

# INTERNATIONAL STANDARD

## NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –  
Part 5-28: Application layer service definition – Type 28 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –  
Partie 5-28: Définition des services de la couche application –  
Éléments de type 28**

IECNORM.COM : Click to view the full PDF of IEC 61158-5-28:2023



## THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2023 IEC, Geneva, Switzerland

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

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

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

Tel.: +41 22 919 02 11  
[info@iec.ch](mailto:info@iec.ch)  
[www.iec.ch](http://www.iec.ch)

### About the IEC

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

### About IEC publications

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

#### IEC publications search - [webstore.iec.ch/advsearchform](http://webstore.iec.ch/advsearchform)

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

#### IEC Just Published - [webstore.iec.ch/justpublished](http://webstore.iec.ch/justpublished)

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

#### IEC Customer Service Centre - [webstore.iec.ch/csc](http://webstore.iec.ch/csc)

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: [sales@iec.ch](mailto:sales@iec.ch).

#### IEC Products & Services Portal - [products.iec.ch](http://products.iec.ch)

Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

#### Electropedia - [www.electropedia.org](http://www.electropedia.org)

The world's leading online dictionary on electrotechnology, containing more than 22 300 terminological entries in English and French, with equivalent terms in 19 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

### A propos de l'IEC

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

### A propos des publications IEC

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

#### Recherche de publications IEC -

[webstore.iec.ch/advsearchform](http://webstore.iec.ch/advsearchform)

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

#### IEC Just Published - [webstore.iec.ch/justpublished](http://webstore.iec.ch/justpublished)

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

#### Service Clients - [webstore.iec.ch/csc](http://webstore.iec.ch/csc)

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: [sales@iec.ch](mailto:sales@iec.ch).

IEC Products & Services Portal - [products.iec.ch](http://products.iec.ch)

Découvrez notre puissant moteur de recherche et consultez gratuitement tous les aperçus des publications. Avec un abonnement, vous aurez toujours accès à un contenu à jour adapté à vos besoins.

#### Electropedia - [www.electropedia.org](http://www.electropedia.org)

Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 300 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 19 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

# INTERNATIONAL STANDARD

## NORME INTERNATIONALE

**Industrial communication networks –Fieldbus specifications –  
Part 5-28: Application layer service definition – Type 28 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –  
Partie 5-28: Définition des services de la couche application –  
Éléments de type 28**

INTERNATIONAL  
ELECTROTECHNICAL  
COMMISSION

COMMISSION  
ELECTROTECHNIQUE  
INTERNATIONALE

ICS 25.040

ISBN 978-2-8322-6583-3

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

## CONTENTS

|                                                                        |    |
|------------------------------------------------------------------------|----|
| FOREWORD.....                                                          | 5  |
| INTRODUCTION.....                                                      | 7  |
| 1 Scope.....                                                           | 8  |
| 1.1 Overview.....                                                      | 8  |
| 1.2 Specifications .....                                               | 9  |
| 1.3 Conformance .....                                                  | 9  |
| 2 Normative references .....                                           | 9  |
| 3 Terms, definitions, symbols, abbreviated terms and conventions ..... | 10 |
| 3.1 Referenced terms and definitions.....                              | 10 |
| 3.1.1 ISO/IEC 7498-1 terms.....                                        | 10 |
| 3.1.2 ISO/IEC 8822 terms.....                                          | 10 |
| 3.1.3 ISO/IEC 9545 terms.....                                          | 11 |
| 3.1.4 ISO/IEC 8824 terms.....                                          | 11 |
| 3.2 Additional terms and definitions for this IEC 61158-5-28 .....     | 11 |
| 3.3 Abbreviations and symbols .....                                    | 12 |
| 3.4 Conventions.....                                                   | 13 |
| 3.4.1 Overview .....                                                   | 13 |
| 3.4.2 Conventions for class definitions .....                          | 13 |
| 3.4.3 Conventions for service definitions .....                        | 14 |
| 4 Concepts .....                                                       | 15 |
| 5 Data type ASE .....                                                  | 15 |
| 5.1 General.....                                                       | 15 |
| 5.1.1 Overview .....                                                   | 15 |
| 5.1.2 Overview of basic types.....                                     | 16 |
| 5.1.3 Overview of fixed-length types.....                              | 16 |
| 5.1.4 Overview of constructed types.....                               | 17 |
| 5.2 AL defined data types .....                                        | 17 |
| 5.2.1 Fixed length types .....                                         | 17 |
| 5.2.2 String type.....                                                 | 19 |
| 5.2.3 Structure type.....                                              | 20 |
| 6 Type 28 communication model specification .....                      | 25 |
| 6.1 General.....                                                       | 25 |
| 6.2 Protocol stack for Type 28 fieldbus .....                          | 26 |
| 6.3 Data mapping modelling.....                                        | 27 |
| 6.4 Overview of Type 28 communication model .....                      | 29 |
| 6.4.1 General .....                                                    | 29 |
| 6.4.2 P/S .....                                                        | 29 |
| 6.4.3 C/S .....                                                        | 30 |
| 6.5 AL related management information table .....                      | 30 |
| 6.5.1 General .....                                                    | 30 |
| 6.5.2 Time management information table .....                          | 31 |
| 6.5.3 Network management information table .....                       | 32 |
| 6.6 ASEs .....                                                         | 33 |
| 6.6.1 Overview .....                                                   | 33 |
| 6.6.2 RT data ASE.....                                                 | 34 |
| 6.6.3 nRT data ASE.....                                                | 36 |

|                       |                                                                           |    |
|-----------------------|---------------------------------------------------------------------------|----|
| 6.6.4                 | Time ASE .....                                                            | 38 |
| 6.6.5                 | Resource ASE .....                                                        | 41 |
| 6.6.6                 | Addressing ASE .....                                                      | 43 |
| 6.6.7                 | Management ASE .....                                                      | 48 |
| 6.6.8                 | Virtualization ASE .....                                                  | 55 |
| 6.7                   | Application service .....                                                 | 58 |
| 6.7.1                 | Overview .....                                                            | 58 |
| 6.7.2                 | Read service .....                                                        | 58 |
| 6.7.3                 | Write service .....                                                       | 58 |
| Annex A (informative) | Example of service data mapping DTU message .....                         | 60 |
| Annex B (informative) | Example of OPC UA data model and Type 28 data model mapping .....         | 61 |
| Annex C (informative) | Example of RTA service processing .....                                   | 62 |
| Annex D (informative) | Type 28 virtualization solution example .....                             | 63 |
| Bibliography          | .....                                                                     | 66 |
| Figure 1              | Data type class hierarchy example .....                                   | 16 |
| Figure 2              | BITMAP data type .....                                                    | 20 |
| Figure 3              | AL in fieldbus protocol stack architecture .....                          | 26 |
| Figure 4              | Type 28 AL protocol architecture .....                                    | 26 |
| Figure 5              | Data mapping modelling object structure .....                             | 27 |
| Figure 6              | Diagram of mapping data buffers and service data variables .....          | 28 |
| Figure 7              | Diagram of P/S model .....                                                | 29 |
| Figure 8              | P/S model of PUSH mode .....                                              | 30 |
| Figure 9              | P/S model of PULL mode .....                                              | 30 |
| Figure 10             | Diagram of C/S communication model .....                                  | 30 |
| Figure 11             | RT data transmission model .....                                          | 35 |
| Figure 12             | Diagram of nRT data request response model .....                          | 37 |
| Figure 13             | Diagram of nRT data AP interaction based on C/S communication model ..... | 38 |
| Figure 14             | Diagram of nRT data AP interaction based on P/S communication model ..... | 38 |
| Figure 15             | Time synchronization application interaction process .....                | 40 |
| Figure 16             | Time query process .....                                                  | 41 |
| Figure 17             | Resource AP interaction based on C/S communication model .....            | 43 |
| Figure 18             | Resource ASE local service function diagram .....                         | 43 |
| Figure 19             | NETWORKID identification field structure .....                            | 44 |
| Figure 20             | NETWORKID interaction process diagram .....                               | 46 |
| Figure 21             | Diagram of MAC mapping table service function .....                       | 46 |
| Figure 22             | IP mapping table service function diagram .....                           | 47 |
| Figure 23             | Type 28 AL data message mapped to IP payload data payload .....           | 47 |
| Figure 24             | Type 28 AL protocol header and IP protocol header mapping .....           | 48 |
| Figure 25             | IP data message is mapped to valid data of Type 28 AL data message .....  | 48 |
| Figure 26             | Static configuration initialization process .....                         | 51 |
| Figure 27             | Join the network interaction process .....                                | 52 |
| Figure 28             | Passively exit the network interaction process .....                      | 53 |

|                                                                                                |    |
|------------------------------------------------------------------------------------------------|----|
| Figure 29 – Actively exit network interaction process .....                                    | 54 |
| Figure 30 – Diagram of diagnostic object mapping .....                                         | 54 |
| Figure 31 – Diagram of the logging process .....                                               | 55 |
| Figure 32 – A physical bus divided into two virtual bus domain diagrams .....                  | 56 |
| Figure 33 – Diagram of virtualized AP interaction based on C/S communication model .....       | 57 |
| Figure 34 – Virtualization ASE local implementation diagram .....                              | 58 |
| Figure A.1 – Data transmission unit message content example .....                              | 60 |
| Figure B.1 – OPC UA data model and mapping of Type 28 data model .....                         | 61 |
| Figure C.1 – Example of RTA service processing .....                                           | 62 |
| Figure D.1 – Diagram of Type 28 industrial bus service deployment .....                        | 63 |
| Figure D.2 – Diagram of networking topology based on the virtualized bus .....                 | 63 |
| Figure D.3 – Virtual network topology based on logical business function RT1 .....             | 64 |
| Table 1 – TIMEV type data coding .....                                                         | 17 |
| Table 2 – TIMEDATE type coding .....                                                           | 18 |
| Table 3 – TIMEOFDAY type coding .....                                                          | 18 |
| Table 4 – TIMEDIFFER type coding .....                                                         | 19 |
| Table 5 – VISIBLESTRING type data coding .....                                                 | 19 |
| Table 6 – Description of time information structure TIMEINFO_S .....                           | 20 |
| Table 7 – Description of clock option information structure CLOCK_OPTION_INFO_S .....          | 21 |
| Table 8 – Description of network configuration parameter structure<br>NETWORK_CFG_PARA_S ..... | 22 |
| Table 9 – Description of MAC mapping table NETWORKID_MAC_MAP_TABLE_S .....                     | 23 |
| Table 10 – Description of IP mapping table structure IP_MAP_TABLE_S .....                      | 24 |
| Table 11 – Service data mapping table parameter list .....                                     | 29 |
| Table 12 – Time management information table .....                                             | 31 |
| Table 13 – Network management information table .....                                          | 32 |
| Table 14 – Communication models supported by Type 28 AL ASEs .....                             | 33 |
| Table 15 – Application service interface return value list .....                               | 33 |
| Table 16 – RT data service interface parameters .....                                          | 34 |
| Table 17 – nRT data service interface parameters .....                                         | 36 |
| Table 18 – Timing service interface parameters .....                                           | 39 |
| Table 19 – Resource service interface parameters .....                                         | 42 |
| Table 20 – Addressing service interface parameters .....                                       | 45 |
| Table 21 – Management service interface parameters .....                                       | 49 |
| Table 22 – Virtual service interface parameters .....                                          | 56 |
| Table 23 – Read service parameters .....                                                       | 58 |
| Table 24 – Write service parameters .....                                                      | 59 |
| Table 25 – Application service errcode .....                                                   | 59 |
| Table D.1 – Virtual bus environment node NETWORKID list .....                                  | 65 |

## INTERNATIONAL ELECTROTECHNICAL COMMISSION

**INDUSTRIAL COMMUNICATION NETWORKS –  
FIELDBUS SPECIFICATIONS –****Part 5-28: Application layer service definition –  
Type 28 elements****FOREWORD**

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

Attention is drawn to the fact that the use of the associated protocol type is restricted by its intellectual-property-right holders. In all cases, the commitment to limited release of intellectual-property-rights made by the holders of those rights permits a layer protocol type to be used with other layer protocols of the same type, or in other type combinations explicitly authorized by its intellectual-property-right holders.

NOTE Combinations of protocol types are specified in the IEC 61784-1 series and the IEC 61784-2 series.

IEC 61158-5-28 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation. It is an International Standard.

The text of this International Standard is based on the following documents:

| Draft         | Report on voting |
|---------------|------------------|
| 65C/1206/FDIS | 65C/1235/RVD     |

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

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement available at [www.iec.ch/members\\_experts/refdocs](http://www.iec.ch/members_experts/refdocs). The main document types developed by IEC are described in greater detail at [www.iec.ch/publications](http://www.iec.ch/publications).

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

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under [webstore.iec.ch](http://webstore.iec.ch) in the data related to the specific document. At this date, the document will be

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

IECNORM.COM : Click to view the full PDF of IEC 61158-5-28:2023

## INTRODUCTION

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

The application service is provided by the application protocol making use of the services available from the data-link or other immediately lower layer. This document defines the application service characteristics that fieldbus applications and/or system management can exploit.

Throughout the set of fieldbus standards, the term "service" refers to the abstract capability provided by one layer of the OSI Basic Reference Model to the layer immediately above. Thus, the application layer service defined in this document is a conceptual architectural service, independent of administrative and implementation divisions.

