECK_TOKEN	V(LL)<V(LN)> = 0 && V(LN) <> V(MN) => V(LN)++	CHECK_TOKEN
27	CHECK_TOKEN	/V(LN) = V(MN) && SYN_frame .PN = V(TN) && SYN_mode = "False" => -- Non-SYN mode, on MAC control time -- IF Required COM frame THEN RMSDU:= BUILD_PDU (COM) RMC_SEND_DATA.req { RMSDU } START_TIMER (T(SL), V(SL)) ENDIF	ACTIVE_IDLE
28	CHECK_TOKEN	/V(LN) = V(MN) && SYN_frame .PN <> V(TN) && SYN_mode = "True" => -- in SYN mode, on MAC control time -- START_TIMER (T(MAC), V(MAC))	NEXT_CYCLE
29	CHECK_TOKEN	RMC_RECV_DATA.ind { RMSDU } / RMSDU.FC = CMP && V(LN)++ = V(TN) => -- Get token -- -- matching V(LN) and on receiving CMP frame -- STOP_TIMER (T(SCMP)) Speed-Class:= 3 -- High Priority START_TOKEN_HOLD_TIMER () SEND_ENABLE.ind { Speed-Class } START_TIMER (T(SL), V(SL))	USE_TOKEN
30	CHECK_TOKEN	RMC_RECV_DATA.ind { RMSDU } / RMSDU.FC = CMP && V(LN)++ <> V(TN) => -- not matching V(LN) and on receiving CMP frame -- START_TIMER (T(SCMP), V(Tsl))	CHECK_TOKEN

Table 51 *(Continued)*

31	CHECK_TOKEN	RMC_RECV_DATA.ind { RMSDU } / RMSDU.FC = DT-CMP && V(LN)++ = V(TN) => -- matching V(LN) and on receiving DT-CMP frame -- Speed-Class:= 3 -- High Priority START_TOKEN_HOLD_TIMER () SEND_ENABLE.ind { Speed-Class } START_TIMER (T(SL), V(SL))	USE_TOKEN
32	CHECK_TOKEN	RMC_RECV_DATA.ind { RMSDU } / RMSDU.FC = DT-CMP && V(LN)++ <> V(TN) => -- not matching V(LN) and on receiving DT-CMP frame START_TIMER (T(SCMP), V(Tsl))	CHECK_TOKEN
33	CHECK_TOKEN	RMC_RECV_DATA.conf { RMSDU } / RMSDU.FC = DT => -- sending DT frame -- ACM_RECV_DATA.ind { RMSDU } START_TIMER (T(SL), V(SL))	CHECK_TOKEN
34	CHECK_TOKEN	RMC_RECV_DATA.conf { RMSDU } / RMSDU.FC <> RTE-TCnet FRAME => -- Receiving SPORADIC Frame -- ACM_RECV_DATA.ind { RMSDU } START_TIMER (T(SL), V(SL))	CHECK_TOKEN
35	CHECK_TOKEN	EXPIRED_TIMER (T(SCMP)) / => V(LN)++ if SYN_mode = "True" then MAINT_LL(V(LN)) endif START_TIMER (T(SL), V(SL))	CHECK_TOKEN
36	CHECK_TOKEN	RMC_SEND_DATA.conf { RMSDU } / RMSDU.FC = CMP => -- sending CMP frame-- V(LN)++	CHECK_TOKEN
37	CHECK_TOKEN	EXPIRED_TIMER (T(SL)) = "True" / => -- expired Silence Timer - SYN_mode:= "False" START_TIMER (T(SL), V(SL))	IDLE
38	USE_TOKEN	ACM_SEND_DATA.req { DLPDU } / => -- send DT or Sporadic frame -- RMPDU:= BUILD_PDU (DLPDU) RMC_SEND_DATA.req { DLPDU } START_TIMER (T(SL), V(SL))	USE_TOKEN
39	USE_TOKEN	RMC_SEND_DATA.conf { RMC_Status } / (DLPDU.FC = DT) || (DLPDU.FC <> RTE-TCnet && EXPIRED_TIMER (T(ATHT)) = "False" => ACM_SEND_DATA.conf { RMC_Status } SEND_ENABLE.ind { Speed-Class } START_TIMER (T(SL), V(SL))	USE_TOKEN

Table 51 *(Continued)*

40	USE_TOKEN	RMC_SEND_DATA.conf { RMC_Status } / (DLPDU.FC = DT) || (DLPDU.FC <> RTE-TCnet) && EXPIRED_TIMER (T(ATHT)) = "True" && Speed-Class <> 0 => -- Expired THT, Next Speed Class -- ACM_SEND_DATA.conf { RMC_Status } Speed-Class:= Speed-Class -1 START_THT (Speed-Class) START_TTRT (Speed-Class) SEND_ENABLE.ind { Speed-Class } START_TIMER (T(SL), V(SL))	USE_TOKEN
41	USE_TOKEN	ACM_SEND_DATA.req { NIL } / Speed-Class <> 0 => -- Next Speed-Class -- Speed-Class:= Speed-Class -1 START_THT (Speed-Class) START_TTRT (Speed-Class) SEND_ENABLE.ind { Speed-Class } START_TIMER (T(SL), V(SL))	USE_TOKEN
42	USE_TOKEN	ACM_SEND_DATA.req { NIL } / Speed-Class = 0 => -- All sent -- RMSDU:= BUILD_PDU (CMP) RMC_SEND_DATA.req { RMSDU } START_TIMER (T(SL), V(SL))	CHECK_TOKEN
43	USE_TOKEN	EXPIRED_TIMER (T(SL)) = "True" / => -- expired Silence Timer - SYN_mode:= "False" START_TIMER (T(SL), V(SL))	IDLE
44	NEXT_CYCLE	/EXIPERED_TIMER (T(MAC)) = "True" =>	ACTIVE_IDLE
45	NEXT_CYCLE	RMC_RECV_DATA.ind { RMSDU } / RMSDU.FC = REQ => -- receiving REQ from non-SYN node -- V(LL)<RMSDU.SN>:= 1 RMSDU:= BUILD_PDU (SYN) RMC_SEND_DATA.req { RMSDU } SYN_frame:= RMSDU START_TIMER (T(SL), V(SL))	ACTIVE_IDLE
46	NEXT_CYCLE	EXPIRED_TIMER (T(SL)) = "True" / => -- expired Silence Timer - SYN_mode:= "False" START_TIMER (T(SL), V(SL))	IDLE

7.9.5.4.1 Functions used by ACM

All the functions used by ACM is summarized in Table 52.

Table 52 – ACM Function table

Function Name	Input	Output	Operation
BUILD_PDU	FC	RMSDU	According to requested FC(Frame Control), assemble the RTE-TCnet frame
EXPIRED_TIMER	T(x)	True/False	When requested timer x has been expired, “True” is returned, otherwise False is returned.
MAINT_LL	V(LN)	(none)	The node specified V(LN) is read when T(SCMP) was expired. When T(SCMP) is expired all at number-of-times continuation of C(ASCMP) in the past, the corresponding bit of V(LL) is cleared.
START_TIMER	T(x), V(y)	(none)	Timer x is set by value of y, and is activated.
START_THT	Speed-Class	(none)	T(ATHT) is activated according to the V(TTRT) remaining value of Access-Class specified by Speed-Class.
START_TTRT	Speed-Class	(none)	T(ATTRT) is activated according to the V(TTRT) value of Access-Class specified by Speed-Class
STOP_TIMER	T(x)	(none)	Timer x is deactivated.

7.9.6 Redundancy medium control (RMC)

7.9.6.1 Overview

The redundancy medium control (RMC) manages the selection of two inputs from two RX framers, and moreover of two outputs to two Octet Serializers for reliable support, without interruption of higher-level data transfer services. The function of RMC is as follows.

a) Received DLPDUs from the ACM, the RMC breaks them down into octet symbol requests to both two Octet Serializers so that the DLPDUs are send out over both of the transmission medium A and B simultaneously.

b) If collision has happened during sending out DLPDUs over one of two transmission mediums, the data transmission over which collision happened is stopped immediately and the JAM signal specified by ISO/IEC 8802-3 is sent out for the next DLPDU ready to be sent out. Nevertheless if on another transmission medium there is no collision happens, the data transmission over the transmission medium is continued without interruption.

c) The RMC selects one of two outputs from the Octet Deserializers. The selection is on first-come basis to a frame without error, and the RMC submits octets of frame to the ACM.

d) The RMC provides a function purposely to select and designate one of two inputs from two RX framers. This function is managed according to the value of RMSEL parameter in the CW field of received SYN DLPDU. The relationship between the redundant medium selection and the value of RMSEL parameter is specified in 7.8.2.1.1.2.2. In the case of a single medium system the control command by RMSEL is ignored.

7.9.6.2 Primitive definitions

7.9.6.2.1 Primitives exchanged between ACM and RMC

Table 53 shows all of the primitives exchanged between ACM and RMC. Table 54 is the primitives between RMC and Serializer /Deserializer, furthermore the primitives between RMC and Ph-layer is shown in Table 55.

Table 53 – Primitives exchanged between ACM and RMC

Primitive names	Source	Associated parameters
RMC_SEND_DATA.req	ACM	RMSDU
RMC_SEND_DATA.conf	RMC	RMC_Status
RMC_RECV_DATA.ind	RMC	RMSDU

Table 54 – Primitives exchanged between RMC and Serializer / Deserializer

Primitive names	Source	Associated parameters
TX_DATA_A.req	RMC	RMPDU
TX_DATA_B.req	RMC	RMPDU
TX_DATA_A.conf	Serializer_A	Status
TX_DATA_B.conf	Serializer_B	Status
RX_DATA_A.ind	Deserializer_A	RMPDU, Status
RX_DATA_B.ind	Deserializer_B	RMPDU, Status

Table 55 – Primitives exchanged between RMC and Ph-layer

Primitive names	Source	Associated parameters
PLS_CARRIER_A.indication	Ph-layer_A	CARRIER_STATUS(ON/ OFF)
PLS_SIGNAL_A.indication	Ph-layer_A	SIGNAL_STATUS(ERROR/NO _ERROR）
PLS_DATA_VALID_A.indication	Ph-layer_A	DATA_VALID_STATUS(DATA_VALID/DATA_NOT_VALID）
PLS_CARRIER_A.indication	Ph-layer_B	CARRIER_STATUS(CARRIER_ON/CARRIER_OFF)
PLS_SIGNAL_B.indication	Ph-layer_B	SIGNAL_STATUS(SIGNAL_ERROR/NO_SIGNAL_ERROR）
PLS_DATA_VALID_A.indication	Ph-layer_B	DATA_VALID_STATUS(DATA_VALID/DATA_NOT_VALID）

7.9.6.2.2 Parameters used between RMC and ACM, Ph-layer

Table 56 and Table 57 show the parameters used for interaction between RMC and ACM or Ph-layer respectively.

Table 56 – Parameters between RMC and ACM

Parameter names	Description
RMSDU	Service data unit used between RMC and ACM
RMC_Status	Status for result of request, condition of receiving etc
TX_ENABLE	Enable/Disable control for data transmission

Table 57 – Parameters between RMC and Ph-layer

Parameter names	Description
CARRIER_STATUS	Status for condition of carrier signal on or not over medium
SIGNAL_STATUS	Status for collision happened
DATA_VALID_STATUS	Status for received data available

7.9.6.3 RMC state table

7.9.6.3.1 State table for Sending and Send arbitration

Figure 30 shows the state transition diagram of RMC Sending and Send arbitration. The state table of RMC Sending is shown in Table 58, and furthermore the state table of RMC Send arbitration is shown in Table 59.