IECNORM.COM : Click to view the full PDF of IEC 61158-5-28:2023

## **INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –**

### **Part 5-28: Application layer service definition – Type 28 elements**

#### **1 Scope**

##### **1.1 Overview**

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

This part of IEC 61158 provides common elements for basic time-critical and non-time-critical messaging communications between application programs in an automation environment and material specific to Type 28 fieldbus. The term "time-critical" is used to represent the presence of a time-window, in which one or more specified actions are required to be completed with some defined level of certainty.

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

- an abstract model for defining application resources (objects) capable of being manipulated by users via the use of the FAL service,
- the primitive actions and events of the service,
- the parameters associated with each primitive action and event, and the form which they take, and
- the interrelationship between these actions and events, and their valid sequences.

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

- the FAL user at the boundary between the user and the Application Layer of the Fieldbus Reference Model, and
- Systems Management at the boundary between the Application Layer and Systems Management of the Fieldbus Reference Model.

This document specifies the structure and services of the IEC fieldbus Application Layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498-1) and the OSI Application Layer Structure (ISO/IEC 9545).

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

Although these services specify, from the perspective of applications, how to request and response 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 behavioural aspects of the applications are not specified; only a definition of what requests and responses they can send/receive is specified. This permits greater flexibility to the FAL users in standardizing such

object behaviour. In addition to these services, some supporting services are also defined in this document to provide access to the FAL to control certain aspects of its operation.

## 1.2 Specifications

The principal objective of this document is to specify the characteristics of conceptual application layer services suitable for time-critical communications, and thus supplement the OSI Basic Reference Model in guiding the development of application layer protocols for time-critical communications.

A secondary objective is to provide migration paths from previously existing industrial communications protocols. It is this latter objective which gives rise to the diversity of services standardized as the various types of IEC 61158, and the corresponding protocols standardized in the IEC 61158-6 series.

This document can be used as the basis for formal application programming-interfaces. Nevertheless, it is not a formal programming interface, and any such interface will address implementation issues not covered by this document, including

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

## 1.3 Conformance

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

There is no conformance of equipment to this application layer service definition standard. Instead, conformance is achieved through implementation of conforming application layer protocols that fulfils the Type 28 application layer services as defined in this document.

## 2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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.

NOTE All parts of the IEC 61158 series, as well as the IEC 61784-1 series and the IEC 61784-2 series are maintained simultaneously. Cross -references to these documents within the text therefore refer to the editions as dated in this list of normative references.

IEC 61158-1:2023, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61158-4-28:2023, *Industrial communication networks – Fieldbus specifications – Part 4-28: Data-link layer protocol specification – Type 28 elements*

IEC 61158-6-28:2023, *Industrial communication networks – Fieldbus specifications – Part 6-28: Application layer protocol specification – Type 28 elements*

ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*

ISO/IEC 7498-3:1997, *Information technology – Open Systems Interconnection – Basic Reference Model: Naming and addressing*

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

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

ISO/IEC 9545, *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/IEC/IEEE 8802-3:2021, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Standard for Ethernet*

IETF RFC 791, J. Postel, *Internet Protocol*, September 1981, available at <https://www.rfc-editor.org/info/rfc791> [viewed 2022-02-18]

IETF RFC 2460, S. Deering and R. Hinden, *“Internet Protocol, Version 6 (IPv6) Specification”*, December 1998, available at <https://www.rfc-editor.org/info/rfc2460> [viewed 2022-02-18]

### **3 Terms, definitions, symbols, abbreviated terms and conventions**

For the purposes of this document, the following terms, definitions, symbols, abbreviated terms and conventions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <https://www.electropedia.org/>
- ISO Online browsing platform: available at <https://www.iso.org/obp>

#### **3.1 Referenced terms and definitions**

##### **3.1.1 ISO/IEC 7498-1 terms**

For the purposes of this document, the following terms given in ISO/IEC 7498-1 apply:

- application entity
- application process
- application protocol data unit
- application service element
- application entity invocation
- application process invocation
- application transaction
- real open system
- transfer syntax

##### **3.1.2 ISO/IEC 8822 terms**

For the purposes of this document, the following terms given in ISO/IEC 8822 apply:

- abstract syntax

- presentation context

### 3.1.3 ISO/IEC 9545 terms

For the purposes of this document, the following terms given in ISO/IEC 9545 apply:

- application-entity-invocation
- application-entity-type
- application-process-invocation
- application-process-type
- application-service-element

### 3.1.4 ISO/IEC 8824 terms

For the purposes of this document, the following terms given in ISO/IEC 8824 apply:

- object identifier
- type

## 3.2 Additional terms and definitions for this IEC 61158-5-28

The following terms and definitions apply to this document.

### 3.2.1

#### **clock query**

get clock information from the management node or terminal node device

### 3.2.2

#### **clock synchronization**

clock calibration of the terminal node device

### 3.2.3

#### **control device**

device that controls all field devices for logical operations, timing, calculations, etc.

### 3.2.4

#### **cyclic**

events which repeat in a regular and repetitive manner

### 3.2.5

#### **cyclic time**

time of cyclic processing of a device or module

### 3.2.6

#### **field device**

physical entity that is connected to the process or to factory equipment and has at least one signalling element that communicates with other signalling element(s) via a cable

### 3.2.7

#### **management node**

device for allocating and managing Type 28 network physical communication resources

### 3.2.8

#### **non-RT Data**

data insensitive to time deterministic requirements

### 3.2.9

#### RT Data

data sensitive to time deterministic requirements

### 3.2.10

#### terminal node

device in Type 28 network that communicates based on allocated physical communication resources

## 3.3 Abbreviations and symbols

|                 |                                     |
|-----------------|-------------------------------------|
| <b>AL</b>       | Application layer                   |
| <b>ALCE</b>     | Application layer clock entity      |
| <b>ALDE</b>     | Application layer data entity       |
| <b>ALME</b>     | Application layer management entity |
| <b>ALS</b>      | Application layer service           |
| <b>AE</b>       | Application entity                  |
| <b>AP</b>       | Application process                 |
| <b>APO</b>      | AP object                           |
| <b>AR</b>       | Application relationship            |
| <b>ASE</b>      | Application service elements        |
| <b>C/S</b>      | Client/Server                       |
| <b>DL-</b>      | Data-link layer (as a prefix)       |
| <b>DLCE</b>     | Data-link clock entity              |
| <b>DLDE</b>     | Data-link data entity               |
| <b>DLE</b>      | Data-link entity                    |
| <b>DLL</b>      | Data-link layer                     |
| <b>DLS</b>      | Data-link service                   |
| <b>DLP</b>      | Data-link protocol                  |
| <b>DLS-user</b> | Data-link service user              |
| <b>DTC</b>      | Data transmission channel           |
| <b>DTS</b>      | Data transmission service           |
| <b>DTU</b>      | Data transmission unit              |
| <b>DLSDU</b>    | Data-link service data unit         |
| <b>DLPDU</b>    | Data-link protocol data unit        |
| <b>MAC</b>      | Medium access control               |
| <b>MN</b>       | Management node                     |
| <b>NodeID</b>   | Node identifier                     |
| <b>nRT</b>      | non-Real-Time                       |
| <b>PhL</b>      | Physical layer                      |
| <b>RT</b>       | Real-Time                           |
| <b>RTA</b>      | Real-Time acyclic                   |
| <b>RTC</b>      | Real-Time cyclic                    |
| <b>TN</b>       | Terminal node                       |

### 3.4 Conventions

#### 3.4.1 Overview

The FAL is defined as a set of object-oriented ASEs. Each ASE is specified in a separate subclause. Each ASE specification is composed of two parts, including its class specification and its service specification.

The class specification defines the attributes of the class. The service specification defines the services that are provided by the ASE.

#### 3.4.2 Conventions for class definitions

Class definitions are described using templates. Each template consists of a list of attributes for the class. The general form of the template is shown below:

| FAL ASE:      |                   |                | ASE Name               |
|---------------|-------------------|----------------|------------------------|
| CLASS:        | Class Name        |                |                        |
| CLASS ID      | #                 |                |                        |
| PARENT CLASS: | Parent Class Name |                |                        |
| ATTRIBUTES:   |                   |                |                        |
| 1             | (o)               | Key Attribute: | numeric identifier     |
| 2             | (o)               | Key Attribute: | Name                   |
| 3             | (m)               | Attribute:     | attribute name(values) |
| 4             | (m)               | Attribute:     | attribute name(values) |
| 4.1           | (s)               | Attribute:     | attribute name(values) |
| 4.2           | (s)               | Attribute:     | attribute name(values) |
| 4.3           | (s)               | Attribute:     | attribute name(values) |
| 5             | (c)               | Constraint:    | constraint expression  |
| 5.1           | (m)               | Attribute:     | attribute name(values) |
| 5.2           | (o)               | Attribute:     | attribute name(values) |
| 6             | (m)               | Attribute:     | attribute name(values) |
| 6.1           | (s)               | Attribute:     | attribute name(values) |
| 6.2           | (s)               | Attribute:     | attribute name(values) |
| SERVICES:     |                   |                |                        |
| 1             | (o)               | OpsService     | service name           |
| 2             | (c)               | Constraint:    | constraint expression  |
| 3.1           | (o)               | OpsService     | service name           |
| 3             | (m)               | MgtService:    | service name           |

- (1) The "FAL ASE:" entry is the name of the FAL ASE that provides the services for the class being specified.
- (2) The "CLASS:" entry is the name of the class being specified. All objects defined using this template will be an instance of this class. The class may be specified by this document, or by a user of this document.
- (3) The "CLASS ID:" entry is a number that identifies the class being specified. This number is unique within the FAL ASE that will provide the services for this class. When qualified by the identity of its FAL ASE, it unambiguously identifies the class within the scope of the FAL. The value "NULL" indicates that the class cannot be instantiated. Class IDs between 1 and 99, 240 and 767 are reserved by this document to identify standardized classes. CLASS IDs between 100 and 199, 768 and 1 279 are allocated for identifying user defined classes.

- (4) The "PARENT CLASS:" entry is the name of the parent class for the class being specified. All attributes defined for the parent class and inherited by it are inherited for the class being defined, and therefore do not have to be redefined in the template for this class.

NOTE The parent-class "TOP" indicates that the class being defined is an initial class definition. The parent class TOP is used as a starting point from which all other classes are defined. The use of TOP is reserved for classes defined by this document.

- (5) The "ATTRIBUTES" label indicate that the following entries are attributes defined for the class.
- a) Each of the attribute entries contains a line number in column 1; a mandatory (m) / optional (o) / conditional (c) / selector (s) indicator in column 2; an attribute type label in column 3; a name or a conditional expression in column 4; and optionally a list of enumerated values in column 5. In the column following the list of values, the default value for the attribute may be specified.
  - b) Objects are normally identified by a numeric identifier or by an object name, or by both. In the class templates, these key attributes are defined under the key attribute.
  - c) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting is used to specify:
    - i) fields of a structured attribute (4.1, 4.2, 4.3),
    - ii) attributes conditional on a constraint statement (5). Attributes may be mandatory (5.1) or optional (5.2) if the constraint is true. Not all optional attributes require constraint statements as does the attribute defined in (5.2),
    - iii) the selection fields of a choice type attribute (6.1 and 6.2).
- (6) The "SERVICES" label indicates that the following entries are services defined for the class.
- a) An (m) in column 2 indicates that the service is mandatory for the class, while an (o) indicates that it is optional. A (c) in this column indicates that the service is conditional. When all services defined for a class are defined as optional, at least one has to be selected when an instance of the class is defined.
  - b) The label "OpsService" designates an operational service (1).
  - c) The label "MgtService" designates a management service (2).
  - d) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting within the list of services is used to specify services conditional on a constraint statement.

### 3.4.3 Conventions for service definitions

#### 3.4.3.1 General

The service model, service primitives and time-sequence diagrams used are entirely abstract descriptions; they do not represent a specification for implementation.

#### 3.4.3.2 Service parameters

Service primitives are used to represent service user or service provider interactions (ISO/IEC 10731). They convey parameters which indicate information available in the user or provider interaction. In any particular interface, not all parameters shall be explicitly stated.

The service specifications of this document use a tabular format to describe the component parameters of the ASE service primitives. The parameters which apply to each group of service primitives are set out in tables. Each table consists of up to five columns for the

- a) parameter name,
- b) request primitive,
- c) indication primitive,
- d) response primitive, and

e) confirm primitive.

One parameter (or component 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 specified in the column:

M parameter is mandatory for the primitive

U parameter is a user option, and can be provided or not depending on dynamic usage of the service user. When not provided, a default value for the parameter is assumed.

— (blank) parameter is never present.

S parameter is a selected item.

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.

### 3.4.3.3 Service procedures

The procedures are defined in terms of

- the interactions between application entities through the exchange of fieldbus Application Protocol Data Units, and
- the interactions between an application layer service provider and an application layer service user in the same system through the invocation of application layer service primitives.

These procedures are applicable to instances of communication between systems which support time-constrained communications services within the fieldbus Application Layer.

## 4 Concepts

The fundamental concepts and templates used to describe the application layer service in this document follow IEC 61158-1, Clause 9.

## 5 Data type ASE

### 5.1 General

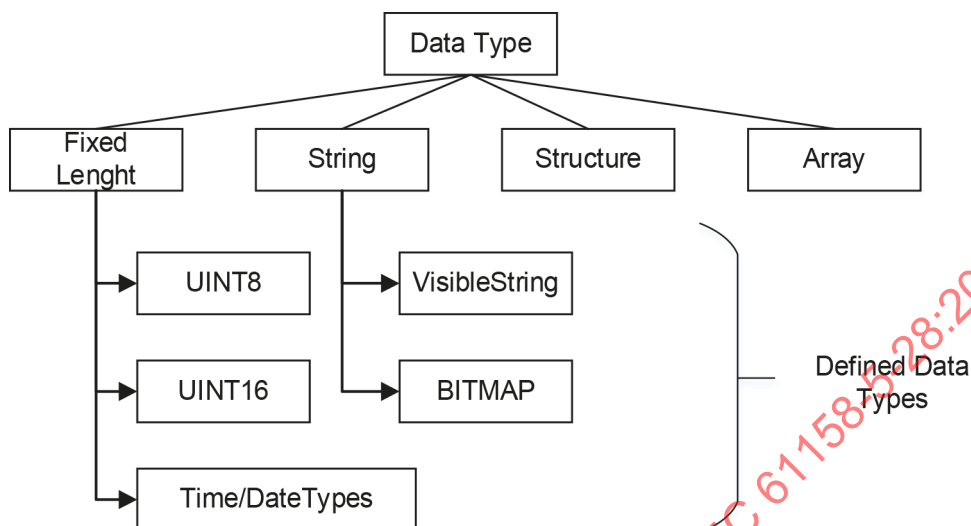
#### 5.1.1 Overview

In the communication Type 28 FAL, Data type ASE complies with IEC 61158-1, Clause 10, and the other protocol-specific data types are not defined in this document.

Fieldbus data types specify the machine independent syntax for the data conveyed by each layer services. The fieldbus application supports the definition and transmission of basic and constructed data types.

Basic types are atomic data types that cannot be decomposed into more elemental types. Constructed data types are composed of basic types and other constructed types. Their complexity and the depth of nesting are not constrained by this document.

Data types are defined as instances of the data type class, as shown in Figure 1. Only a subset of the data types defined in Clause 5 are shown in Figure 1. Defining new types is accomplished by providing a new numeric ID and supplying values for the attributes defined for the data type class.



IEC

**Figure 1 – Data type class hierarchy example**

The data type definitions in Figure 1 are represented as a class/format/instance structure beginning with data type class entitled "Data Type". The formats for data types are defined by the data type class and represented in Figure 1.

The basic data classes are used to define fixed length and bit string data types. Standard types taken from ISO/IEC 8824 are referred to as simple data types. Other standard basic data types are defined specifically for fieldbus applications and are referred to as specific types.

The constructed types specified in this document are strings, arrays and structures. There are no standard types defined for arrays and structures.

### 5.1.2 Overview of basic types

Most basic types are defined from a set of ISO/IEC 8824 types (simple types). Some ISO/IEC 8824 types have been extended for fieldbus specific use (specific types).

Simple types are ISO/IEC 8824 universal types. They are defined in this document to provide them with fieldbus class identifiers.

Specific types are basic types defined specifically for use in the fieldbus environment. They are defined as simple class subtypes.

Basic types have a constant length. Two variations are defined, one for defining data types whose length is an integral number of octets, and one for defining data types whose length is a number of bits.

NOTE Boolean, Integer, OctetString, VisibleString, and UniversalTime are defined in this document for the purpose of assigning fieldbus class identifiers to them. This document does not change their definitions as specified in ISO/IEC 8824.

### 5.1.3 Overview of fixed-length types

The length of fixed-length types is an integral number of octets.

## 5.1.4 Overview of constructed types

### 5.1.4.1 Strings

A string is composed of an ordered set, variable in number, of homogeneously typed fixed-length elements.

### 5.1.4.2 Arrays

An array is composed of an ordered set of homogeneously typed elements. There are no restrictions on the data type of array elements placed, but each element shall be of the same type. Once defined, the number of elements in an array can be changed, or not.

### 5.1.4.3 Structures

A structure is made of an ordered set of heterogeneously-typed elements called fields. Like arrays, this document does not restrict the data type of fields. However, the fields within a structure do not have to be of the same type.

### 5.1.4.4 Nesting level

This document permits arrays and structures to contain other arrays and structures. It places no restriction on the number of nesting levels allowed.

When an array or structure contains constructed elements, access to a single element in its entirety is provided. Access to sub elements of the constructed element is also provided.

## 5.2 AL defined data types

### 5.2.1 Fixed length types

#### 5.2.1.1 Date/Time types

##### 5.2.1.1.1 Time type (TIMEV)

The time data type is an unsigned integer of 64 bits, which represents the time incremented with 1 ns. Table 1 shows the coding.

**Table 1 – TIMEV type data coding**

| octet | bit      |          |          |          |          |          |          |          |
|-------|----------|----------|----------|----------|----------|----------|----------|----------|
|       | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |
| 1     | $2^{63}$ | $2^{62}$ | $2^{61}$ | $2^{60}$ | $2^{59}$ | $2^{58}$ | $2^{57}$ | $2^{56}$ |
| 2     | $2^{55}$ | $2^{54}$ | $2^{53}$ | $2^{52}$ | $2^{51}$ | $2^{50}$ | $2^{49}$ | $2^{48}$ |
| 3     | $2^{47}$ | $2^{46}$ | $2^{45}$ | $2^{44}$ | $2^{43}$ | $2^{42}$ | $2^{41}$ | $2^{40}$ |
| 4     | $2^{39}$ | $2^{38}$ | $2^{37}$ | $2^{36}$ | $2^{35}$ | $2^{34}$ | $2^{33}$ | $2^{32}$ |
| 5     | $2^{31}$ | $2^{30}$ | $2^{29}$ | $2^{28}$ | $2^{27}$ | $2^{26}$ | $2^{25}$ | $2^{24}$ |
| 6     | $2^{23}$ | $2^{22}$ | $2^{21}$ | $2^{20}$ | $2^{19}$ | $2^{18}$ | $2^{17}$ | $2^{16}$ |
| 7     | $2^{15}$ | $2^{14}$ | $2^{13}$ | $2^{12}$ | $2^{11}$ | $2^{10}$ | $2^9$    | $2^8$    |
| 8     | $2^7$    | $2^6$    | $2^5$    | $2^4$    | $2^3$    | $2^2$    | $2^1$    | $2^0$    |

**CLASS:**

**DATA TYPE**

**ATTRIBUTES:**

- |   |                              |         |
|---|------------------------------|---------|
| 1 | Data Type Numeric Identifier | = 16    |
| 2 | Data Type Name               | = TIMEV |

3 Format = FIXED LENGTH

4.1 Octet Length = 8

### 5.2.1.1.2 Date type coding (TIMEDATE)

The TIMEDATE data type length is seven octets, indicating date and time. Table 2 shows the meaning of each octet.

**Table 2 – TIMEDATE type coding**

| octet | Bit      |          |          |          |          |          |       |       | Description                                                                                                                                                        |
|-------|----------|----------|----------|----------|----------|----------|-------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | 7        | 6        | 5        | 4        | 3        | 2        | 1     | 0     |                                                                                                                                                                    |
| 1     | $2^{15}$ | $2^{14}$ | $2^{13}$ | $2^{12}$ | $2^{11}$ | $2^{10}$ | $2^9$ | $2^8$ | Indicates the millisecond value less than a minute                                                                                                                 |
| 2     | $2^7$    | $2^6$    | $2^5$    | $2^4$    | $2^3$    | $2^2$    | $2^1$ | $2^0$ |                                                                                                                                                                    |
| 3     | $2^7$    | $2^6$    | $2^5$    | $2^4$    | $2^3$    | $2^2$    | $2^1$ | $2^0$ | Indicates the minute value less than an hour                                                                                                                       |
| 4     | $2^7$    | $2^6$    | $2^5$    | $2^4$    | $2^3$    | $2^2$    | $2^1$ | $2^0$ | Indicates the hour values less than a day                                                                                                                          |
| 5     | $2^2$    | $2^1$    | $2^0$    | $2^4$    | $2^3$    | $2^2$    | $2^1$ | $2^0$ | The first 3 bits means the day of the week, and the last 5 bits is the day of the current month                                                                    |
| 6     | $2^7$    | $2^6$    | $2^5$    | $2^4$    | $2^3$    | $2^2$    | $2^1$ | $2^0$ | Indicates the month                                                                                                                                                |
| 7     | $2^7$    | $2^6$    | $2^5$    | $2^4$    | $2^3$    | $2^2$    | $2^1$ | $2^0$ | Indicates the year. The value range from 0 to 50 corresponds to range from 2000 to 2050, and the value range from 51 to 99 corresponds to range from 1951 to 1999. |

**CLASS:**

**DATA TYPE**

**ATTRIBUTES:**

1 Data Type Numeric Identifier = 17  
 2 Data Type Name = TIMEDATE  
 3 Format = FIXED LENGTH  
 4.1 Octet Length = 7

### 5.2.1.1.3 Time of day type coding (TIMEOFDAY)

TIMEOFDAY data type length is six octets and indicates the time of day. Table 3 shows the meaning of each octet.

**Table 3 – TIMEOFDAY type coding**

| octet | Bit      |          |          |          |          |          |          |          | Description                                                              |
|-------|----------|----------|----------|----------|----------|----------|----------|----------|--------------------------------------------------------------------------|
|       | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |                                                                          |
| 1     | $2^{31}$ | $2^{30}$ | $2^{29}$ | $2^{28}$ | $2^{27}$ | $2^{26}$ | $2^{25}$ | $2^{24}$ | Indicates the nanosecond count starting at 0:00 midnight                 |
| 2     | $2^{23}$ | $2^{22}$ | $2^{21}$ | $2^{20}$ | $2^{19}$ | $2^{18}$ | $2^{17}$ | $2^{16}$ |                                                                          |
| 3     | $2^{15}$ | $2^{14}$ | $2^{13}$ | $2^{12}$ | $2^{11}$ | $2^{10}$ | $2^9$    | $2^8$    |                                                                          |
| 4     | $2^7$    | $2^6$    | $2^5$    | $2^4$    | $2^3$    | $2^2$    | $2^1$    | $2^0$    |                                                                          |
| 5     | $2^{15}$ | $2^{14}$ | $2^{13}$ | $2^{12}$ | $2^{11}$ | $2^{10}$ | $2^9$    | $2^8$    | Indicates the number of days that have been counted since 1 January 1984 |
| 6     | $2^7$    | $2^6$    | $2^5$    | $2^4$    | $2^3$    | $2^2$    | $2^1$    | $2^0$    |                                                                          |

**CLASS:****DATA TYPE****ATTRIBUTES:**

|     |                              |                |
|-----|------------------------------|----------------|
| 1   | Data Type Numeric Identifier | = 18           |
| 2   | Data Type Name               | = TIMEOFDAY    |
| 3   | Format                       | = FIXED LENGTH |
| 4.1 | Octet Length                 | = 6            |

**5.2.1.1.4 Time difference type coding (TIMEDIFFER)**

TIMEDIFFER data type length is six octets and indicates time difference. Table 4 shows the meaning of each octet.

**Table 4 – TIMEDIFFER type coding**

| octet | Bit      |          |          |          |          |          |          |          | Description                                                   |
|-------|----------|----------|----------|----------|----------|----------|----------|----------|---------------------------------------------------------------|
|       | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |                                                               |
| 1     | $2^{31}$ | $2^{30}$ | $2^{29}$ | $2^{28}$ | $2^{27}$ | $2^{26}$ | $2^{25}$ | $2^{24}$ | Indicates the deviation of nanosecond values less than a day. |
| 2     | $2^{23}$ | $2^{22}$ | $2^{21}$ | $2^{20}$ | $2^{19}$ | $2^{18}$ | $2^{17}$ | $2^{16}$ |                                                               |
| 3     | $2^{15}$ | $2^{14}$ | $2^{13}$ | $2^{12}$ | $2^{11}$ | $2^{10}$ | $2^9$    | $2^8$    |                                                               |
| 4     | $2^7$    | $2^6$    | $2^5$    | $2^4$    | $2^3$    | $2^2$    | $2^1$    | $2^0$    |                                                               |
| 5     | $2^{15}$ | $2^{14}$ | $2^{13}$ | $2^{12}$ | $2^{11}$ | $2^{10}$ | $2^9$    | $2^8$    | Indicates the number of days of deviation                     |
| 6     | $2^7$    | $2^6$    | $2^5$    | $2^4$    | $2^3$    | $2^2$    | $2^1$    | $2^0$    |                                                               |

**CLASS:****DATA TYPE****ATTRIBUTES:**

|     |                              |                |
|-----|------------------------------|----------------|
| 1   | Data Type Numeric Identifier | = 19           |
| 2   | Data Type Name               | = TIMEDIFFER   |
| 3   | Format                       | = FIXED LENGTH |
| 4.1 | Octet Length                 | = 6            |

**5.2.2 String type****5.2.2.1 Visible string type (VISIBLESTRING)**

VISIBLESTRING type is a set of octets in order. The length is not fixed, and it is 1 to N octet. Table 5 shows the string coding. For strings of N octets, the MSB of the highest valid octet of this type of data is transmitted at first.

**Table 5 – VISIBLESTRING type data coding**

| octet | bit        |             |             |             |             |             |             |             |
|-------|------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|       | 7          | 6           | 5           | 4           | 3           | 2           | 1           | 0           |
| 1     | $2^{8N-1}$ | $2^{8N-2}$  | $2^{8N-3}$  | $2^{8N-4}$  | $2^{8N-5}$  | $2^{8N-6}$  | $2^{8N-7}$  | $2^{8N-8}$  |
| 2     | $2^{8N-9}$ | $2^{8N-10}$ | $2^{8N-11}$ | $2^{8N-12}$ | $2^{8N-13}$ | $2^{8N-14}$ | $2^{8N-15}$ | $2^{8N-16}$ |
| ...   | ...        | ...         | ...         | ...         | ...         | ...         | ...         | ...         |
| N     | $2^7$      | $2^6$       | $2^5$       | $2^4$       | $2^3$       | $2^2$       | $2^1$       | $2^0$       |

**CLASS:****DATA TYPE****ATTRIBUTES:**

|     |                              |                 |
|-----|------------------------------|-----------------|
| 1   | Data Type Numeric Identifier | = 32            |
| 2   | Data Type Name               | = VISIBLESTRING |
| 3   | Format                       | = STRING        |
| 4.1 | Octet Length                 | = 1 to n        |

### 5.2.2.2 Bitmap type coding (BITMAP)

BITMAP type length ranges in value from 2 to 256 octets, including Width and Mask\_Info information.