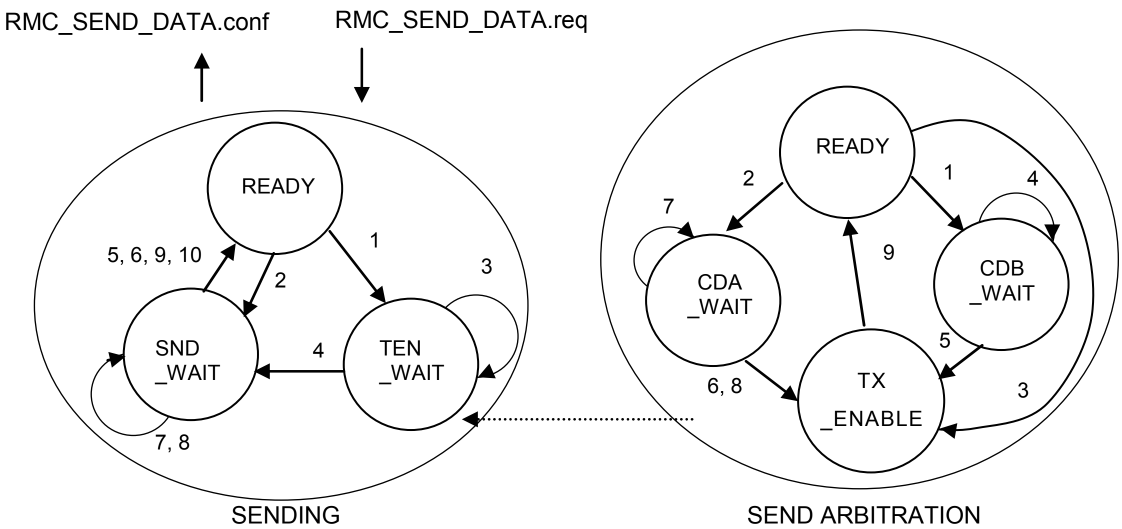


Figure 30 – State transition diagram of RMC Sending and Send arbitration

Table 58 – State table of RMC Sending

No.	Current state	Event /condition ⇒ actions	Next state
1	READY	RMC_SEND_DATA.req { RMSDU } / RMC_STATE <> "TX_ENABLE" ⇒ (no action)	TEN_WAIT
2	READY	RMC_SEND_DATA.req { RMSDU } / RMC_STATE = "TX_ENABLE" ⇒ ----- Under transmission enable SNDC = TX_DATA_AB (RMSDU)	SND_WAIT
3	TEN_WAIT	/ RMC-STATE <> "TX_ENABLE" ⇒ (no action)	TEN_WAIT
4	TEN_WAIT	/ RMC_STATE = "TX_ENABLE" ⇒ ------ Channel A into transmission enable SNDC = TX_DATA_AB (RMSDU)	SND_WAIT
5	SND_WAIT	TX_DATA_A.conf { } / SNDC = 1 ⇒ -------Transmission completed through Channel A RMC_SEND_DATA.conf { Status:= "success"}	READY
6	SND_WAIT	TX_DATA_B.conf { } / SNDC = 1 ⇒ -------Transmission completed through Channel B RMC_SEND_DATA.conf { Status:= "success"}	READY
7	SND_WAIT	TX_DATA_A.conf { } / SNDC = 2 ⇒ ---- Transmission completed on Channel A but B incomplete SNDC:= SNDC − 1	SND_WAIT
8	SND_WAIT	TX_DATA_B.conf { } / SNDC = 2 ⇒ ---- Transmission completed on Channel B but A incomplete SNDC:= SNDC − 1	SND_WAIT
9	SND_WAIT	TX_DATA_A.conf { } / SNDC = 0 ⇒ -------Transmission completed through Channel A RMC_SEND_DATA.conf { Status:= "fail"}	READY
10	SND_WAIT	TX_DATA_B.conf { } / SNDC = 0 ⇒ -------Transmission completed through Channel B RMC_SEND_DATA.conf { Status:= "fail"}	READY

Table 59 – State table of RMC Send arbitration

No.	Current state	Event /condition ⇒ actions	Next state
1	READY	PLS_CARRIER_A.indication { CARRIER_STATUS (OFF) } / ⇒ ---------------- Carrier Off over medium A START_TIMER (T(AIGP$_A$)) START_TIMER (T(ARMGP))	CDB_WAIT
2	READY	PLS_CARRIER_B.indication { CARRIER_STATUS (OFF) } / ⇒ --------------- Carrier Off over medium B START_TIMER (T(AIGP$_B$)) START_TIMER (T(ARMGP))	CDA_WAIT
3	CDB_WAIT	EXPIRED_TIMER (T(AIGPT)) = "True" / ⇒ --------------- No transmission from Channel A	TX_ENABLE
4	CDB_WAIT	PLS_CARRIER_B.indication { CARRIER_STATUS (OFF) } / ⇒ --------------- No transmission from both Channel A and B START_TIMER (T(AIGP$_B$)) STOP_TIMER (T(ARMGP))	CDB_WAIT
5	CDB_WAIT	EXPIRED_TIMER (T(AIGP$_B$)) = "True" / ⇒ --------------- Inter-packet-gap time elapsed over channel B	TX_ENABLE
6	CDA_WAIT	EXPIRED_TIMER (T(ARMGP)) = "True" / ⇒ --------------- No transmission from Channel A	TX_ENABLE
7	CDA_WAIT	PLS_CARRIER_A.indication { CARRIER_STATUS (OFF) } / ⇒ --------------- Carrier Off over both medium A and B START_TIMER (T(AIGP$_A$)) STOP_TIMER (T(ARMGP))	CDB_WAIT
8	CDB_WAIT	EXPIRED_TIMER (T(AIGP$_A$)) = "True" / ⇒ --------------- Inter-packet-gap time elapsed over channel A	TX_ENABLE
9	TX_ENABLE	/ ⇒ --------------- Signal TX_ENABLE to Sending state machine	READY

NOTE 1 Received DLPDUs from the ACM, the RMC breaks them down into octet symbol requests to both two Octet Serializers so that the DLPDUs are send out over both of the transmission medium A and B simultaneously.

NOTE 2 If collision has happened during sending out DLPDUs over one of two transmission mediums, the data transmission over which collision happened is stopped immediately and the JAM signal specified by ISO/IEC 8802-3 is sent out for the next DLPDU ready to be sent out. Nevertheless if on another transmission medium there is no collision happens, the data transmission over the transmission medium is continued without interruption.

NOTE 3 If collision has happened over both two transmission mediums, the JAM signal is sent out over both transmission mediums. In that case there happens no re-transmission.

NOTE 4 Both case of NOTE 2 and NOTE 3 are carried out on basis of DLPDU by DLPDU.

NOTE 5 TX_ENABLE signal is actually activated by later one of the expired timing of T(GP$_A$) and T(GP$_B$).

NOTE 6 After one of T(AIGP*) expired and further waited for the T(ARMGP) expired, if not received that the other T(AIGP*) is expired, the RMC decides the corresponding channel enters into abnormal condition. The reestablishment from abnormal state is by the expiration signal of the T(AIGP*) received.

7.9.6.3.2 State table for RMC Receiving

Figure 31 depicts the state transition diagram of RMC Receiving, and the state table of RMC Receiving is shown in Table 60.

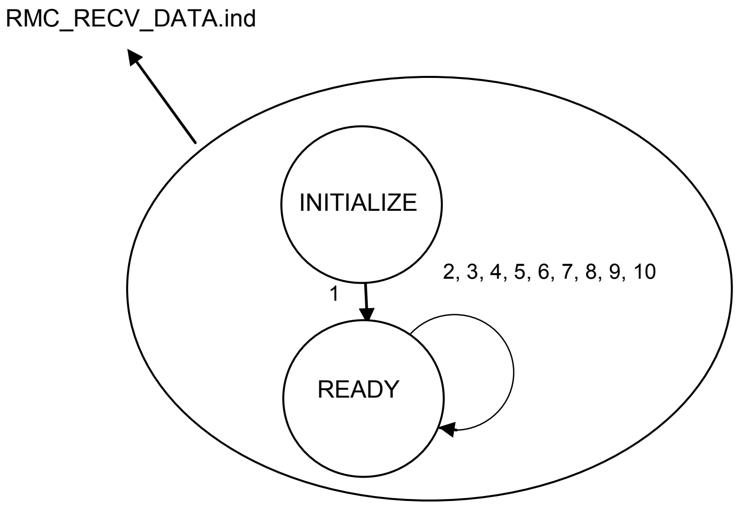


Figure 31 – State transition diagram of RMC Receiving

Table 60 – State table for RMC Receiving

No.	Current state	Event /condition ⇒ actions	Next state
1	INITIALIZE	POWER-ON or RESET / ⇒ -------------- Determine Receive channel RCVM:= DECIDE_RCVM ()	READY
2	READY	FORCE_RCVM(RCS) / ⇒ RCVM:= DECIDE_RCVM ()	READY
3	READY	RX_DATA_A.ind { RMPDU } / RCVM = MEDIA-A && Status:= NORMAL && RMPDU = "SYN" frame ⇒ LAST_RMA:= NORMAL RMSDU:= RMPDU RMC_RECV_DATA.ind { RMSDU } RCVM:= DECIDE_RCVM ()	READY
4	READY	RX_DATA_A.ind { RMPDU } / RCVM = MEDIA-A && Status:= "NORMAL" ⇒ LAST_RMA:= "NORMAL" RMSDU:= RMPDU RMC_RECV_DATA.ind { RMSDU }	READY

Table 60 *(Continued)*

5	READY	RX_DATA_A.ind { RMPDU } / RCVM <> MEDIA-A && Status:= "NORMAL" ⇒ LAST_RMA:= "NORMAL"	READY
6	READY	RX_DATA_A.ind { RMPDU } / Status:= "ERROR" ⇒ LAST_RMA:= ERROR RCVM:= DECIDE_RCVM ()	READY
7	READY	RX_DATA_B.ind { RMPDU } / RCVM = MEDIA-B && Status:= "NORMAL" && RMPDU = "SYN" frame ⇒ LAST_RMB:= "NORMAL" RMSDU:= RMPDU RMC_RECV_DATA.ind { RMSDU } RCVM:= DECIDE_RCVM ()	READY
8	READY	RX_DATA_B.ind { RMPDU } / RCVM = MEDIA-B && Status:= "NORMAL" ⇒ LAST_RMB:= "NORMAL" RMSDU:= RMPDU RMC_RECV_DATA.ind { RMSDU }	READY
9	READY	RX_DATA_B.ind { RMPDU } / RCVM <> MEDIA-B && Status:= "NORMAL" ⇒ LAST_RMB:= "NORMAL"	READY
10	READY	RX_DATA_A.ind { RMPDU } / Status:= "ERROR" ⇒ LAST_RMB:= "ERROR" RCVM:= DECIDE_RCVM ()	READY

NOTE 1 At initial stage Receive-Media A is selected.

NOTE 2 Switching control by “FORCE_RCV()” is executed immediately after received the command.

NOTE 3 Switching policy is as follows,

a) Receive-Media A or B is selected according to “Manual-select”.

b) Receive-Media A or B is selected according to the mode specified by RMSEL in SYN DLPDU received.

c) Once selected and without any error happened, the selected Receive-Media is maintained.

d) If any error happened, the other Receive-Media is selected.

7.9.6.4 Functions used by RMC

The functions used by RMC is summarized in Table 61.

Table 61 – RMC function table

Function name	Description and Operation
EXPIRED_TIMER (T(X))	When requested timer x has been expired, "True" is return, otherwise "False" is returned.
FORCE_RCVM (RCS)	if RCS:= Force_A then RCVM:= MEDIA-A endif if RCS:= Force_B then RCVM = MEDIA-B if RCS:= Automatic then RCVM:= DECIDE_RCVM()
START_TIMER (v(t))	Activate the timer specified . STOP_TIMER() forces to stop the timer specified and running. “Expired Timer” event happens at the time that a designated timer expired.
STOPT_TIMER(v(t))	Forcibly stop the timer specified.
DECIDE_RCVM ()	Receive-media is selected by "FORCE_RECV", "Manual-select", "RMSEL in SYN DLPDU" and is switched on a condition basis of previous DLPDU received under "Automatic" mode. Switching control is executed in the order of descending priority of “FORCE_RCVM”, “Manual-select”, “RMSEL in SYN DLPDU” and a condition basis under “Automatic” mode. If FORCE_RCVM(RCS:= MEDIA-A or:= MEDIA-B) then return MEDIA-A or MEDIA-B f else if “Manual-select”:= MEDIA-A then return MEDIA-A endif else if “Manual-select”:= MEDIA-B then return MEDIA-B endif else if RMSEL of SYNDLPDU:= MEDIA-A then return MEDIA-A endif else if RMSEL of SYN DLPDU := MEDIA-B then return MEDIA-B endif else if RCVM:= MEDIA-A && LAST_RMA <> NORMAL then return MEDIA-B endif else if RCVM:= MEDIA-B && LAST_RMB <> NORMAL then return MEDIA-A endif endif

TX_DATA_AB (PDU)	Send the PDU out over both Media-A and B at the same time, however it is prohibited to send out over the Media banned. The requested counts of TX_DATA_A or B. request is returned.
	
	sndc:= 0;
	if Media-A is permitted to send out
	then TX_DATA_A.req { PDU }
	sndc:= sndc + 1
	endif
	if Media-B is permitted to send out
	then TX_DATA_B.req { PDU }
	sndc:= sndc + 1
	endif
	return sndc

7.9.7 Serializer and Deserializer

Serializer and Deserializer is identical to ISO/IEC 8802-3 specification.

7.9.8 DLL Management protocol

7.9.8.1 Overview

The interface protocol between DLM and DLMS-user is described in this section. This section of the DLL management protocol provides the DLL management services specified in section 6.5.2.5 by making use of the services available by the DLMS-user.

In this section fully implementation matters and local matters are intentionally excluded.

7.9.8.2 Primitive definitions

7.9.8.2.1 Primitives exchanged between DLMS-User and DLM

Table 62 shows all primitives exchanged by DLMS-User and DLM.

Table 62– Primitives exchanged between DLMS-User and DLM

Primitive names	Source	Associated parameters
DLM_Reset.request	DLMS-User	(none)
DLM_Reset.confirm	DLM	DLM_Status
DLM_Set_Value.request	DLMS-User	Variable_Name, Desired-value
DLM_Set_Value.confirm	DLM	DLM_Status
DLM_Get_Variable.request	DLMS-User	Variable_Name
DLM_Get_Variable.confirm	DLM	DLM_Status, Current_value
DLM_Event.indication	DLM	Event-identifier, Additional-information
DLM_Set_Publisher_Configuration. request	DLMS-User	Desired-speed-class, Desired-configuration
DLM_Set_Publisher_Configuration. confirmation	DLM	DLM_Status
DLM_Get_Publisher_Configuration. request	DLMS-User	Desired-speed-class
DLM_Get_Publisher_Configuration. confirmation	DLM	DLM_Status, Current-configuration
DLM_Activate_TCC.request	DLMS-User	Desired-speed-class
DLM_Activate_TCC.confirmation	DLM	DLM_Status
DLM_Deactivate_TCC.request	DLMS-User	Desired-speed-class
DLM_Deactivate_TCC.confirmation	DLM	DLM_Status

7.9.8.2.2 Parameters used with primitives exchanged between DLMS-User and DLM

All parameters used with primitives exchanged between the DLMS-User and the DLM is summarized in Table 63.

Table 63 – Parameters used with primitives exchanged between DL-user and DLM

Parameter names	Description
DLM_Status	Status of the service execution
Variable_Name	DL variable names to be addressed
Desired_value	DL variable value to be set
Current_value	Current DL variable value to be read out
Event-identifier / Additional-information	Event whose occurrence is being announced with related DL variable and the state changed.
Desired-speed	Speed-class for the Time-critical Cyclic data transmission
Desired-configuration	A set of DLCEP-identifiers to be set as a publisher for the Time-critical Cyclic data transmission with desired-speed class
Current-configuration	A current set of DLCEP-identifiers set as a publisher for the Time-critical Cyclic data transmission with desired-speed class

7.9.8.3 DLM state table

The DLM state transition diagram are shown in Figure 32 and Table 64.

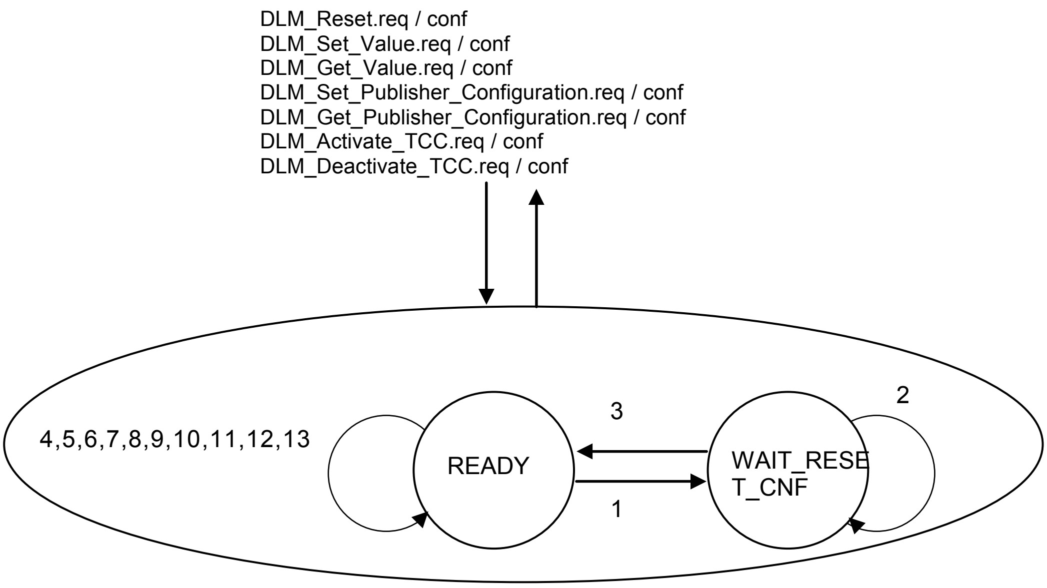


Figure 32 – State transition diagram of DLM

Table 64 – DLM state table

No.	Current state	Event /condition ⇒ actions	Next state
1	READY	DLM_Reset.req {} ⇒ MAC_reset:= MAC_RESET() RMC_reset:= “FALSE”	WAIT_RESET_CNF
2	WAIT_RESET_CNF	/ MAC_reset:= “TRUE” && RMC_reset:= “FALSE” ⇒ RMC_reset:= RMC_RESET	WAIT_RESET_CNF
3	WAIT_RESET_CNF	/ ⇒ -- MAC and RMC reset completed RESET_VAR ( ) DLM_RESET.cnf { DLM_Status:= "Success")	READY
4	READY	DLM_Set_Value.req {Variable_name, Desired_value } / CHECK_VALUE (Variable_name, Desired_value) = "Valid" ⇒ SET_VALUE (Variable_name, Desired_value) DLM_Status:= "Success" DLM_Set_Value.conf { DLM_Status }	READY
5	RE ADY	DLM_Set_Value.req {Variable_name, Desired_value } / CHECK_VALUE (Variable_name, Desired_value) <> "Valid" ⇒ DLM_Status:= "Failure" DLM_Set_Value.conf { DLM_Status }	READY
6	READY	DLM_Get_Value.req {Variable_name } / CHECK_VAR (Variable_name) = "Valid" ⇒ Current_value:= GET_CURRENT_VAL (Variable_name) DLM_Status:= "Success" DLM_Get_Value.conf { Current_value, DLM_Status }	READY
7	READY	DLM_Get_Value.req {Variable_name } / CHECK_VAR (Variable_name) <> "Valid" ⇒ Current_value:= NIL DLM_Status:= "Failure" DLM_Get_Value.conf { Current_value, DLM_Status }	READY
8	READY	DLM_Set_Publisher_Configuration.req { Desired-speed-class, Desired-configuration } / CHECK_PC (Desired-speed-class, Desired-configuration) = "Valid" ⇒ SET_PC (Desired-speed-class, Desired-configuration) DLM_Status:= "Success" DLM_Set_Publisher_Configuration.conf { DLM_Status }	READY
9	READY	DLM_Set_Publisher_Configuration.req { Desired-speed-class, Desired-configuration } / CHECK_PC (Desired-speed-class, Desired-configuration) <> "Valid" ⇒ DLM_Status:= "Failure" DLM_Set_Publisher_Configuration.conf { DLM_Status }	READY

Table 64 – *(continued)*

No.	Current state	Event /condition ⇒ actions	Next state
10	READY	DLM_Get_Publisher_Configuration.req { Desired-speed-class } / CHECK_SPEED (Desired-speed-class) = "Valid" ⇒ Current-configuration:= GET_PC (Desired-speed-class) DLM_Status:= "Success" DLM_Get_Publisher_Configuration.conf { DLM_Status, Current-configuration }	READY
11	READY	DLM_Get_Publisher_Configuration.req { Desired-speed-class } / CHECK_SPEED (Desired-speed-class) <> "Valid" ⇒ Current-configuration:= NIL DLM_Status:= "Failure" DLM_Get_Publisher_Configuration.conf { DLM_Status, Current-configuration }	READY
12	READY	DLM_Activate_TCC.req { Desired-speed-class } / CHECK_ATCC (Desired-speed-class) = "Valid" ⇒ STOP_TCC (Desired-speed-class) DLM_Status:= "Success" DLM_Activate_TCC.conf { DLM_Status }	READY
13	READY	DLM_Deactivate_TCC.req { Desired-speed-class } / CHECK_DTCC (Desired-speed-class) = "Valid" ⇒ START_TCC (Desired-speed-class) DLM_Status:= "Success" DLM_Deactivate_TCC.conf { DLM_Status }	READY
14	READY	DLM_Deactivate_TCC.req { Desired-speed-class } / CHECK_DTCC (Desired-speed-class) <> "Valid" ⇒ DLM_Status:= "Failure" DLM_Deactivate_TCC.conf { DLM_Status }	READY

7.9.8.4 Function

The DLM Functions are summarized in Table 65.

Table 65 – DLM function table

Function name	Description and Operation
CHECK_PC (Desired-speed-class, Desired-configuration)	Check that the requested Desired-speed-class and Desired-configuration are valid. Possible Desired-speed-class value are: High-speed Medium-speed or Low-speed Check that DLCEP-address value and total amount of Desired-configuration. Furthermore, when TCC data transmission of the requested Desired-speed-class is inactive, the value “valid” is returned.
CHECK_SPEED (Desired-speed-class)	Check that the requested Desired-speed-class are valid. Possible Desired-speed-class are: High-speed Medium-speed Low-speed
CHECK_VALUE (Variable-name, Desired_value)	Check that the requested Variable with Desired value are valid. Possible variables with the value are defined in the corresponding part of clause 7 Data Link Layer Protocol speciation.
CHECK_VAR (Variable_name)	Check that the requested Variable are valid. Possible variables are defined in the corresponding part of clause 7 Data Link Layer Protocol speciation.
CHECK_ATCC (Desired-speed-class)	Check that the requested Desired-speed-class are valid. Possible Desired-speed-class are: High-speed Medium-speed Low-speed Furthermore, when TCC data transmission of the requested Desired-speed-class is inactivate, the value “valid” is returned.
CHECK_DTCC (Desired-speed-class)	Check that the requested Desired-speed-class are valid. Possible Desired-speed-class are: High-speed Medium-speed Low-speed Furthermore, when TCC data transmission of the requested Desired-speed-class is inactivate, the value “valid” is returned.
GET_CURRENT_VAL (Variable-name)	Get the value of requested Variable.
RESET_VAR （）	Initialize all valuables and the set of the Publisher-configuration for each Speed-class. TCC data transfer service is deactivated.
START_TCC (Desired-speed)	Activate the TCC data transmission of requested Desired-speed-class.
STOP_TCC (Desired-speed)	Deactivate the TCC data transmission of the requested Desired-speed-class.