Figure 2 shows the bitmap type coding.

NOTE Width has 1octet, its value indicates the number of the octets of Mask\_Info.

NOTE Mask\_Info represented by bit. The octet length is Width value and the bit length is Width × 8.

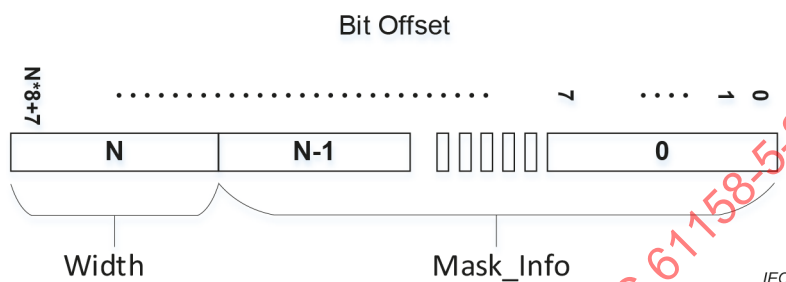


Figure 2 – BITMAP data type

CLASS:

DATA TYPE

ATTRIBUTES:

- |     |                              |          |
|-----|------------------------------|----------|
| 1   | Data Type Numeric Identifier | = 33     |
| 2   | Data Type Name               | = BITMAP |
| 3   | Format                       | = STRING |
| 4.1 | Octet Length                 | = 1 to n |

### 5.2.3 Structure type

#### 5.2.3.1 Time information structure

- Structure name: TIMEINFO\_S
- Number of structural members: 5
- Description of structure member: see Table 6.

Table 6 – Description of time information structure TIMEINFO\_S

| Member name | Type of Data | Description                                                                                                                                                                                                                                                                                                    |
|-------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Time_fmt    | UINT8        | Identify the valid bits of three time formats based on bits, bit0, bit1, and bit2 respectively. Identify whether the time values are valid based on hour, minute and second, millisecond and microsecond and second. When the corresponding position is 1, it means that it is valid, otherwise it is invalid. |
| Time_ns     | TIMEV        | Time count based on nanosecond.                                                                                                                                                                                                                                                                                |
| TimeDate    | TIMEDATE     | Convert to date value of year, month, day, minute, minute, second, and precision is millisecond.                                                                                                                                                                                                               |
| TimeOfDay   | TIMEOFDAY    | The number of days based on nanosecond.                                                                                                                                                                                                                                                                        |
| TimeDiffer  | TIMEDIFFER   | Time offset value based on nanosecond.                                                                                                                                                                                                                                                                         |

**CLASS: DATA TYPE****ATTRIBUTES:**

|        |                  |              |
|--------|------------------|--------------|
| 2      | Data Type Name   | = TIMEINFO_S |
| 5      | Format           | = STRUCTURE  |
| 5.1    | Number of Fields | = 5          |
| 5.2    | List of Fields   |              |
| 5.2.1  | Field Name       | = Time_fmt   |
| 5.2.2  | Field Data Type  | UINT8        |
| 5.2.3  | Field Name       | = Time_ns    |
| 5.2.4  | Field Data Type  | TIMEV        |
| 5.2.5  | Field Name       | = TimeDate   |
| 5.2.6  | Field Data Type  | TIMEDATE     |
| 5.2.7  | Field Name       | = TimeOfDay  |
| 5.2.8  | Field Data Type  | TIMEOFDAY    |
| 5.2.9  | Field Name       | = TimeDiffer |
| 5.2.10 | Field Data Type  | TIMEDIFFER   |

**5.2.3.2 Clock option information structure**

- Structure name: CLOCK\_OPTION\_INFO\_S
- Number of structural members: 5
- Description of structure member, see Table 7.

**Table 7 – Description of clock option information structure CLOCK\_OPTION\_INFO\_S**

| Member name | Type of Data | Description                                                                                              |
|-------------|--------------|----------------------------------------------------------------------------------------------------------|
| Tid_cnt     | UINT8        | Number of clocks on the device or module                                                                 |
| Tid_val     | UINT8        | Clock ID value                                                                                           |
| Tid_attr    | UINT8        | The control-field attribute of the clock, 0 is the control clock (list), 1 is the field clock (list)     |
| Tcd_lst     | UINT8        | TCD command for the corresponding clock, the value range of TCD is identical to IEC 61158-6-28, Clause 4 |
| Tid_errcode | UINT8        | A clock error exception encoding list                                                                    |

**CLASS: DATA TYPE****ATTRIBUTES:**

|       |                  |                       |
|-------|------------------|-----------------------|
| 2     | Data Type Name   | = CLOCK_OPTION_INFO_S |
| 5     | Format           | = STRUCTURE           |
| 5.1   | Number of Fields | = 5                   |
| 5.2   | List of Fields   |                       |
| 5.2.1 | Field Name       | = Tid_cnt             |
| 5.2.2 | Field Data Type  | UINT8                 |
| 5.2.3 | Field Name       | = Tid_val             |
| 5.2.4 | Field Data Type  | UINT8                 |
| 5.2.5 | Field Name       | = Tid_attr            |
| 5.2.6 | Field Data Type  | UINT8                 |
| 5.2.7 | Field Name       | = Tcd_lst             |
| 5.2.8 | Field Data Type  | UINT8                 |
| 5.2.9 | Field Name       | = Tid_errcode         |

## 5.2.10

## Field Data Type

## UINT8

**5.2.3.3 Network configuration parameter structure**

- Structure name: NETWORK\_CFG\_PARA\_S
- Number of structural members: 15
- Description of structure member: see Table 8.

**Table 8 – Description of network configuration parameter structure  
NETWORK\_CFG\_PARA\_S**

| Member name         | Type of Data | Description                                                                                                                                                                                                                                                                  |
|---------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Network_id          | UINT32       | Identify the NETWORKID (see 6.6.6.1.2) of device or module                                                                                                                                                                                                                   |
| Ip_switch           | UINT8        | Whether the IPv4 (defined in IETF RFC 791) address or IPv6 (defined in IETF RFC 2460) address is supported by bit.<br>Bit 0: whether IPv4 is supported or not.<br>Bit 1: whether IPv6 is supported, other bits are not used yet;<br>Corresponding bit = 1 indicates support. |
| Multicast_id0 to 15 | UINT32       | NETWORKID of the multicast type multicast group to which the identified device or module belongs. Each device or module belongs to maximum of 16 multicast groups.                                                                                                           |
| Mac_addr            | ARRAY        | Unique MAC (defined in ISO/IEC/IEEE 8802-3) address of the device or module                                                                                                                                                                                                  |
| Nodeid              | UINT8        | Node ID address of the device                                                                                                                                                                                                                                                |
| Dctime              | UINT16       | Data calculation cycle time                                                                                                                                                                                                                                                  |
| Bctime              | UINT16       | Bus service cycle time                                                                                                                                                                                                                                                       |
| Wctime              | UINT16       | Bus service data write cycle time                                                                                                                                                                                                                                            |
| Rctime              | UINT16       | Bus service data read cycle time                                                                                                                                                                                                                                             |
| Ctime               | UINT16       | Bus service competition cycle time                                                                                                                                                                                                                                           |
| Totcnt              | UINT16       | Number of timeouts allowed by the device or module                                                                                                                                                                                                                           |
| Rt_data_addr        | UINT32       | Start address of RT data buffer                                                                                                                                                                                                                                              |
| Rt_data_size        | UINT32       | Size of RT data buffer                                                                                                                                                                                                                                                       |
| nRt_data_addr       | UINT32       | Start address of nRT buffer                                                                                                                                                                                                                                                  |
| nRt_data_size       | UINT32       | Size of nRT data buffer                                                                                                                                                                                                                                                      |

**CLASS:****DATA TYPE****ATTRIBUTES:**

|       |                  |                                   |
|-------|------------------|-----------------------------------|
| 2     | Data Type Name   | = NETWORK_CFG_PARA_S              |
| 5     | Format           | = STRUCTURE                       |
| 5.1   | Number of Fields | = 15                              |
| 5.2   | List of Fields   |                                   |
| 5.2.1 | Field Name       | = Network_id                      |
| 5.2.2 | Field Data Type  | UINT32                            |
| 5.2.3 | Field Name       | = Ip_switch                       |
| 5.2.4 | Field Data Type  | UINT8                             |
| 5.2.5 | Field Name       | = Multicast_id0 to Multicast_id15 |
| 5.2.6 | Field Data Type  | UINT8                             |
| 5.2.7 | Field Name       | = Mac_addr                        |
| 5.2.8 | Field Data Type  | ARRAY                             |
| 5.2.9 | Field Name       | = Nodeid                          |

|        |                          |                 |
|--------|--------------------------|-----------------|
| 5.2.10 | Field Data Type          | UINT8           |
| 5.2.11 | Field Name               | = Dctime        |
| 5.2.12 | Field Data Type          | UINT16          |
| 5.2.13 | Field Name               | = Bctime        |
| 5.2.14 | Field Data Type          | UINT16          |
| 5.2.15 | Field Name               | = Wctime        |
| 5.2.16 | Field Data Type          | UINT16          |
| 5.2.17 | Field Name               | = Rctime        |
| 5.2.18 | Field Data Type          | UINT16          |
| 5.2.19 | Field Name               | = Ctime         |
| 5.2.20 | Field Data Type          | UINT16          |
| 5.2.21 | Field Name               | = Totcnt        |
| 5.2.22 | Field Data Type          | UINT16          |
| 5.2.23 | Field Name               | = Rt_data_addr  |
| 5.2.24 | Field Data Type          | UINT32          |
| 5.2.25 | Field Name               | = Rt_data_size  |
| 5.2.26 | Field Data Type          | UINT32          |
| 5.2.27 | Field Name               | = nRt_data_addr |
| 5.2.28 | Field Data Type          | UINT32          |
| 5.2.29 | Field Name               | = nRt_data_size |
| 5.2.30 | Field Data Type          | UINT32          |
| 6      | Constraint               | Format = ARRAY  |
| 6.1    | Number of Array Elements | 6               |
| 6.2    | Array Element Data Type  | UINT8           |

#### 5.2.3.4 MAC mapping table structure

- Structure name: NETWORKID\_MAC\_MAP\_TABLE\_S
- Number of structural members: 7
- Description of structure member: see Table 9.

**Table 9 – Description of MAC mapping table NETWORKID\_MAC\_MAP\_TABLE\_S**

| Member name | Type of Data | Description                                                                                                                                                                                                                                 |
|-------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Index       | UINT16       | Entry index                                                                                                                                                                                                                                 |
| NodeID      | UINT8        | Node address                                                                                                                                                                                                                                |
| RoleID      | UINT8        | Control-field attribute of device or module, 0x00 is the control device, 0x01 is the field device, 0x02 is the unknown. Other values are reserved.                                                                                          |
| IP_Flag     | UINT8        | The device or module supports which type of IP.<br>Bit 0 identifies IPv4 (defined in IETF RFC 791).<br>Bit 1 identifies IPv6 (defined in IETF RFC 2460).<br>The corresponding bit position 1 indicates supporting. Other bits are reserved. |
| StatusID    | UINT8        | The status ID of the device or module. The value 0x00 indicates the initialization status is OK, and the value 0x01 indicates the normal working status. Other values are reserved.                                                         |
| Network_ID  | UINT32       | Unique NETWORKID ID of the device or module                                                                                                                                                                                                 |
| MacAddr     | ARRAY        | MAC (defined in ISO/IEC/IEEE 88023-3) address of the device or module                                                                                                                                                                       |

**CLASS:****DATA TYPE****ATTRIBUTES:**

|        |                          |                             |
|--------|--------------------------|-----------------------------|
| 2      | Data Type Name           | = NETWORKID_MAC_MAP_TABLE_S |
| 5      | Format                   | = STRUCTURE                 |
| 5.1    | Number of Fields         | = 7                         |
| 5.2    | List of Fields           |                             |
| 5.2.1  | Field Name               | = Index                     |
| 5.2.2  | Field Data Type          | UINT16                      |
| 5.2.3  | Field Name               | = NodeID                    |
| 5.2.4  | Field Data Type          | UINT8                       |
| 5.2.5  | Field Name               | = RoleID                    |
| 5.2.6  | Field Data Type          | UINT8                       |
| 5.2.7  | Field Name               | = IP_Flag                   |
| 5.2.8  | Field Data Type          | UINT8                       |
| 5.2.9  | Field Name               | = StatusID                  |
| 5.2.10 | Field Data Type          | UINT8                       |
| 5.2.11 | Field Name               | = Network_ID                |
| 5.2.12 | Field Data Type          | UINT32                      |
| 5.2.13 | Field Name               | = MacAddr                   |
| 5.2.14 | Field Data Type          | ARRAY                       |
| 6      | Constraint               | Format = ARRAY              |
| 6.1    | Number of Array Elements | = 6                         |
| 6.2    | Array Element Data Type  | UINT8                       |

**5.2.3.5 Mapping table structure**

- Structure name: IP\_MAP\_TABLE\_S
- Number of structural members: 9
- Description of structure member: see Table 10.

**Table 10 – Description of IP mapping table structure IP\_MAP\_TABLE\_S**

| Member name | Type of Data | Description                                                                                                                                                                                 |
|-------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Index       | UINT16       | Entry index                                                                                                                                                                                 |
| Stated      | UINT8        | Value that identifies the status of the entry.<br>0x00 is invalid; 0x01 is valid; 0x02 is pending.<br>Other values are reserved.                                                            |
| Network_id  | UINT32       | Unique NETWORKID of the device or module                                                                                                                                                    |
| ipv4_addr   | UINT32       | IPv4 address (defined in IETF RFC 791).                                                                                                                                                     |
| ipv4_mask   | UINT32       | IPv4 mask (defined in IETF RFC 791).                                                                                                                                                        |
| ipv6_addr   | UINT128      | IPv6 address (defined in IETF RFC 2460)                                                                                                                                                     |
| ipv6_mask   | UINT128      | IPv6 mask (defined in IETF RFC 2460)                                                                                                                                                        |
| Dmac_addr   | ARRAY        | Unique physical MAC address of device or module                                                                                                                                             |
| Vmac_addr   | ARRAY        | Virtual MAC address of device or module. One physical device has one or more sub-devices or sub-modules, and each sub-device or sub-module uniquely corresponds to one virtual MAC address. |

**CLASS:****DATA TYPE****ATTRIBUTES:**

|        |                          |                  |
|--------|--------------------------|------------------|
| 2      | Data Type Name           | = IP_MAP_TABLE_S |
| 5      | Format                   | = STRUCTURE      |
| 5.1    | Number of Fields         | = 9              |
| 5.2    | List of Fields           |                  |
| 5.2.1  | Field Name               | = Index          |
| 5.2.2  | Field Data Type          | UINT16           |
| 5.2.3  | Field Name               | = Stated         |
| 5.2.4  | Field Data Type          | UINT8            |
| 5.2.5  | Field Name               | = Network_id     |
| 5.2.6  | Field Data Type          | UINT32           |
| 5.2.7  | Field Name               | = ipv4_addr      |
| 5.2.8  | Field Data Type          | UINT32           |
| 5.2.9  | Field Name               | = ipv4_mask      |
| 5.2.10 | Field Data Type          | UINT32           |
| 5.2.11 | Field Name               | = ipv6_addr      |
| 5.2.12 | Field Data Type          | UINT128          |
| 5.2.13 | Field Name               | = ipv6_mask      |
| 5.2.14 | Field Data Type          | UINT128          |
| 5.2.15 | Field Name               | = Dmac_addr      |
| 5.2.16 | Field Data Type          | ARRAY            |
| 5.2.17 | Field Name               | = Vmac_addr      |
| 5.2.18 | Field Data Type          | ARRAY            |
| 6      | Constraint               | Format = ARRAY   |
| 6.1    | Number of Array Elements | 6                |
| 6.2    | Array Element Data Type  | UINT8            |
| 6.3    | Number of Array Elements | 6                |
| 6.4    | Array Element Data Type  | UINT8            |

## 6 Type 28 communication model specification

### 6.1 General

The Type 28 AL processes RT and nRT application services based on the Type 28 DLL and PhL, which provides three kinds of service functions: time service, data service and system management service. The position of AL in the Type 28 protocol stack architecture is shown in Figure 3.

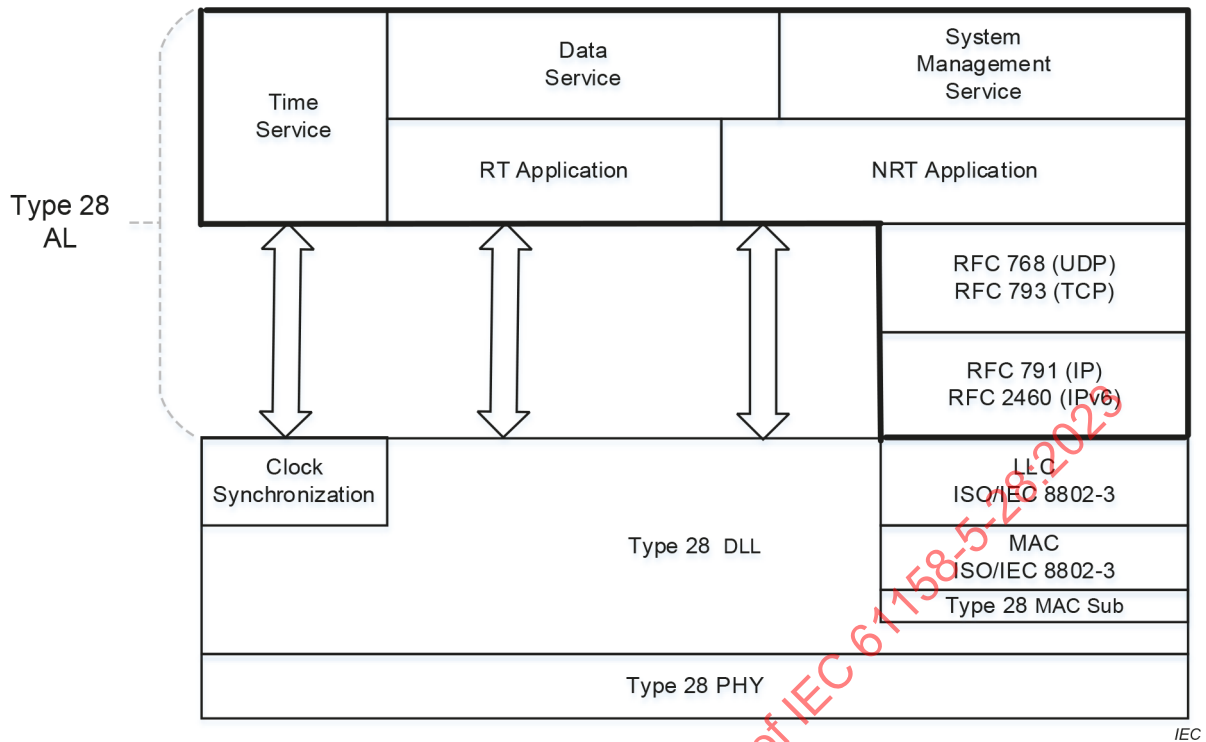


Figure 3 – AL in fieldbus protocol stack architecture

## 6.2 Protocol stack for Type 28 fieldbus

Figure 4 shows the Type 28 AL protocol architecture.

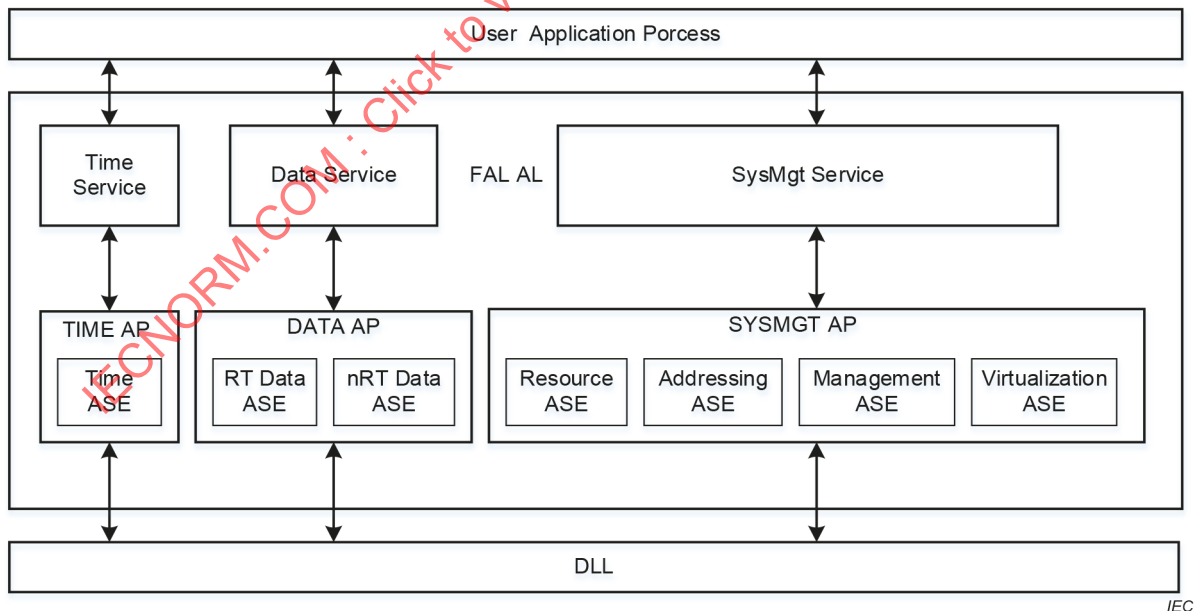


Figure 4 – Type 28 AL protocol architecture

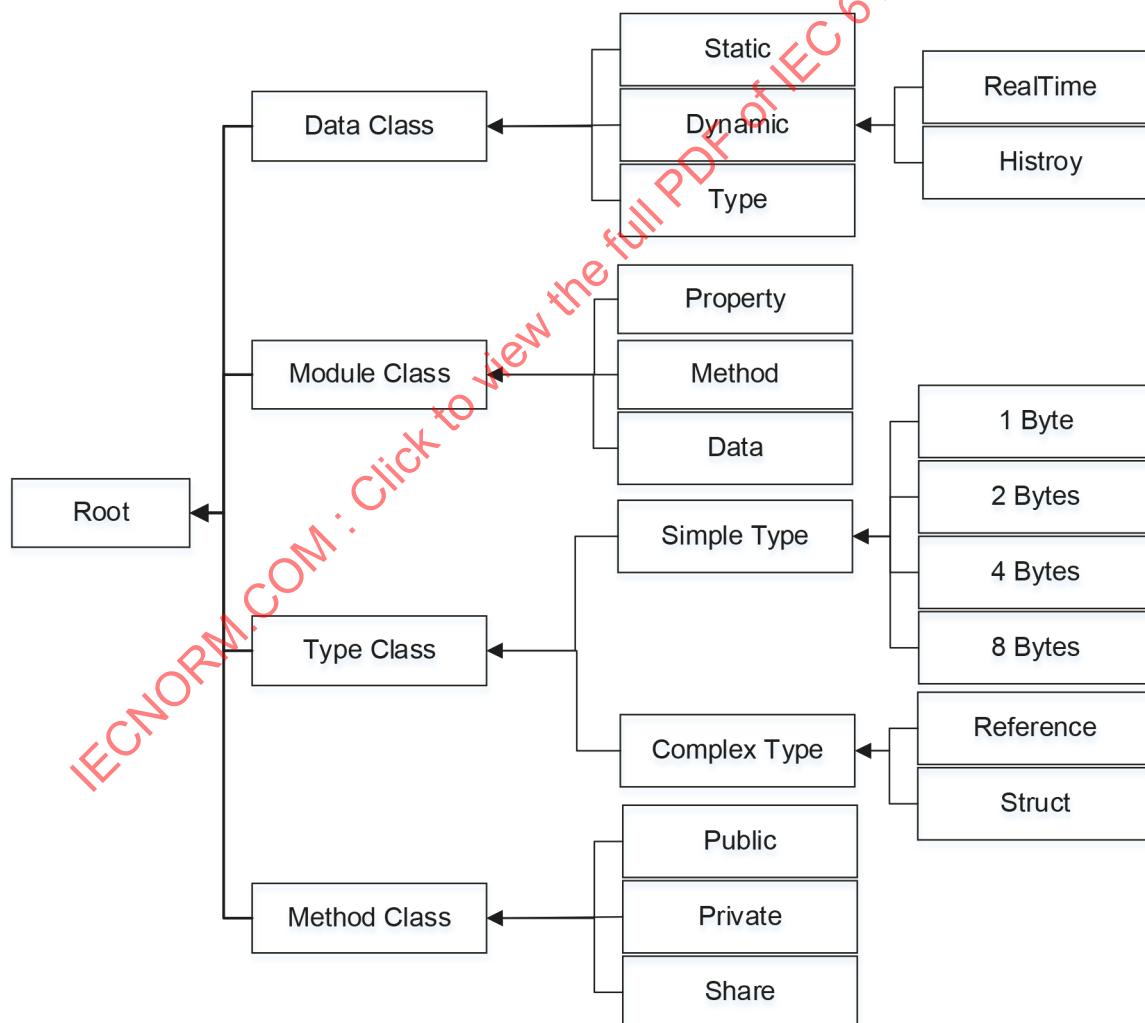
Type 28 AL provides users with time service, data service and system management service, which are implemented by the application service element (ASE) of the corresponding Application Process (AP). Time service implements time synchronization and time query based on time ASE. Data service processes the data with different real-time requirements based on RT data ASE and nRT data ASE. System management service provides the network system

with resource application service, addressing application services, managing application services and virtualization application services. The Type 28 AL protocol implements the specific AL service functions through the Type 28 DLL and service interfaces provided by the lower layer.

### 6.3 Data mapping modelling

In the Type 28 network, there is a service data buffer structure on the device or module with RT data buffer or the nRT data buffer based on the RT of the service data processing. The RT data buffer distinguishes the real-time cyclic (RTC) data buffer and real-time acyclic (RTA) data buffer according to the different cyclic processing methods of the data. There are several receiving and transmitting addresses for each type of data buffer. The AP related service data in the Type 28 network maps to the corresponding position in the data buffer using data modelling.