8 Application Layer Service definitions

8.1 Introduction

This part of RTE-TCnet specification is one of a series produced to facilitate the interconnection of automation system component. It is related to other standards in the set as defined by the "three-layer" Fieldbus Reference model, which is based in part on the Basic Reference Model for Open Systems Interconnection. Both Reference Models subdivide the area of standardization for interconnection into a series of layers of specification, each of manageable size.

The Application Layer Service is provided by the Application Layer Protocol making use of the services available from the Data Link or other immediately lower layer. This part defines the Application Layer Service characteristics that any immediately higher-level protocol may exploit. The relationship between the International Standards for Fieldbus Data Link Service, Fieldbus Data Link Protocol, Fieldbus Application Service, Fieldbus Application Protocol and System Management is illustrated in Figure 33.

NOTE Systems Management, as used in this PASs a local mechanism for managing the layer protocols.

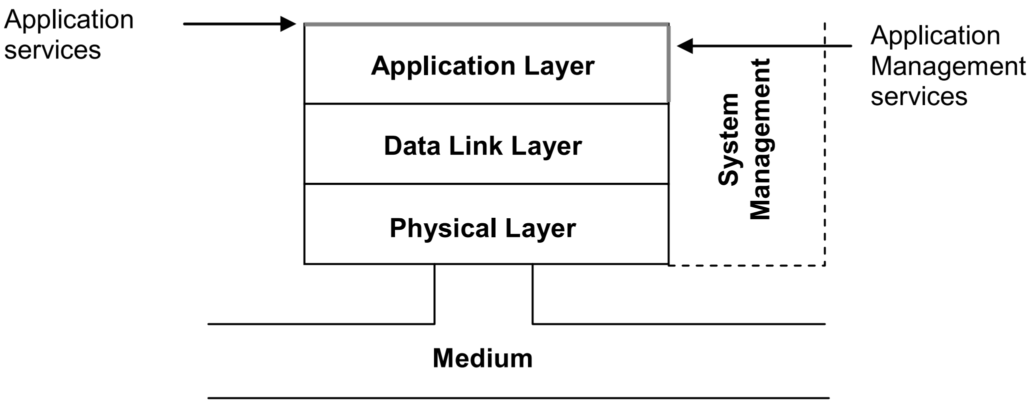


Figure 33 – Relationship of the RTE-TCnet Application layer to other RTE layers and to users of RTE Application Layer service

8.2 Scope

The RTE Application Layer provides user programs with a means to access the RTE communication environment. In this respect, the RTE Application Layer can be viewed as a "window" between corresponding application program.

RTE-TCnet Application Layer is an Application Layer Communication Specification designed to support the conveyance of time-critical and non-time-critical application requests and responses among devices in an automation environment. The term "time-critical" is used to represent the presence of an application time-window, within which one or more specified actions are required to be completed with some defined level of certainty.

This part specifies the structure and services of the RTE-TCnet Application Layer. It is specified in conformance with the OSI Basic Reference Model (ISO/IEC 7498) and the OSI Application Layer Structure (ISO/IEC 9545)

RTE-TCnet Application services and protocols are provided by RTE-TCnet application-entities (AE) contained within the application process. The RTE-TCnet 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 ASEs is a management ASE that provides a common set of services for management of the instance of RTE-TCnet Application layer classes.

This part of RTE-TCnet specifies interactions between remote applications in term of

- an abstract model for defining application resources (objects) capable of being manipulated by users via the use of RTE-TCnet Application Layer Service;
- the primitives (interactions between the RTE-TCnet Application Layer and the RTE-TCnet Application user) associated with each RTE-TCnet Application Layer Service;
- the parameters associated with each primitive;
- the interrelationship between and the valid sequences of the primitives for each service.

Several models of communications are specified in this part. Each model is specified as a communication "type". Each type has its own separate clause in the document.

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 RTE-TCnet users in standardizing such object behavior. In addition to these services, some supporting services are also defined in this PAS to provide access to the RTE-TCnet Application layer to control certain aspects of its operation.

8.3 Void

8.4 Term and definitions

8.4.1 ISO/IEC 7498-1 terms

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

8.4.2 ISO/IEC 8822 terms

For the purpose of this PAS, the following terms as defined in ISO/IEC 8822 apply.

a) abstract syntax
b) presentation context

8.4.3 ISO/IEC 9545 terms

For the purpose of this PAS, the following terms as defined in ISO/IEC 9545 apply.

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

8.4.4 ISO/IEC 8824-1 terms

For the purpose of this PAS, the following terms as defined in ISO/IEC 8824 apply.

a) object identifier
b) type

8.4.5 Fieldbus Data Link Layer terms

For the purpose of this PAS, the following terms as defined in IEC 61158-3 and IEC 61158-4 apply.

a) DLCEP
b) DLC
c) DLPDU
d) DLSDU
e) DLSAP
f) fixed tag
g) generic tag
h) link
i) MAC ID
j) network address
k) node address
l) node
m) tag
n) scheduled
o) unscheduled

8.4.6 Fieldbus Application Layer terms

For the purpose of this PAS, the following definitions apply.

a) application
b) application layer interoperability
c) application objects
d) application process

e) application process identifier
f) application process object
g) application process object class
h) application relationship
i) application relationship application service element
j) application relationship endpoint
k) attribute
l) behavior
m) bit-no
n) class
o) class attributes
p) class code
q) class specific service
r) connection
s) conveyance path
t) cyclic
u) dedicated AR
v) endpoint
w) frame
x) pre-established AR endpoint
y) publisher
z) push publisher
aa) service
ab) subscriber

8.4.7 RTE-TCnet specific terms

For the purpose of this PAS, the following definitions apply.

8.4.7.1 Common memory

Virtual common memory over RTE-TCnet, which is shared by participating RTE-TCnet nodes and is primarily used for the Time-critical Cyclic data to distribute in temporal and spatial coherency.

8.4.8 Symbols and abbreviations

AE	Application Entity
AL	Application Layer
AP	Application Process
APDU	Application Protocol Data Unit
AR	Application Relationship
AREP	Application Relationship End Point
ASE	Application Service Element
CM	Common Memory
DL-	(as a prefix) Data Link-
DLC	Data Link Connection

DLCEP	Data Link Connection End Point
DLL	Data Link Layer
FAL	Fieldbus Application Layer
IEC	International Electro technical Commission
ind	Indication
ISO	International Organization for Standardization
PDU	Protocol Data Unit
Req	Request
RTE	Real Time Ethernet
SAP	Service Access Point
SDU	Service Data Unit

8.4.9 Conventions

8.4.9.1 General

This PAS uses the descriptive conventions given in ISO/IEC 61158-5, 3.8.

8.4.9.2 Conventions for RTE-TCnet

None.

8.5 Concept

8.5.1 Overview

The RTE-TCnet Application Layer service utilizes RTE-TCnet specific common memory system, so-called Common Memory. The Common Memory is a virtual common memory over RTE-TCnet, is used and globally shared by the participating node, application processes running over each node. Further the Common Memory by means of Time-critical Cyclic data transfer services provides data distribution in temporal and spatial coherency.

Size and capacity depends on the implementation. However the Common Memory is divided into numbers of block with several size of memory. The number and the size depends also on implementation. Time-critical Cyclic data transfer, specified by the DL-service and the DL-protocol in this PAS, is carried out on each data block basis and each data of block is multicasted to member nodes from a node as publisher.

Each block in the Common Memory is associated with one of Application Relationship End Point (AREP), and is identified by the AREP and is used by multiple application processes in common. The block is a container for application data in general use and provides flexibility to apply in a variety of industrial application processes.

Figure 34 shows one of the application examples using the Common Memory system.

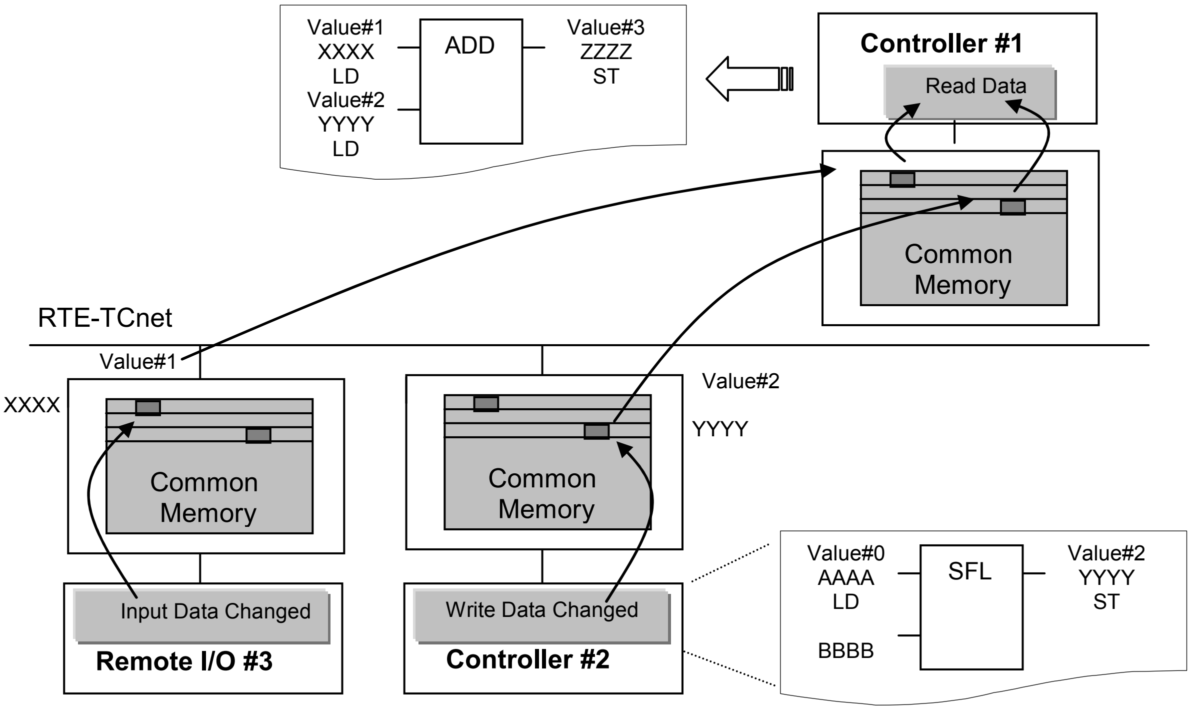


Figure 34 – RTE-TCnet Application (Example)

8.5.2 Common memory concept

Figure 35 shows the concept of RTE-TCnet Cyclic data transmission with common memory. It utilizes a cyclic broadcast transmission mechanism with common memory that is actually implemented in each node and given the same address space on the network. The common memory is divided into dedicated areas for each node's transmitting data. This is refreshed in the same memory area of all nodes on a fixed cyclic period. By this means, the controllers can quickly access each other's data avoiding troublesome communication procedures.