The APO definition involved in the AP of Type 28 AL protocol adopts the object-oriented method. All APO classes inherit from the unique root node (Root), and the definition of all APOs includes four object classes, Data Class, Module Class, Type Class, and Method Class as shown in Figure 5.



IEC

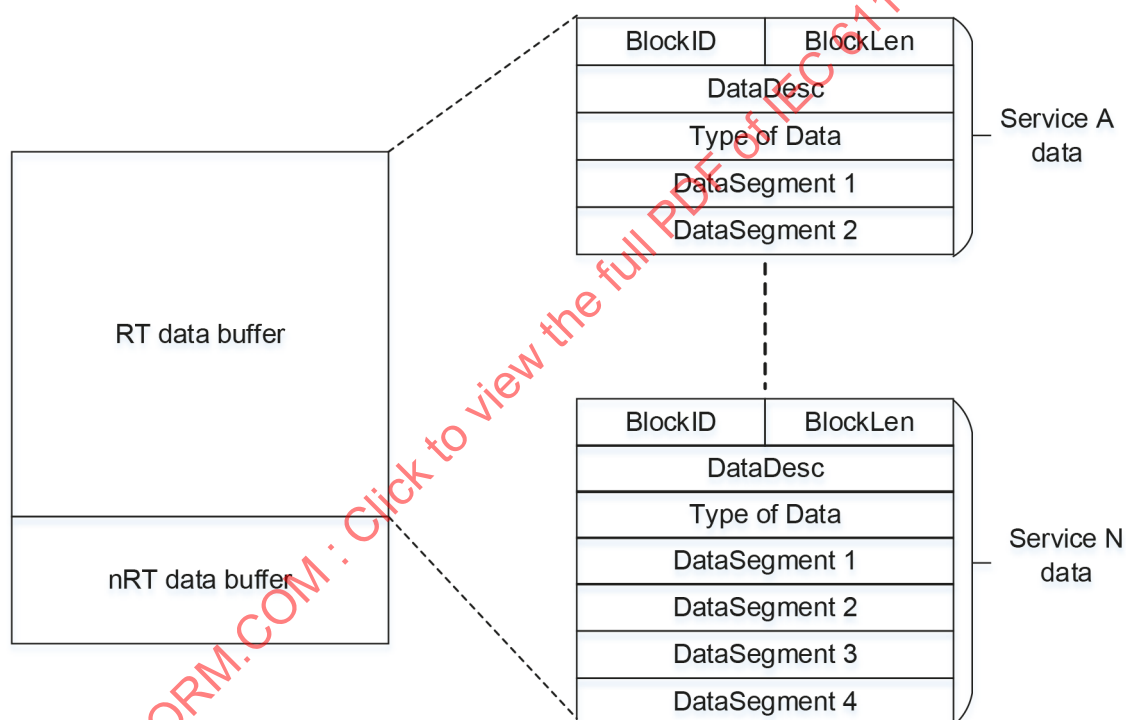
**Figure 5 – Data mapping modelling object structure**

In the Type 28 network environment, devices or modules describe specific AP interaction using data and method. The data content includes the attributes, features, types and data that need to be processed interactively of device or module. The data is described in simple data type or

complex data type. The methods can be private methods, public methods or shared methods. When device or module performs specific AP interaction, it needs to describe with the specific data and methods, and maps the buffer data to the actual data message to transmit. This modelling methods and data transmission methods are directly related to the data message format, which can improve the communication efficiency and performance of the bus (from the perspective of data message encapsulation and parsing). See Annex A for an example of service data mapping for DTU messages.

The data mapping modelling method provided by the Type 28 AL protocol is suitable to various industrial bus environments. If the object dictionary is used to define the AL data, based on the object class defined by the Type 28 AL protocol, different object data is instantiated separately, and then mapped into data messages transmitted by the communication to meet the AP interaction requirements. The mapping model method is well compatible with the data model of OPC UA. See Annex B for a data modelling description using a sensor as an example.

Modelling the APO data in the Type 28 network environment, the actual service data variables are represented by buffer blocks in the data buffer, and each block has deterministic information such as the block ID and the block length as shown in Figure 6.



IEC

**Figure 6 – Diagram of mapping data buffers and service data variables**

Description:

- BlockID: The starting address of data buffer block corresponding to the service data.
- BlockLen: Valid data length of the data buffer block corresponding to the service data, that is, the total length from the beginning of the data segment to the end of the data segment.
- DataDesc: Description of the association with the actual application service.
- Type of Data: Determines the size of the data segment.
- DataSegment: Corresponds to the actual service data variable.

In the Type 28 network, the interaction of AP should access the buffer block data by the globally unique DataID. There is a service data mapping table keyed using the unique DataID that exists

on the device or module. The content of every unique entry is associated with the actual data buffer block data. The parameters of the table are shown in Table 11.

**Table 11 – Service data mapping table parameter list**

| Entry name | Type of Data | Description of entry                                                                           |
|------------|--------------|------------------------------------------------------------------------------------------------|
| DataID     | UINT16       | ID that uniquely identifies the service data                                                   |
| BuffType   | UINT8        | Buffer type, 0x00: RT, 0x04: nRT, other values are reserved.                                   |
| DataState  | UINT8        | Status of the data block, 0x00: valid, 0x04: in use, 0x0F: invalid, other values are reserved. |
| BlockID    | UINT32       | Data buffer block ID                                                                           |
| NetworkID  | UINT32       | NETWORKID of device or module to which the service data belongs                                |

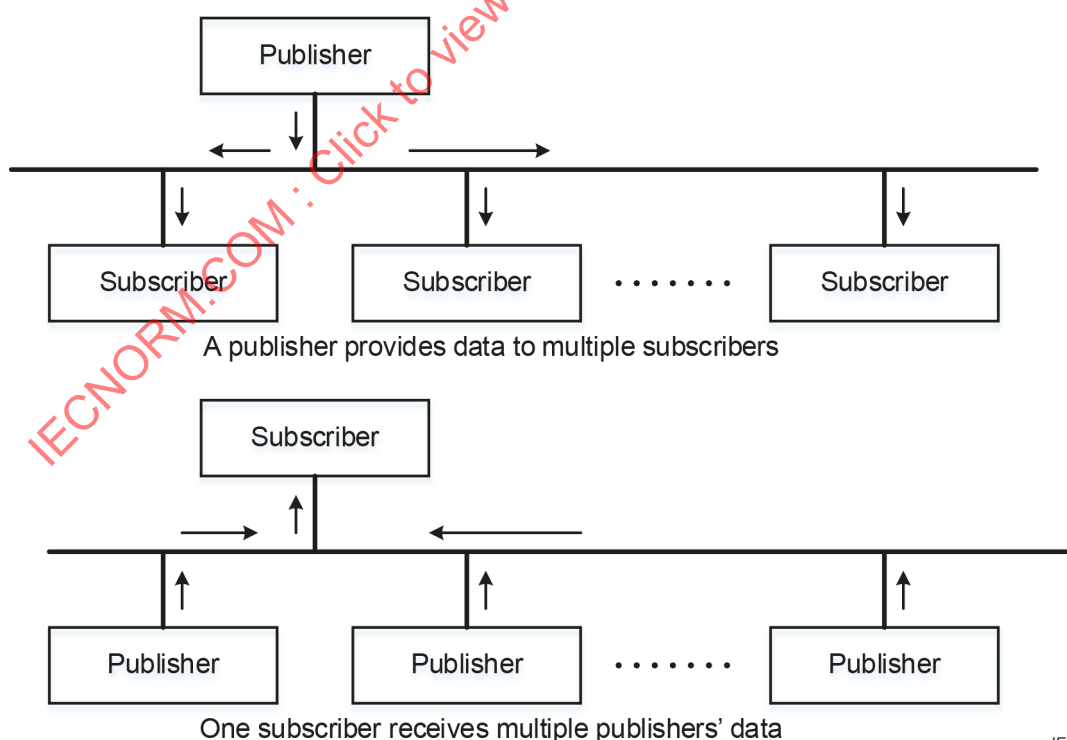
## 6.4 Overview of Type 28 communication model

### 6.4.1 General

The AP in the Type 28 network implements the determined service function objectives through interaction. The interactions between the APs can be based on the request/response message or on the data/events issued by each other APs. The communication models of these interactions methods are publisher/subscriber model (P/S) and client/server model (C/S).

### 6.4.2 P/S

In the process of AP interaction based on P/S model, one publisher provides data to one or more subscribers, and one subscriber receives data provided by one or more publishers. Diagram of P/S model is shown in Figure 7.

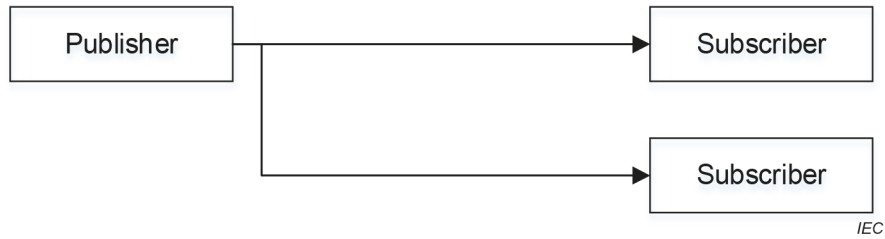


IEC

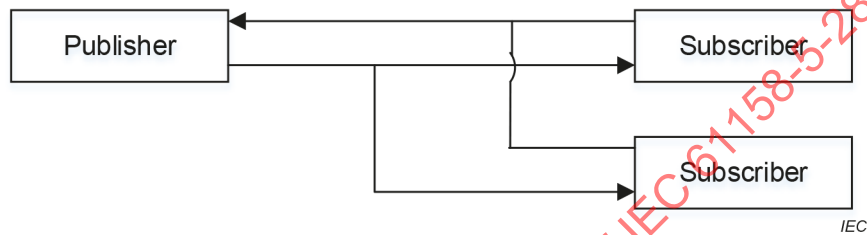
**Figure 7 – Diagram of P/S model**

On the network, the device or module provides service data as a publisher or receives service data as a subscriber based on AP interaction demand. The AP interaction of RT data and nRT

data between APs should adopt the P/S model. In the Type 28 network, the P/S-based communication model supports both push (PUSH) mode and pull (PULL) mode as shown in Figure 8 and Figure 9.



**Figure 8 – P/S model of PUSH mode**

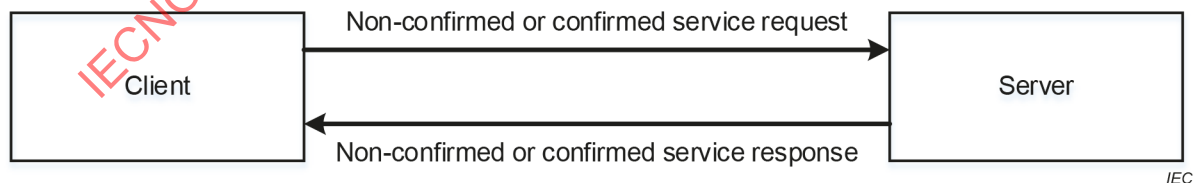


**Figure 9 – P/S model of PULL mode**

For PUSH mode, publisher actively sends data to the subscriber. This interactive communication mode does not need to confirm whether the subscriber receives the data correctly. For PULL mode, the subscriber actively requests the publisher to send data when data is needed, and application service based on this mode is usually required to be confirmed.

### 6.4.3 C/S

The AP interaction based on the C/S model uses bidirectional data stream to implement the determined service function by request and response. Based on this model, there are two modes: confirmed and non-confirmed. In the confirmed mode, the server needs to confirm the status and completion of the interaction process according to the response of the client to the response message. In non-confirmed mode, the server will confirm the status and the completion of the interaction process after making response processing against the client's request. The user's choice of confirmed and non-confirmed mode depends on the specific application interaction service. A C/S communication model diagram is shown in Figure 10.



**Figure 10 – Diagram of C/S communication model**

## 6.5 AL related management information table

### 6.5.1 General

The AL management information is consistent with the corresponding items in the system management configuration file, and exists in the form of software variables in the device. AL management information includes time management information and network management information. The AL system management user or the AL user should access the corresponding information entry through the AL management service interface.

## 6.5.2 Time management information table

**Table 12 – Time management information table**

| Parameter index  | Parameter name | RW  | Data Type  | Description                                                                                                                                                                                                                                                                                                                                                        |
|------------------|----------------|-----|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x2000           | time_fmt       | R/W | UINT8      | Different time formats represented by bit<br>Bit 0: indicates that time format is based on hour, minute and second.<br>Bit 1: indicates that time format is based on milliseconds and microseconds.<br>Bit 2: indicates that the time format is based on second pulse.<br>The corresponding bit set to 0 is invalid, set to 1 is valid<br>Bit 3 to bit 7: reserved |
| 0x2001           | time_ns        | R/W | TIMEV      | Time count based on nanosecond                                                                                                                                                                                                                                                                                                                                     |
| 0x2002           | timedate       | R/W | TIMEDATE   | Convert to date value of year, month, day, minute, , and second, which is accurate to milliseconds                                                                                                                                                                                                                                                                 |
| 0x2003           | timeofday      | R/W | TIMEOFDAY  | The number of days based on ns                                                                                                                                                                                                                                                                                                                                     |
| 0x2004           | timediffer     | R/W | TIMEOFDIFF | The time difference based on nanosecond                                                                                                                                                                                                                                                                                                                            |
| 0x2005           | tid_cnt        | R/W | UINT8      | Number of clocks on the device or module<br>Value range: 1 to 8<br>Value 0 is invalid.<br>Other values are reserved                                                                                                                                                                                                                                                |
| 0x2006           | tid_val        | R/W | UINT8      | Clock ID<br>Value range: 0 to 7<br>Other values are reserved                                                                                                                                                                                                                                                                                                       |
| 0x2007           | tid_attr       | R/W | UINT8      | Control field attribute of clock<br>Value range: 0x00 or 0x01<br>0x00: control clock<br>0x01: field clock<br>Other values are reserved                                                                                                                                                                                                                             |
| 0x2008           | tcd_lst        | R/W | ARRAY      | ARRAY of type UINT8<br>TCD command for the corresponding clock<br>Value range is specified in IEC 61158-6-28, Clause 4.                                                                                                                                                                                                                                            |
| 0x2009 to 0x208F | reserved       |     |            | Protocol reserved                                                                                                                                                                                                                                                                                                                                                  |
| 0x2090 to 0x20FF | reserved       |     |            | Reserve for equipment manufacturer                                                                                                                                                                                                                                                                                                                                 |

### 6.5.3 Network management information table

**Table 13 – Network management information table**

| Parameter index  | Parameter name                  | RW  | Data Type | Description                                                                                                                                                                                                                                            |
|------------------|---------------------------------|-----|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x2100           | network_id                      | R/W | UINT32    | Indicate NETWORKID of node or module<br>See 6.6.6.1.2.                                                                                                                                                                                                 |
| 0x2101           | ip_switch                       | R/W | UINT8     | Whether supporting the IPv4 or the IPv6 represented by bit<br>Bit 0: Whether supporting IPv4<br>Bit 1: Whether supporting IPv6<br>The corresponding bit set to 0 indicates "not support" and set to 1 indicates "support".<br>Bit 2 to bit 7: reserved |
| 0x2102           | multicast_id0 to multicast_id15 | R/W | UINT32    | NETWORKID with group type indicates the multicast group to which the device or module belongs to.<br>Each device or module belongs to a maximum of 16 multicast groups.                                                                                |
| 0x2103           | mac_addr                        | RW  | UINT64    | Unique MAC address of the device or module<br>Value range: 0 to $(2^{48} - 1)$<br>Other values are reserved.                                                                                                                                           |
| 0x2104           | nodeid                          | RW  | UINT8     | Node ID address of the device<br>Value range: 0 to 255<br>See IEC 61158-4-28, Clause 4.                                                                                                                                                                |
| 0x2105           | dctime                          | RW  | UINT16    | Data calculation cycle time, unit is $\mu$ s<br>Value range: 0 to 65 535                                                                                                                                                                               |
| 0x2106           | bctime                          | RW  | UINT16    | Bus service cycle time, unit is $\mu$ s<br>Value range: 0 to 65 535                                                                                                                                                                                    |
| 0x2107           | wctime                          | RW  | UINT16    | Bus service data writing cycle time, unit is $\mu$ s<br>Value range: 0 to 65 535                                                                                                                                                                       |
| 0x2108           | rctime                          | RW  | UINT16    | Bus service data reading cycle time, unit is $\mu$ s<br>Value range: 0 to 65 535                                                                                                                                                                       |
| 0x2109           | cctime                          | RW  | UINT16    | Bus service competition cycle time, unit is $\mu$ s<br>Value range: 0 to 65 535                                                                                                                                                                        |
| 0x210A           | totcnt                          | RW  | UINT16    | Number of timeouts allowed by device or module<br>Value range: 0 to 65 535                                                                                                                                                                             |
| 0x210B           | rt_data_addr                    | RO  | UINT32    | Real-time data buffer start address<br>Value range: 0x00000000 to 0xFFFFFFFF                                                                                                                                                                           |
| 0x210C           | rt_data_size                    | RW  | UINT32    | Real-time data buffer size<br>Value range: 0x00000000 to 0xFFFFFFFF                                                                                                                                                                                    |
| 0x210D           | nrt_data_addr                   | RW  | UINT32    | Non-real-time data buffer start address<br>Value range: 0x00000000 to 0xFFFFFFFF                                                                                                                                                                       |
| 0x210E           | nrt_data_size                   | RW  | UINT32    | Non-real-time data buffer size<br>Value range: 0x00000000 to 0xFFFFFFFF                                                                                                                                                                                |
| 0x210F to 0x28FF | reserved                        |     |           | Protocol reserved                                                                                                                                                                                                                                      |
| 0x2900 to 0x2FFF | reserved                        |     |           | Reserved for equipment manufacturer                                                                                                                                                                                                                    |

## 6.6 ASEs

### 6.6.1 Overview

Type 28 AL provides users with time service, data service and system management service based on the DLL service interface. The AL user addresses and accesses devices or modules in the Type 28 network through NETWORKID. The time AP provides time synchronization and time query service functions through time ASE. The data AP provides reading and writing operations of RT data and nRT data through RT Data ASE and nRT Data ASE. The system management AP provides application service functions of system such as resource management, address management, network system management, virtualization management through resource ASE, addressing ASE, management ASE and virtualization ASE. These application entities implement specific data communication transmission services through C/S or P/S communication models. Table 14 shows the description of the communication model used by Type 28 AL ASEs.

**Table 14 – Communication models supported by Type 28 AL ASEs**

| ASE                  | C/S         | P/S         |
|----------------------|-------------|-------------|
| RT Data cyclic       | Not support | Support     |
| RT Data acyclic      | Support     | Support     |
| nRT Data             | Support     | Support     |
| Time synchronization | Not support | Support     |
| Time query           | Support     | Not support |
| Resource             | Support     | Support     |
| Addressing           | Support     | Not support |
| Management           | Support     | Support     |
| Virtualization       | Support     | Not support |

The return value of the AL service is shown in Table 15. The AL user determines whether the service is successful and the reason for the failure according to the return value.

**Table 15 – Application service interface return value list**

| Wrong name         | Wrong coding | Wrong description                                         |
|--------------------|--------------|-----------------------------------------------------------|
| SUCCESS            | 0x00000000   | Service processing succeeded                              |
| FAILED             | 0x00000001   | Service processing failed                                 |
| ERR_NOTSUPPORT     | 0x00000002   | Device or module does not support                         |
| ERR_NOTINITIALIZED | 0x00000003   | Device or module has not been initialized                 |
| ERR_NOTEXIST       | 0x00000004   | Device or module does not exist, service is unavailable   |
| ERR_NOTREADY       | 0x00000005   | Device initialization finishes but no valid status is set |
| ERR_INVALIDPARA    | 0x00000006   | Invalid parameter information                             |
| ERR_UNKNOWN        | 0x00000007   | Unknown error                                             |
| ERR_TIMEOUT        | 0x00000008   | Service processing timeout                                |

## 6.6.2 RT data ASE

### 6.6.2.1 General

RT data ASE defines service interfaces and interface parameters that support RT application services. The RT application service processes the RT data in cyclic or acyclic manner. The RTC data is cyclically read or written according to the specified cycle time. The processing of RTA data does not have a fixed cycle time, but it should be completed within the specified time when once occurred.

The RTC data is transmitted based on the P/S communication model, and the RTA data is transmitted based on the P/S communication model or the C/S communication model. Before processing the RT application service, the user needs to pre-configure the channel resource based on the resource ASE, and the channel resource will be occupied by the user all the time. If the channel resource for the RT data transmission is changed, the system should be restarted to meet the latest requirement of RT data service.

### 6.6.2.2 Interface

ALS-RT-DATA-Proc (

Dst\_NetworkID,  
Src\_NetworkID,  
ProcessID,  
DataID,  
OptionRTLst,  
ALS\_Result

)

Table 16 defines RT Data service interface parameters.

**Table 16 – RT data service interface parameters**

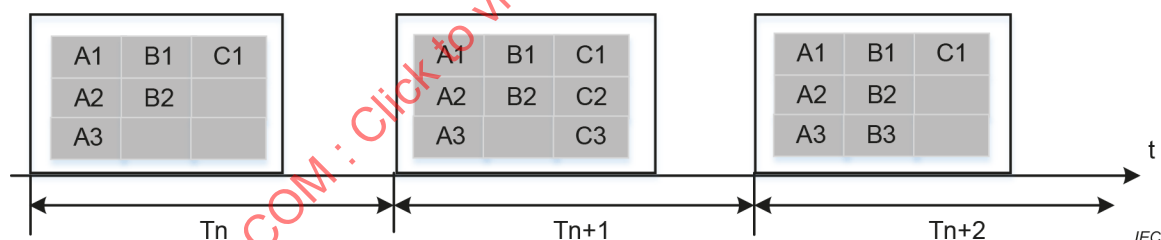
| Parameter name | Data type | Input/output |
|----------------|-----------|--------------|
| Dst_NetworkID  | UINT32    | IN           |
| Src_NetworkID  | UINT32    | IN           |
| ProcessID      | UINT16    | IN           |
| DataID         | UINT32    | IN OUT       |
| OptionRTLst    | STRUCT    | IN OUT       |
| ALS_Result     | UINT32    | OUT          |

- Dst\_NetworkID: NETWORKID of the destination device or module.
- Src\_NetworkID: NETWORKID of the source device or module.
- ProcessID: RT data processing mode ID, which identifies the RT data processing method based on bits and if the corresponding bit position is 1, execute the corresponding processing, defined as following:
  - Update (bit 0): Updates data after comparing the content of the data based on the bit value;
  - Reset (bit 1): Restores the data in data area content to the default value;
  - Clear (bit 2): Clears the data of the specified data area and set the zero operation;
  - Read (bit 3): Reads the data of the specified data area, and does not clear the corresponding data content;
  - Read clear (bit 4): Reads the data of the specified data area, and clear the data of the corresponding data area;
  - Write (bit 5): Writes data directly to the specified data area, and overwrites the original data;

- Write clear (bit 6): Clears the operation for the specified data area, and then write new data;
  - Query (bit 7): Queries data status;
  - Request (bit 8): Requests data;
  - Response (bit 9): Responds to the request, response data;
  - Confirmation (bit 10): Confirms the data transmission status;
  - Send (bit 11): Sends data to the specified destination device or module;
  - Receive (bit 12): Receives data sent by a specified source device or module;
- DataID: Service data encoding ID, unique ID of the specific service data. The ID can be associated with the data area corresponding to the data buffer. For RTC data and RTA data, the corresponding DataID is different.
  - OptionRTLst: A list of RT data processing information options, including the status of the data information after processing, as described below:
    - StateInfo: UINT16 type value, status information of the data area;
    - DataSize: UINT16 type value, the valid data size required for reading and writing;
    - RetCode: UINT32 type value, RT data information processing result encoding, the default value is 0;
  - ALS\_Result: The value that the RT data service returns, as specified in Table 15.

### 6.6.2.3 Interaction process

The processing of RT data is deterministic and timely. The determinacy means that the data should be processed at a certain time, while the timeliness means that the time required for data to be transmitted from the source device or module to the destination device or module is expected. RTC data needs to be cyclically processed in a fixed cycle, while RTA data does not need to be cyclically processed in the fixed cycle.



**Figure 11 – RT data transmission model**

As shown in Figure 11, the service data blocks A1, A2, A3, B1, B2 and C1 are always cyclically processed in each fixed period, like  $T_n$ ,  $T_{n+1}$ ,  $T_{n+2}$ , those are RTC data interactions. Service data blocks B3, C2 and C3 are not processed in every cycle. When transmission is required, RT processing is completed within a certain time period, that is RTA data interaction.

Since RTA data are usually control, configuration, alarms or exceptions information, they are not cyclically processed. See Annex C for an example of RTA service processing procedure. When the RTA data application service adopts the C/S communication model, the client application sends a request to the server. The server application acquires the request for the corresponding processing and sends the response data to the client application according to the requested content within a specified time period. The client application obtains the response data content sent by the server and confirms the completion of the interactive service.

### 6.6.3 nRT data ASE

#### 6.6.3.1 General

In the Type 28 network, for nRT application service data, time-critical is not necessary and the length of message is not fixed. Before the nRT data transmission is performed, the resource ASE is used to apply for the channel resource for transmission to the MN. The maximum times of this channel resources applied to the user should be specified.

nRT data ASE defines interfaces and related parameters for nRT data processing, also provides users with operations such as reading and writing nRT data. The user uses the resource ASE to apply for channel resources for nRT data processing, and the user needs to specify the channel resource information in nRT data processing.

When the nRT data AP interacts based on C/S communication model, the client application sends a request for reading or writing to the server application, and the server application performs response processing. When the nRT data AP interacts based on P/S communication model, the data is transmitted through the idle default channel resources allocated by the system. The system schedules the channel resources according to the pre-assigned nRT application service priority, and preferentially allocates the idle channel resources to nRT application service users with high priority.

#### 6.6.3.2 Interface

```
ALS-NRT-DATA-Proc (
    Dst_NetworkID,
    Src_NetworkID,
    ProcessID,
    DataID,
    OptionNRTLst,
    ALS_Result
)
```

Table 17 defines nRT data service interface parameters.