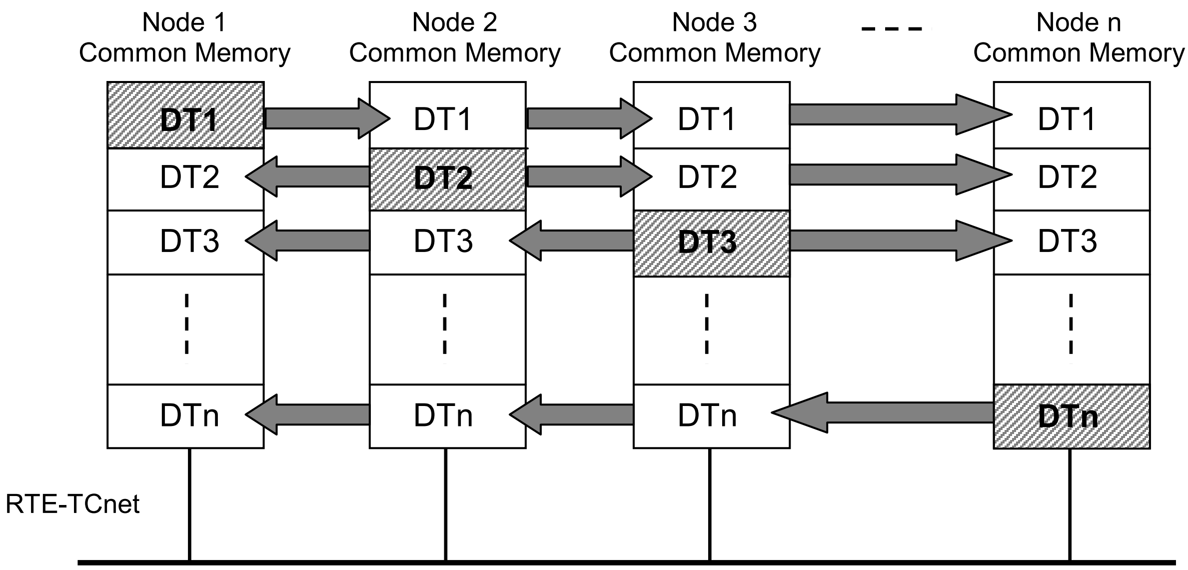


Figure 35 – Global Common Memory concept over RTE-TCnet

8.5.3 Relationship of common memory and AREP

The Common Memory is divided into numbers of block with several size of memory, of which. number and size depends on implementation however the size is recommended from 16 to 64 for efficient application.

Each block in the Common Memory is associated with one of Application Relationship End Point (AREP), which is unique, is commonly used and identified in the RTE-TCnet domain. The unique number assigned to each block associated with one of AREP is used to identify and determine actual position of the Common Memory address.

Each node is assigned a number of blocks of AREPs as Publisher, and broadcasts data of each block, receives data of each block from other node as a subscriber and updates the contents of the corresponding block on the local physical memory which is identical configuration to the Common Memory.

When on creation of new AREP, AL-user specifies three kind of class, that is High-Speed、Medium-Speed and Low-Speed class, to the AREP.

Figure 36 depicts the relationship between the Common Memory and the AREP on each node.

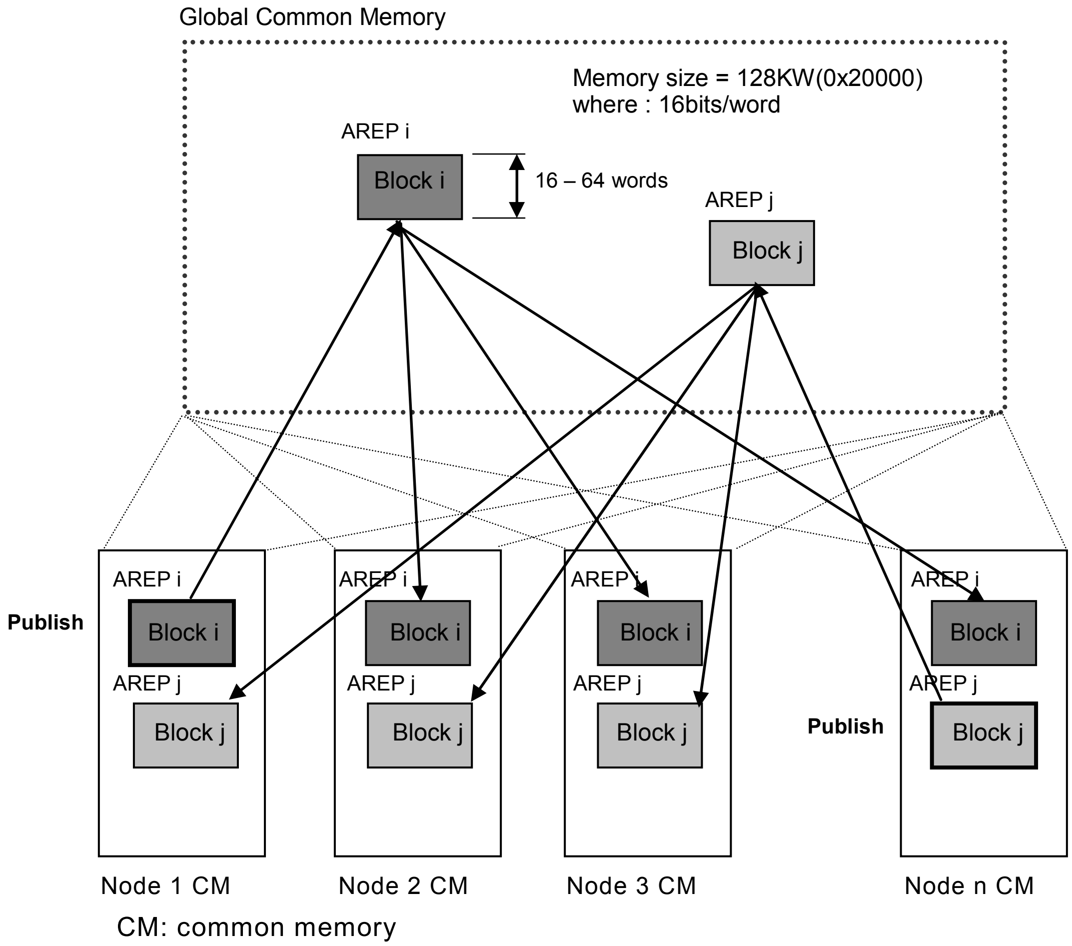


Figure 36 – Relationship of Common Memory and AREP

8.5.4 Common memory data type

The data type applied to a block which is associated with one of AREP, is primarily based on the basic data type defined in IEC 61158-5, Clause 5. Using these data type, Array or Structure is built.

The primitive data types forming Array or Structure in a block of the Common Memory is as follows.

Boolean;

BitString8;

BitString16;

BitSting32;

BinaryDate2000;

DATE;

TimeOfDay without date indication;

TimeDifference without date indication;

Integer8;

Integer16;

Integer32;

Unsigned8;

Unsigned16;

Unsigned32;

Float32;

BInaryTime0;

BInaryTime1;

BInaryTime2;

BInaryTime3;

BInaryTime4;

BInaryTime5;

BInaryTime6;

BInaryTime7;

BInaryTime0;

OctetString2;

OctetString4;

VisibleString2;

VisibleString4.

8.5.5 RTE-TCnet ASE and services

The RTE-TCnet Application Layer provides the Update_Memory service for performing to update the contents of the Common Memory. For this purpose, RTE-TCnet AE is provided with the Common Memory ASE and the AR ASE.

Figure 37 depicts the structure or RTE-TCnet AL ASE.

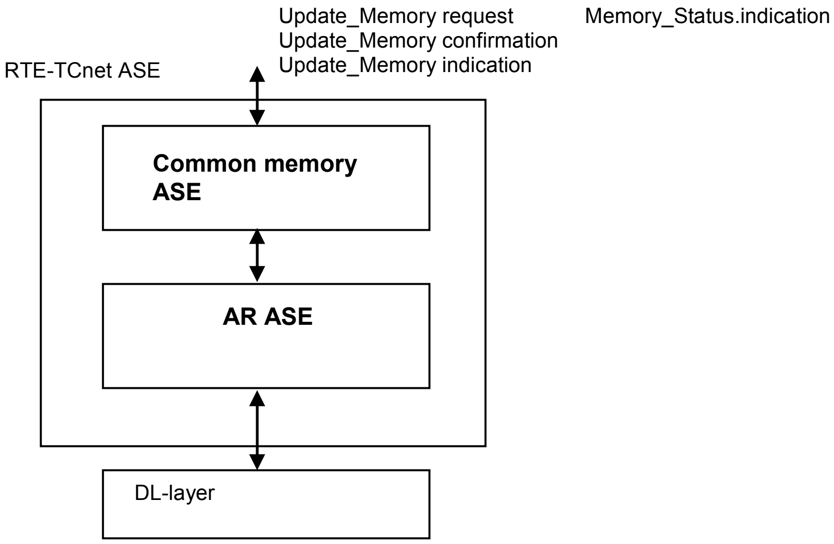


Figure 37 – Structure of RTE-TCnet AL ASE.

8.6 Common memory ASE

8.6.1 Overview

The Common Memory ASE provides the Update_Memory service for the ALS-user to read and write data from/to the Common Memory in order to intercommunication between processes running on remote nodes.

8.6.2 Common memory model class specification

8.6.2.1 Common memory formal model

ASE: COMMON MEMORY ASE

CLASS: COMMON MEMORY

CLASS ID:

PARENT CLASS: TOP

ATTRIBUTES:

1	(m)	Key Attribute:	Not used
2	(m)	Attribute:	Role (Publisher, Subscriber)
3	(m)	Attribute:	State
4	(m)	Attribute:	Common memory
4.1	(m)	Attribute:	Total memory size
4.2	(m)	Attribute:	Block memory size

SERVICES:

1	(o)	Ops Service	Update-Memory
2	(o)	Ops Service	Memory-Status

8.6.2.2 Attributes

Role

This attribute specifies the role for each common memory block with the same meaning as AREP. The valid values are:

Publisher Endpoint of this type publishes their data by issuing Update-Memory service request primitive;

Subscriber Endpoint of this type subscribes the data in response to Update-Memory service indication primitive.

State

This attribute indicates the current state of the Common Memory ASE (FSPM) that is defined in detail in clause 9.

Common memory

The following attributes specify the capacity of the Common Memory.

Total Memory size

The capacity of the whole common memory is specified in octet.

Block Memory size

The size of the block associated with a AREP is specified in octet.

8.6.2.3 Services

Update Memory

The Update_Memory request primitive is used by the ALS-user to update the content of a bock of the Common Memory as a publisher.

The Update_Memory indication primitive is used to notify the ALS-user that the designated block is updated.

Memory-Status

The Memory Status indication primitive is used to notify the ALS-user the timing to publish.

8.6.3 Service specification of Common memory (CM) ASE

8.6.3.1 Supported services

The services provided by Common Memory ASE are as follows.

Update Memory

Memory-Status

8.6.3.2 Update Memory service

8.6.3.2.1 Service overview

The Update_Memory service is used by the ALS-user to update the content of the bock of the Common Memory as a publisher. AR is based on the Push model of Publisher and Subscriber.

8.6.3.2.2 Service primitives

The service parameters for this service is shown in Table 66.

Table 66 – Update Memory service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
AREP	M	M(=)		
Memory Contents	M	M(=)		

Argument

The argument contain the parameters of the Update-Memory service request.

Memory Contents

This parameter contains the Common memory corresponding to the AREP.

8.6.3.2.3 Service procedure

The ALS-user issues Update_Memory request primitive to the Common Memory ASE, and the ASE assembles APDU in unacknowledged type and hands over the APDU to the AR ASE.

The ASE on remote node as subscriber informs the ALS-user on reception of the APDU from other remote node using Update_Memory indication primitive.

8.6.3.3 Memory-Status service

8.6.3.3.1 Service overview

Memory Status service informs the ALS-user the status of the local memory identical the global Common Memory.

8.6.3.3.2 Service primitives

The service parameters for this service are shown in Table 67.

Table 67 – Memory-Status service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument		M		
AREP		M		
Memory status code		M		

Argument

This argument contains the parameter corresponding to Memory-Status service.

Memory status code

This parameter indicates the status of the Common Memory.

Now-Updating. -- for use in Publisher side

8.6.3.3.3 Service procedure

The ASE issues the Memory_status indication primitive to the ALS-user.

8.7 Application Relationship ASE

8.7.1 Overview

8.7.1.1 General

Figure 38 shows the Common Memory Publisher/Subscriber model. The set of AREPs which is commonly used in RTE-TCnet, is predefined and pre-established in static connection of each other. The relationship between local and remote APs is distinguished by the AREP identifier. In relationship between Publisher and Subscriber associated with AREP identifier, the AR is one to many relationship.

Figure 38 also depicts the data flow on the model of AREP class for Publisher/Subscriber in the Common Memory, that is Buffered Network-Scheduled Uni-directional Pre-Established Connection（BNU-PEC) AREP.

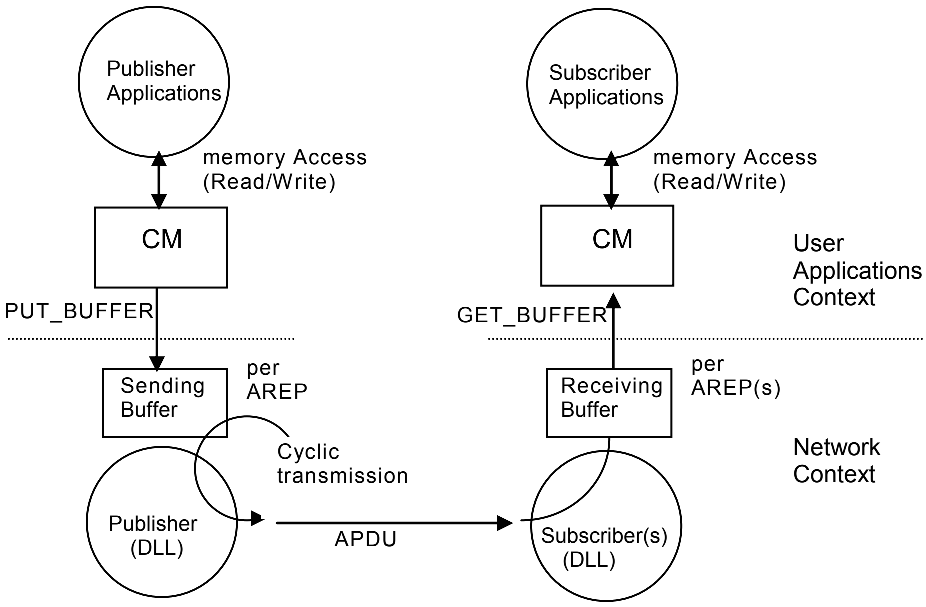


Figure 38 – Common Memory Publisher/Subscriber model

8.7.1.2 AR-endpoint

The application on the Common Memory uses the predefined common AREP among each other. For one AREP a single publisher is defined and assigned in RTE-TCnet. If defined duplicated publishers for one AREP, that shall be alarmed and notified.

8.7.1.3 AR-endpoint class

The AR is established at each AREP common in RTE-TCnet domain.

8.7.1.4 AR cardinality

ARs characterize communications between APs. One of the characteristics of an AR is the number of AREPs in the AR. BNU-PEC AREPs convey services from one AP to a number of APs and have a cardinality of one -to-many.

8.7.1.5 Accessing object through ARs

The AR provides with the means to access the Common Memory using the services by the ASE.

8.7.1.6 AR conveyance paths

The AR is modelled as the communication path with uni-directional data transfer among AREPs. The AREP at the receiver side on the AR receives the data sent by the AREP at the sender side.

8.7.1.7 AREP role

The role of AREP is based on the Publisher / Subscriber model with Push type.

8.7.1.8 AREP buffers

The AREP is modelled from the viewpoint of data buffer provided with the DL service and protocol. The APDU to be conveyed over through the BNU-PEC AREP, is storing into the Send-buffer of the DL and is sent out by the DLE over the transmission media.

Once the Send-buffer updated, the APDU is sent out. In that case the APDU is retained in the Send-buffer and is read out from the Send-buffer for cyclic data transmission.

On the contrary at the receiver side the situation is inversive.

The APDU received at the receiver side is stored into the Receive-buffer of the DL. The consecutive APDU received is overwritten to the Receive-buffer of Data Link layer.

The reading out of the Receive-buffer is not in destructive nature and the APDU received is maintained in the Receive-buffer so that the ALS-user can read out over and over.

8.7.1.9 Network scheduled conveyance

The BNU-PEC AREP cyclically issues the request to send out data at the interval specified. The interval is controlled and maintained by the DLE.

8.7.1.10 Identification of ARs

The AR is identified and established by the DLCEP-identifier.

8.7.1.11 Definitions and establishment of AREPs

The definition of the AREP specifies the instance of an AREP class.

The AREPs are pre-defined and pre-established.

8.7.1.12 AR establishment and termination

The AREP is predefined and is identified by the AREP identifier, of which assignment is maintained until power-off or RESET request by the MLE.

8.7.2 Application relationship endpoint class specification

8.7.2.1 Description

For RTE-TCnet new AREP of BNU-PEC are defined and supported.

The corresponding ARs carry out Unconfirmed Services (UCS) over a buffer, network scheduled , unidirectional AR which utilizes a pre-established data link connection.

8.7.2.2 Formal model

ASE: **AR ASE**

CLASS: **Buffered Network Scheduled Unidirectional Pre-Established Connection (BNU-PEC) AREP**

CLASS ID:

PARENT CLASS: **AR Endpoint** (Refer IEC 61158-5, 6.2.3.2**)**

ATTRIBUTE:

1	(m)	Attribute:	Role (Publisher, Subscriber)
2	(o)	Attribute:	Dedicated (TRUE, FALSE)
3	(m)	Attribute:	State
4	(m)	Attribute:	DL Mapping Reference

SERVICES:

1	(m)	Ops Service	AR-Unconfirmed Send
2	(m)	Ops Service	AR-Get-Buffered-Message
3	(o)	Ops Service	AR-Status

8.7.2.3 Attributes

Role

This attribute specifies possible role of this end point. The possible value are Publisher and Subscriber.

Publisher AREPs of this type publish their data issuing unconfirmed service request-PDUs.

Subscriber AREPs of this type receive data published in unconfirmed service request-PDUs.

Dedicated

This attribute specifies whether the publisher AREP is dedicated to publish one common memory block.

State

This attribute indicates the current state of Application Relationship Protocol Machine (ARPM) that is defined in detail in clause 9. The possible value for this attribute are specified there.

DL Mapping Reference

For Publisher AREPs, this attribute specifies the mapping to the transmit conveyance path. For Subscriber AREPs, this attribute specifies the mapping to the receive conveyance path. DL mapping attributes for the Data Link Layer (Clause 6) are specified in Clause 9.

8.7.2.4 Services

AR-Unconfirmed Send

This local service is used to send an unconfirmed service on the specified AR. The structure of the PDU is specified and described in the corresponding DL-Mapping State Machine.

AR-Get-Buffered-Message

This local service is used to retrieve an APDU from the buffer used by an AR.

AR-Status

This optional service is used to report status of the AR.

8.7.3 Application relationship ASE service specifications

8.7.3.1 Supported services

This subclause contains the definition of service that are unique to this ASE. The services define for this ASE are

AR-Unconfirmed Send

AR-Get Buffered Message

AR-Status

8.7.3.2 AR-Unconfirmed send service

8.7.3.2.1 Service overview

This service is used to send AR-Unconfirmed request APDUs for FAL CM ASE. The AR-Unconfirmed Send service may be requested at the publisher endpoint of a one-to-many AR.

NOTE This service is described abstractly in such a way that it is capable of operating with ARs that convey FAL APDUs through buffers. This service may be implemented in such a way that the capability is provided to load the buffer, and subsequently post it for transfer by the underlying Data Link Layer. Alternatively, this service may be implemented such that these capabilities are combined so that the user may load the buffer and request its transfer in a single operation.

8.7.3.2.2 Service primitives

The service parameters for this service are shown in Table 68.

Table 68 – AR-Unconfirmed Send

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
AREP	M	M(=)		
FAL APDU Body	M	M(=)		

Argument

The argument contains the parameters of the service request.

FAL APDU Body

This parameter contains the service dependent body for the APDU.

8.7.3.2.3 Service procedure

The AR-Unconfirmed Send Service is a service that operates through a buffer.

The requesting FAL ASE submits an AR-Unconfirmed send request primitive to its AR ASE. The AR ASE builds an AR-Unconfirmed Send request APDU.

The AR ASE replaces the previous contents of the buffer with the APDU contained in the service primitive.

The AR ASE requests the DL to transfer the data at the scheduled time. The Data Link mapping indicates how the AR ASE coordinates its requests to transmit the data with the Data Link Layer.

NOTE The transmission schedule is managed by the underlying layer, not the AR-ASE. Refer to clause 6 and 7 for further details.

Upon receipt of the AR-Unconfirmed Send request APDU, the receiving AR ASE delivers an AR-Unconfirmed Send .indication primitive to the appropriate FAL ASE as indicated by the FAL Service Type Parameter.

8.7.3.3 AR-Gut Buffer message service

8.7.3.3.1 Service overview

This local service is used by application process to request the AR ASE to retrieve a message which is being maintained in a buffer in the local Data Link Layer.

This service does not result in the conveyance of an APDU. It is provided so that the FAL user may access a buffer through the FAL AR.

8.7.3.3.2 Service primitives

The service parameters for this service are shown in Table 69.

Table 69 – AR-Gut Buffered message service

Parameter name	Req	Ind	Rsp	Cnf
Argument	M			
AREP	M			
Result(+)				S
AREP				M
Decoded Buffer Data				M
Result(-)				S
AREP				M
Error Info				M

Argument

The argument contains the parameters of the service request.

Result (+)

This selection type parameter indicates that the service request succeeded.

Decoded Buffer Data

This parameter contains the user data in the FAL APDU read from the buffer.

Result (-)

This selection type parameter indicates that the service request failed.

8.7.3.3.3 Service procedure

This service requests the FAL to return the current contents of the buffer in a confirmation(+) primitive. If the buffer is empty, a confirmation(-) primitive is returned.

8.7.3.4 AR-Status service

8.7.3.4.1 Service overview

This local service provides the FAL user notification of a status change the AREP.

8.7.3.4.2 Service primitives

This service parameters for this service are shown in Table 70.

Table 70 – AR-Status service

Parameter name	Req	Ind	Rsp	Cnf
Argument		M		
AREP		M		
Status code		M		

Argument

This parameter carries the parameters of the service invocation.

Status Code

This specifies the status change being reported. The following status codes are defined

buffer received;

buffer update invocation;

lower layer reset.

8.7.3.4.3 Service procedure

This service indicates that a significant event, as defined by the status code parameter, occurred in the communication stack.

9 Application Layer Protocol specification

9.1 Introduction

This part of RTE-TCnet specification is one of a RTE (Real Time Ethernet) 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" RTE Reference Model, which is based in part on the Basic Reference Model for Open Systems Interconnection. Both Reference Models subdivide the area of standardization for interconnection into a series of layers of specification, each of manageable size.

The Application Protocol provides the Application Service by making use of the services available from the Data Link Layer or other immediately lower layer. This section defines the Application Protocol specification of RTE-TCnet. The relationship between the International Standards for Fieldbus Data Link service, Fieldbus Data Link protocol, Fieldbus Application service, Fieldbus Application protocol and System Management is illustrated in Figure 39 –. Relationship of RTE-TCnet Application layer to other RTE layers and to users of RTE Application service.

NOTE Systems Management, as used in this PAS, is a local mechanism for managing the layer protocols.

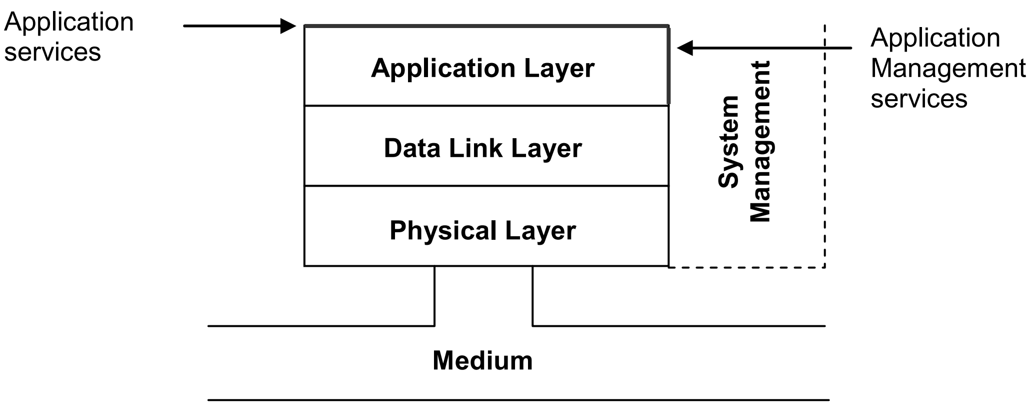


Figure 39 – Relationship of RTE-TCnet Application layer to other RTE layers and to users of RTE Application service

9.2 Scope

The RTE-TCnet Application Layer (FAL) is an Application Layer communication specification designed to support the conveyance of time-critical application requests and responses among devices in an automation environment. 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 PAS specifies interactions between remote applications in terms of

– the encoding rules that are applied to all the Application Layer Protocol Data Units (APDUs);
– the formal Abstract Syntax definitions of such APDUs;
– the protocol state machine descriptions that handle the APDUs and the primitives in the correct sequences;
– the mappings of the APDUs to and from the Data Link Layer services defined in IEC 61158-3.

The FAL encoding rules are designed assuming that both the encoder (sender) and the decoder (receiver) have the common knowledge of the abstract syntax. Wherever possible, data types identifiers are not encoded and transferred over the network.

NOTE This is why the Abstract Syntax Notation One / Basic Encoding Rule is not practical for the FAL.

The purpose of this subclause is to define the protocol provided

– to the Fieldbus Data Link Layer at the boundary between the Application and Data Link Layers of the Fieldbus Reference Model, and
– to the System Management at the boundary between the System Management and Application Layers of the Fieldbus Reference Model.

This PAS defines Application Layer protocols corresponding to the Application Layer service definitions specified in Clause8. They are identified in the IEC 61158 series of specifications as Type 1 to Type 15. The RTE-TCnet protocol is defined to Type 11 in this series of specifications.

9.3 Void

9.4 Term and definitions

9.4.1 Summary

For the purpose of this PAS, the following definitions apply:

9.4.2 Terms and definitions from other ISO/IEC standards

9.4.2.1 Term and definitions from ISO/IEC 7498-1

a) application entity;
b) application process;
c) application protocol data unit;
d) application service element;
e) transfer syntax.

9.4.2.2 Term and definitions from ISO/IEC 8822

a) abstract syntax.

9.4.2.3 Term and definitions from ISO/IEC 9545

a) application-association;
b) application-context;
c) application-service-element;
d) application control service element.

9.4.2.4 Term and definitions from ISO/IEC 8824-1

a) type;
b) value;
c) simple type;
d) structured type;
e) tag;
f) Boolean type;
g) true;
h) false;

i) integer type;
j) bitstring type;
k) octetstring type;
l) sequence type;
m) choice type;
n) any type.

9.4.2.5 Term and definitions from ISO/IEC 8825-1

a) encoding(of data value);
b) data value;
c) Identifier Octets;
d) Length Octet(s);
e) Contents Octets.

9.4.2.6 Term and definitions from IEC 61158-5

a) application relationship;
b) conveyance path;
c) dedicated AR;
d) pre-established AR endpoint;
e) publisher;
f) subscriber.

9.4.2.7 Term and definitions from this PAS (RTE-TCnet)

a) Pre-Established Connection

9.4.3 Symbols and abbreviations

9.4.3.1 Common note

NOTE Type 1 of IEC 61158-6, 3.5 abbreviations and symbols may apply to all the types except where the same abbreviation or symbols are used in different ways with other types. In this case the latter has precedence.

9.4.3.2 RTE-TCnet Abbreviations and symbols

CM Common Memory

9.4.4 Conventions

9.4.4.1 General concept

The FAL is defined as a set of object-oriented ASEs. Each ASE is specified in aseparate subclause. Each ASE specification is composed of three parts: its class definitions, its service, and its protocol specification. The first two are contain in clause 8. The protocol specification for each of the ASEs is defined in this PAS.

The class definition define the attributes of the classes supported by each ASE. The attributes are accessible from instances of the class using Management ASE services specified in IEC 61158-5 PAS. The service specification defines the services that are provided by the ASE.

This PAS uses the descriptive conventions given in ISO/IEC 10731.

9.4.4.2 Conventions for RTE-TCnet

None.

9.5 FAL Syntax description

9.5.1 General

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

9.5.2 FAL-AR PDU abstract syntax

9.5.2.1 Top level definition

```
FalArPDU::= UnconfirmedSend-CommandPDU
```

9.5.2.2 Unconfirmed send service

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

9.5.3 Abstract syntax of PDU Body

9.5.3.1 FalArHeader

```
FalArHeader::= Unsigned8 {
  -- bit 8        FAL Protocol Specifier    (Always 1 for the protocol defined by this Clause of RTE-TCnet)
  -- 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)
}
```

9.5.3.2 InvokeID

```
InvokeID::= Unsigned8
```

9.5.3.3 Unconfirmed PDUs

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

9.5.3.4 CMArep

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

9.5.3.5 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
              }
         }
}
```

9.5.4 Data Type

BitString8, BitString16, BitString32, Integer16, Integer32, Unsigned16, Unsigned32, Floating32, OctetString2, OctetString4, VisibleString2, VisibleString4, BinaryTime0, BinaryTime1, BinaryTime2, BinaryTime3, BinaryTime4, BinaryTime5, BinaryTime6, The data type of BinaryTime7 is same as described at IEC 61158-6, 4.1.5.

9.6 Transfer Syntax

9.6.1 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 71 defines how this header shall be used.

Table 71 – 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 The all other definitions are reserved.						

9.6.2 Encoding Rule

9.6.2.1 Overview

The Encoding Rule of RTE-TCnet is a preferable encoding rule that is compatible with existing standards. The FAL-PDUs encoded with the TER 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 40, traditional tag(or identifier),length and value(or contents) like TER(Traditional Encoding Rule) little endian format .

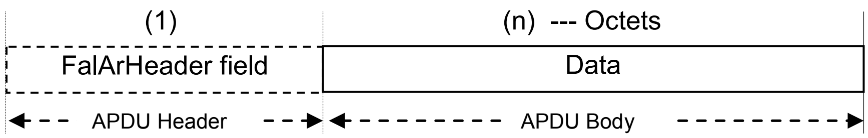


Figure 40 – APDU overview

9.6.2.2 APDU header encoding

The APDU Header part is always present in all APDUs which conform to this PAS. It consists of one field: the FalArHeader Field. Refer to 4.2.1 for the encoding rule of the FalArHeader field.

9.6.2.3 APDU body encoding

The Encoding Rule of FAL is based on the terms and definitions of the ISO/IEC standards, and consists in order to the three encoding components as below. For time-critical and using fixed length data, Identifier Octet and Length Octets do not exist.

- Identification Octet
- Content Length Octets
- Content Octets

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

9.6.2.4 Encoding of simple variable

9.6.2.4.1 Encoding of a Boolen value

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

b) The Identifier Octet and Length Octet(s) shall not be present. The ContentsOctets shall consist of a single octet.

c) If the Boolean value is FALSE, the ContentsOctets shall be 0 (zero). IF the Boolean value is TRUE, the ContentsOctets shall be 0xFF.

9.6.2.4.2 Encoding of a fixed length Integer value

a) The encoding of a fixed-length Integer value of Integer8, Integer16 and Integer32 types shell be primitive, and the ContentsOctets shall consist of exactly one, two or four octets, respectively.

b) The Identifier Octet and Length Octet(s) shall not be present.

c) The ContentsOctets 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 ContentsOctets.

NOTE The value of a two's complement binary number is derived by numbering the bits in the ContentOctets, starting with bit 1 of the first octet and ending the numbering with bit 8 of the last octet. Each bit is assigned a numerical value of 2^{N-1}, where N is its position in the above numbering sequence. The value of the two's complement binary number is obtaind 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.

9.6.2.5 Encoding of a fixed length Unsigned value

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

b) The Identifier Octet and Length Octet(s) shall not be present.

c) The ContentsOctets 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 ContentsOctets.

NOTE The value of binary number is derived by numbering the bits in the ContentOctets, starting with bit 1 of the first octet as bit zero and ending the numbering with bit 8 of the last octet. Each bit is assigned a numerical value of 2^{N-1}, where N is its position in the above numbering sequence. The value of the binary number is obtaind by adding the numerical values assigned to each bit for those bits which are set to one.

9.6.2.6 Encoding of a Floating Point value

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

b) The Identifier Octet and Length Octet(s) shall not be present.

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 ContentsOctets shall contain floating point values defined in conformance with IEC 60559.

9.6.2.7 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, respoctively.

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 secondoctet, followed by bits 8 to 1 of each octet up to and including the last octet of theContents Octets.

9.6.2.8 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.

9.6.2.9 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.

9.6.2.10 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.

9.6.3 Encoding of structured types

9.6.3.1 General

When structured type is alsp encoeded, The identifier or length of the structure are not provided in RTE-TCnet.

9.6.3.2 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.

9.7 FAL protocol state machines structures

9.7.1 Overview

As shown in Figure 41, the protocol machine of FAL consists of three, FAL service protocol machine (FSPM), application relationship protocol machine (ARPM) and data link layer mapping protocol machine (DMPM).

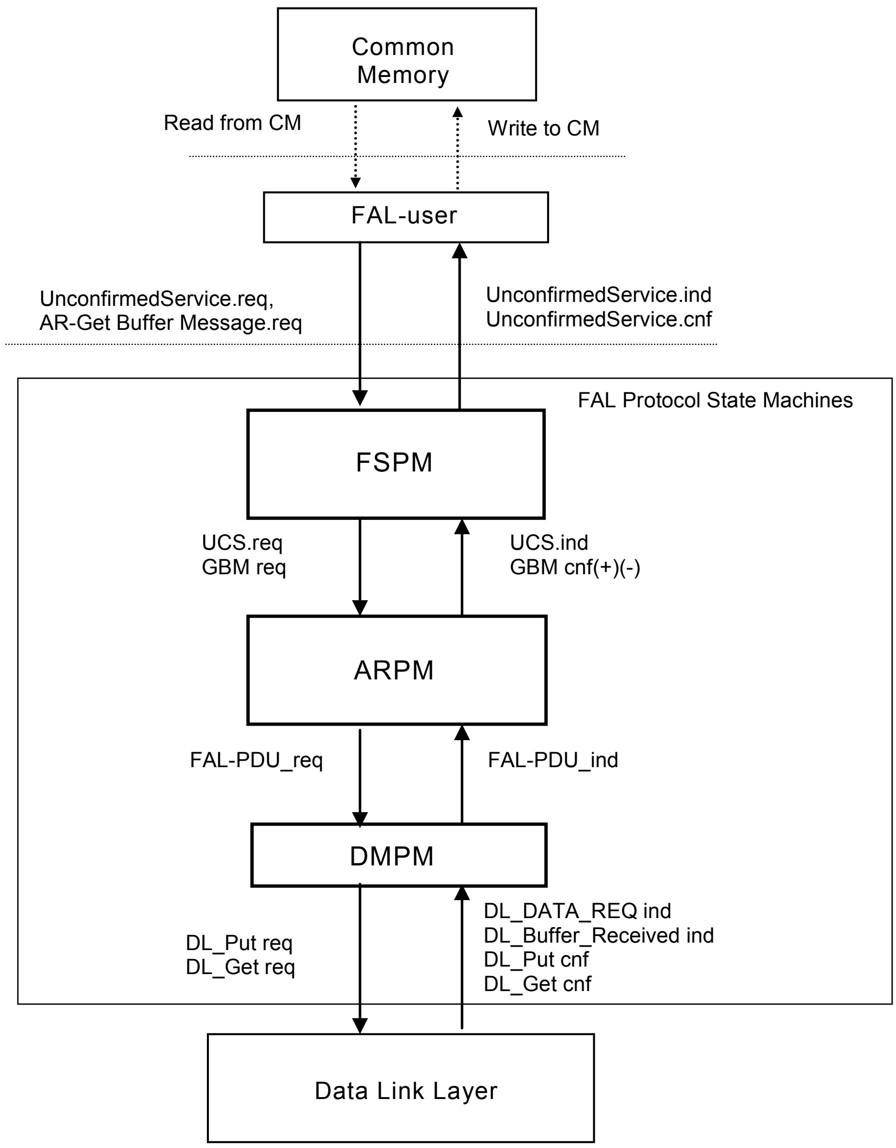


Figure 41 – Relationship between FSPM, ARPM, DMPM and External Physical CM

9.8 FAL service protocol machine (FSPM)

9.8.1 General

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.

9.8.2 Primitives definitions

9.8.2.1 Primitive exchanged between FAL user and FSPM

The primitive exchanged between the FAL user and the FSPM are described in Table 72 and Table 73.

Table 72 – 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 73 – 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 .

9.8.2.2 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.

9.8.3 FSPM state tables

9.8.3.1 General

The FSPM state machines are described in Figure 42, Table 74 and Table 75 .

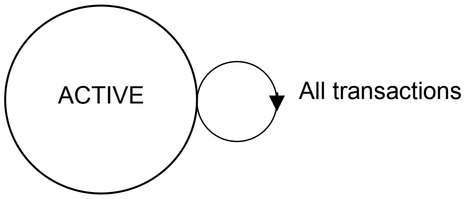


Figure 42 – State transition diagram of FSPM

Table 74 – FSPM state table – sender transactions

No.	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 75 – FSPM state table – receiver transactions

No.	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

9.8.3.2 Functions

The function used in this state machine is as shown in Table 76.

Table 76 – 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 exist.			

9.9 Application relationship protocol machine (ARPM)

9.9.1 General

The RTE-TCnet define a Application Relation (AR) and their associated ARPM, which is Buffered Network-scheduled Unidirectional – Pre-established Connection (BNU-PEC).

9.9.2 Primitive definitions

9.9.2.1 Primitives exchanged between ARPM and FSPM

Table 77 and Table 78 list the primitives exchanged between the FSPM and the ARPM.

Table 77 – 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 78 – 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, user_data	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.

9.9.2.2 Parameters of FSPM/ARPM primitives

The parameters used with the primitives exchanged between the FSPM and the ARPM are described in Table 79 .

Table 79 – 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 PAS.
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 Reson_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.

9.9.3 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

9.9.3.1 Attributes

9.9.3.1.1 LocalDlcepIdentifier

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

9.9.3.1.2 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.

9.9.3.2 DLL services

Refer to 9.10.4, Data Link Layer Service Selection, for DLL service descriptions.

9.9.4 BNU-PEC ARPM states machine

9.9.4.1 BNU-PEC ARPM state

The defined states together with their descriptions of the BNU-PEC ARPM are listed in Table 80 and Figure 43.

Table 80 – 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.

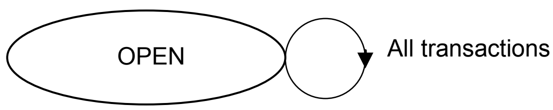


Figure 43 – State transition diagram of the BNU-PEC

9.9.4.2 BNU-PEC ARPM state table

Table 81 and Table 82 define the BNU-PEC state machines.

Table 81 – BNU-PEC ARPM state table – sender transactions

No.	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 82 – BNU-PEC ARPM state table – receiver transactions

No.	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

9.9.4.3 Functions used by BNU-PEC ARPM

Table 83 through Table 84 define the function used by this service machine.

Table 83 – Function GetArepId ()

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

Table 84 – Function BuildFAL-PDU

Name	BuildFAL-PDU	Used in	ARPM
input		Output	
fal_pdu_name, fal_data		dlsdu	
Function			
Build an FAL-PDU out of the parameters given as input variables.			

Table 85 – 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			
This function decodes the FAL-PDU that is conveyed in the fal_pdu parameter and retrieves one of the FAL-PDU types.			

9.10 DLL Mapping Protocol Machine (DMPM)

9.10.1 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.

A 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 RTE-TCnet Data Link Layer, DLSDU-length is defined to partition the end of DLSDU. In actual implementation, usage of this parameter is depended on implementation. In this DMPM, DLSDU-length is not included.

A remarks about configuration and initialization of the Data Link Layer:

The RTE-TCnet Data Link Layer specification defines configuration service to set the resource of layer or the class of connection.

In the specification of RTE-TCnet Data Link Layer, the configuration service is defined to set the class of resource and connection.

9.10.2 Primitive definitions

9.10.2.1 Primitives exchanged between DMPM and ARPM

Table 86 and Table 87 list the service primitives between the ARPM and the DMPM.

Table 86 – 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 87 – 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.

9.10.2.2 Parameters of ARPM/DMPM primitives

The parameters used with the primitives exchanged between the ARPM and the DMPM are described Table 88.

Table 88 – 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.

9.10.2.3 Primitives exchanges between Data Link Layer and DMPM

Table 89 summarizes the primitives exchanged between the DLL and the DMPM.

Table 89 – 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

9.10.2.4 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 the Data Link Layer Service definition (Clause 6). They are prefixed by "dl_" to indicate that they are used by the FAL.

9.10.3 DLL Mapping Protocol Machine (DMPM)

9.10.3.1 DMPM States

The defined state of the DMPM together with its description are listed in Table 90 and Figure 44 .

Table 90 – 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.

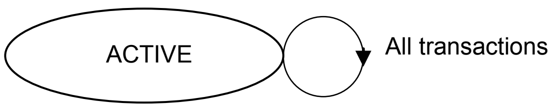


Figure 44 – State transition diagram of DMPM

9.10.3.2 DMPM state table

The DMPM state machines are defined in Table 91 and Table 91.

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

Table 91 – DMPM state table – sender transactions

No.	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 92 – DMPM state table – receiver transactions

No.	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

9.10.3.3 Function used by DMPM

Table 93 and Table 93 define the function used by DMPM.

Table 93 – 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 94 – Function FindAREP

Name	FindAREP	Used in	DMPM
input		Output	
local_dlcep_identifier		True || False	
Function	This 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.		

9.10.4 Data Link Layer service selection

9.10.4.1 General

This subclause briefly describes the Data Link Layer service utilized by the FAL. These Data Link Layer service are fully defined in the Data Link Layer specification (Clause 6).

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

9.10.4.2 DL_Put

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

9.10.4.3 DL_Get

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

9.10.4.4 DL_Buffer_Received

The DL_Buffer_Received service is used to inform the FAL of new update on the specified receive buffer.

9.10.4.5 DL_DATA_REQ

The DL_DATA_REQ service is used to inform the FAL that the specified buffer content became update timing.

Bibliography

IEC 60559:1989, *Binary floating-point arithmetic for microprocessor systems*

IEC 61158-2:2003, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 2: Physical layer specification and service definition*

ISO/IEC 646:1991, *Information technology – ISO 7–bit coded character set for information interchange*

ISO/IEC 7498 (all parts), *Information Technology – Open Systems Interconnnection – Basic Reference Model*

ISO/IEC 8859-1:1998, *Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1*

ISO/IEC 10646-1:1993, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Architecture and Basic Multilingual Plane*

IEEE Std 754-1985 (R1990), *IEEE Standard for Binary Floating Point Arithmetic*

Standards Survey

The IEC would like to offer you the best quality standards possible. To make sure that we continue to meet your needs, your feedback is essential. Would you please take a minute to answer the questions overleaf and fax them to us at +41 22 919 03 00 or mail them to the address below. Thank you!

Customer Service Centre (CSC)

International Electrotechnical Commission
3, rue de Varembé
1211 Genève 20
Switzerland

or

Fax to: **IEC**/CSC at +41 22 919 03 00

Thank you for your contribution to the standards-making process.

Nicht frankieren
Ne pas affranchir

Non affrancare
No stamp required

A Prioritaire

RÉPONSE PAYÉE

SUISSE

Customer Service Centre (CSC)
International Electrotechnical Commission
3, rue de Varembé
1211 GENEVA 20
Switzerland

Q1 Please report on **ONE STANDARD** and **ONE STANDARD ONLY**. Enter the exact number of the standard: *(e.g. 60601-1-1)*

....

Q2 Please tell us in what capacity(ies) you bought the standard *(tick all that apply).* I am the/a:

- purchasing agent ❑
- librarian ❑
- researcher ❑
- design engineer ❑
- safety engineer ❑
- testing engineer ❑
- marketing specialist ❑
- other...

Q3 I work for/in/as a: *(tick all that apply)*

- manufacturing ❑
- consultant ❑
- government ❑
- test/certification facility ❑
- public utility ❑
- education ❑
- military ❑
- other...

Q4 This standard will be used for: *(tick all that apply)*

- general reference ❑
- product research ❑
- product design/development ❑
- specifications ❑
- tenders ❑
- quality assessment ❑
- certification ❑
- technical documentation ❑
- thesis ❑
- manufacturing ❑
- other...

Q5 This standard meets my needs: *(tick one)*

- not at all ❑
- nearly ❑
- fairly well ❑
- exactly ❑

Q6 If you ticked NOT AT ALL in Question 5 the reason is: *(tick all that apply)*

- standard is out of date ❑
- standard is incomplete ❑
- standard is too academic ❑
- standard is too superficial ❑
- title is misleading ❑
- I made the wrong choice ❑
- other ....

Q7 Please assess the standard in the following categories, using the numbers:
(1) unacceptable,
(2) below average,
(3) average,
(4) above average,
(5) exceptional,
(6) not applicable

- timeliness ....
- quality of writing....................................
- technical contents.................................
- logic of arrangement of contents ....
- tables, charts, graphs, figures...............
- other ....

Q8 I read/use the: *(tick one)*

- French text only ❑
- English text only ❑
- both English and French texts ❑

Q9 Please share any comment on any aspect of the IEC that you would like us to know:

....
....
....
....
....
....
....
....
....
....
....
....

ISBN 2-8318-8153-6

9 782831 881539

ICS 25.040.40; 35.240.50; 35.100.05

Typeset and printed by the IEC Central Office
GENEVA, SWITZERLAND