**Table 17 – nRT data service interface parameters**

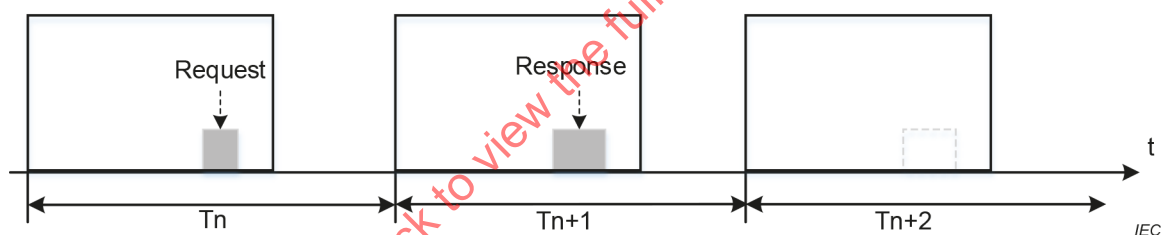
| Parameter name | Data type | Input/output |
|----------------|-----------|--------------|
| Dst_NetworkID  | UINT32    | IN           |
| Src_NetworkID  | UINT32    | IN           |
| ProcessID      | UINT16    | IN           |
| DataID         | UINT32    | IN           |
| OptionNRTLst   | STRUCT    | IN OUT       |
| ALS_Result     | UINT32    | OUT          |

- Dst\_NetworkID: NETWORKID of the destination device or module.
- Src\_NetworkID: NETWORKID of the source device or module.
- ProcessID: nRT data processing mode ID. The data processing mode is represented by bit, wherein the bit 15 position 1 indicates that the data is transmitted based on the TCP/IP protocol stack, and the IP information is obtained by the NETWORKID. The bit 15 position 0 indicates that the processing is based on the Type 28 DLL layer, and see the description of the ProcessID described in Subclause 6.6.2.2 for the description of remaining bit.
- DataID: Service data ID, which is associated with the corresponding address of the data buffer. For the nRT data corresponding to the service interface, you can choose not to associate with the buffer. In this case, the ID is invalid.
- OptionNRTLst: A list of option parameters, including the following information:

- Opt\_Cmd: UINT8 type value, optional command word, which identifies whether the service needs confirmation; 0x00: the service does not need to be confirmed; 0x01: the service needs to be confirmed;
  - DataSize: UINT16 type value, the effective data size that needs to be read and written;
  - DataAddr: UINT32 type data storage area address, this value is valid when the DataID is invalid;
  - Data\_Flag: UINT8 type value, which identifies the release operation after reading and writing of nRT data block; 0x00: no processing; 0x01: clear release; 0x02: no clear release;
- ALS\_Result: The nRT data service processes the return value. See Table 15 for the specific return values.

### 6.6.3.3 Interaction process

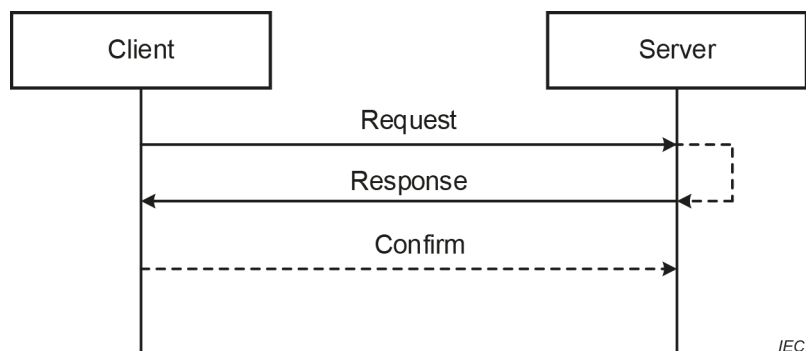
In the Type 28 network, when the nRT data AP interaction uses the C/S communication model, the nRT data service interface provides confirmed mode and non-confirmed mode to meet the requirement of different services. In the confirmed mode, the client sends a confirmed message to the server after correctly receiving and processing the response message sent by the server. In the non-confirmed mode, the client should not send the confirmed message to the server after correctly receiving and processing the response message sent by the server. In the process of implementing the nRT data application service by using the C/S communication model, after the client application initiates the request, it waits for the asynchronous response of the server application, and the asynchronous interval time period is usually one or more bus cycles as shown in Figure 12. The time period of the interval is related to the configured bus cycle and the specific size of the nRT data that needs to be transmitted.



**Figure 12 – Diagram of nRT data request response model**

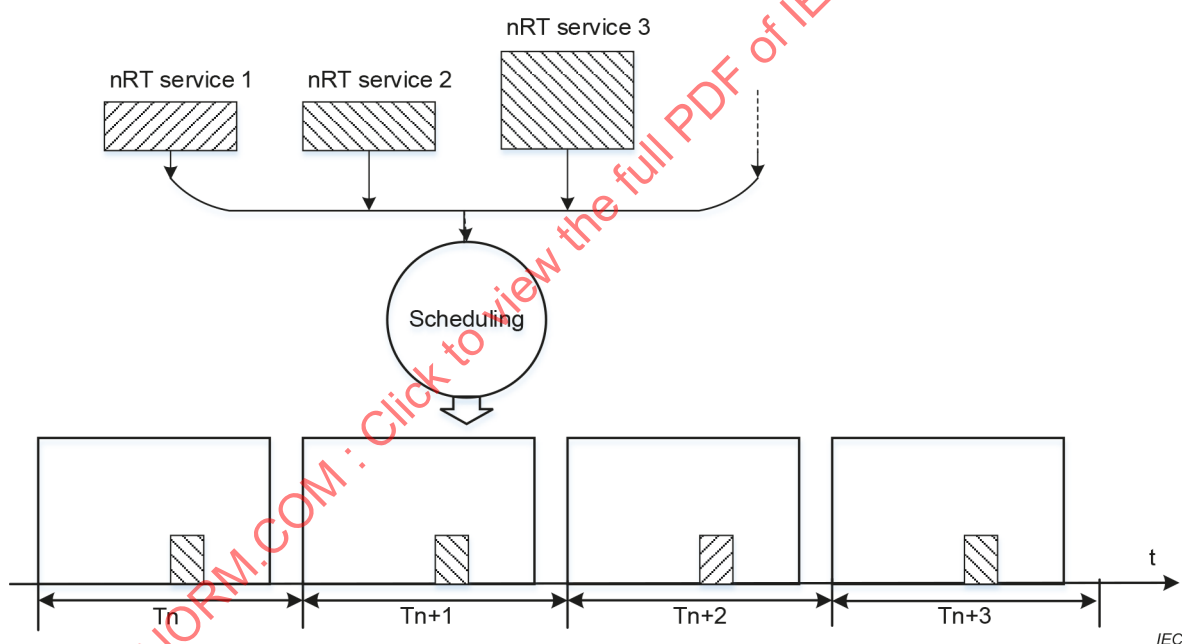
When the nRT data AP interaction adopts the C/S communication model, the client application initiates one nRT data service request. The server application responds to the client application request, and completes the data processing locally or sends the response data to the remote client application. According to the different specific content of the client application request, the data transmission service based on the request and response is continuously completed between the server application and the client application.

Figure 13 shows the nRT data AP interaction based on C/S communication model.



**Figure 13 – Diagram of nRT data AP interaction based on C/S communication model**

When nRT data AP interaction adopts P/S model, it will schedule channel resources according to the priority of nRT data application service and send nRT data through the idle default channel. According to the features of nRT data, nRT data application service users will use different default channels and occupancy periods. A diagram of nRT data AP interaction based on P/S communication model is shown in Figure 14, where both the nRT service 1 and the nRT service 2 can use one default channel for data transmission, and the nRT service 3 has higher priority and needs to use default channel twice to complete the corresponding data transmission.



**Figure 14 – Diagram of nRT data AP interaction based on P/S communication model**

#### 6.6.4 Time ASE

##### 6.6.4.1 General

The time ASE defines the interface and parameters of the time service. The devices and modules in the Type 28 network implement the time synchronization and time query service functions based on the time ASE.

On the network, the AL user triggers the time synchronization service at the MN. The MN will send message including the transmission delay time information between TN and all TNs, and then the TN will parse the delay time information from the message sent by the MN, and set the time of the station to complete synchronization.

The time query service provides a query for the synchronizing state of TN's time. The object of the query service on MN is any node in the network but the node itself on TN.

#### 6.6.4.2 Interface

ALS-TIMING-Proc (

Dst\_NetworkID,  
Src\_NetworkID,  
ProcessID,  
OptionClkInfo,  
TimeInfo,  
ALS\_Result

)

Table 18 defines timing service interface parameters.

**Table 18 – Timing service interface parameters**

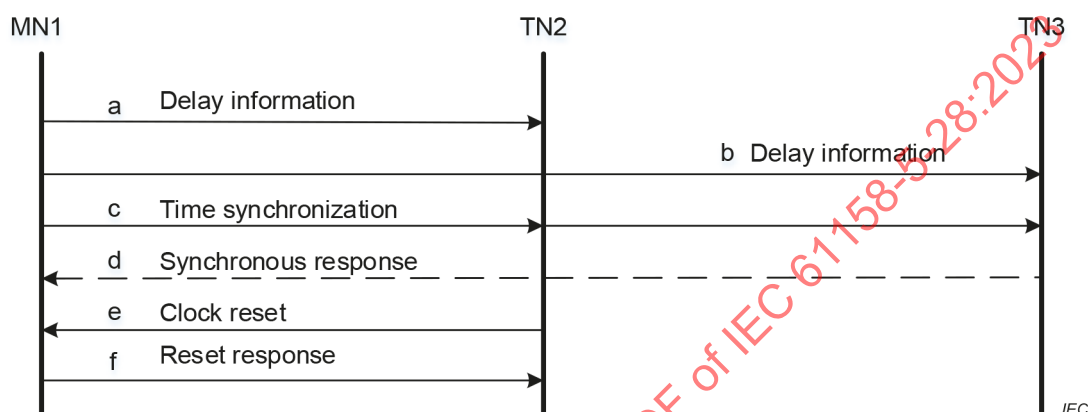
| Parameter name | Data type | Input/output |
|----------------|-----------|--------------|
| Dst_NetworkID  | UINT32    | IN           |
| Src_NetworkID  | UINT32    | IN           |
| ProcessID      | UINT16    | IN           |
| OptionClkInfo  | STRUCT    | IN OUT       |
| TimeInfo       | STRUCT    | IN OUT       |
| ALS_Result     | UINT32    | OUT          |

- Dst\_NetworkID: NETWORKID of the destination device or module.
- Src\_NetworkID: NETWORKID of the source device or module.
- ProcessID: Time Service Processing ID:
  - 0x0000: Time synchronization request;
  - 0x0001: Time synchronization response;
  - 0x0002: Time synchronization exception;
  - 0x0004: Time query request;
  - 0x0008: Time query response;
  - 0x0010: Time query error.
- OptionClkInfo: Time option information. When implementing time synchronization service, this parameter is used as an input parameter, and specifies clock device information list that needs to be set. When this parameter is used as an output parameter, it shall specify the set output information. If the setting is successful, it will output the current state, current value and current attribute of the output time; if the setting fails, the corresponding error state code is output. See Table 7 for the description of the structure CLOCK\_OPTION\_INFO\_S. When the time query service function is implemented, this parameter is an output parameter.
- TimeInfo: Time information that needs to be set for the time. See Table 6 for the description of the structure TIMEINFO\_S.
- ALS\_Result: Time Service processing return value. See Table 15 for details of the return value.

### 6.6.4.3 Interaction process

#### 6.6.4.3.1 Time synchronization interaction process

The MN sends its own time information and the measured transmission delay between the corresponding nodes to each TN, and the TN synchronizes time and updates the local time information based on the time information of MN. The time synchronization interaction process adopts the P/S communication model, and the time application of the MN acts as a publisher and periodically sends time information to the time application of the TN, but the release period is set by the MN. If a synchronization failure or error occurs in the client time application, a time synchronization reset operation can be requested from the MN time application.



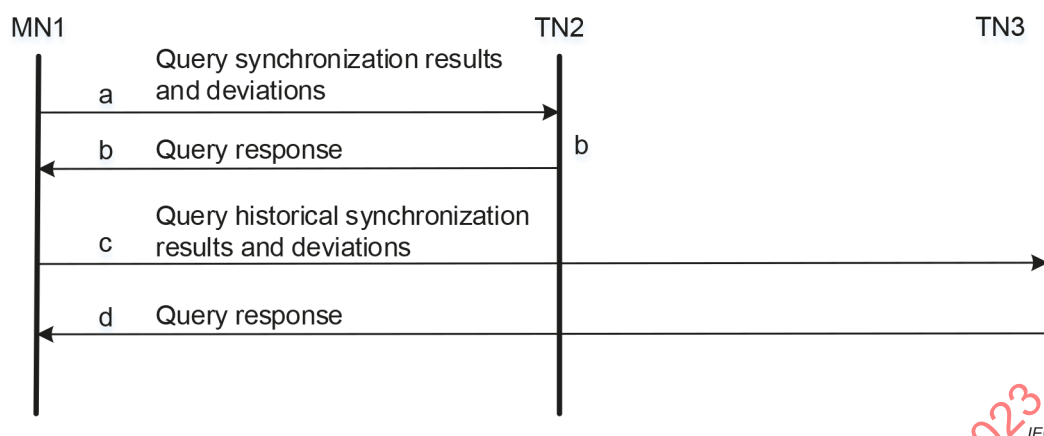
**Figure 15 – Time synchronization application interaction process**

See Figure 15. Time synchronization is initiated by the MN:

- The MN sends the transmission delay value of corresponding MN to the TN.
- The MN initiates time synchronization, sends the time value to the TN through the time service notification message, and requests the TN to respond to the synchronization result.
- After receiving the time information in the time service notification message sent by the MN, the TN implements the local time synchronization through the local DLL interface, and determines whether to respond to the synchronization result according to the requirements of the MN.
- If an error occurs on the TN during the time synchronization, the response error message shall be transmitted to the MN.
- If an error occurs on the TN during the time synchronization, the TN can actively send a time reset request to the MN, and ask the MN to update the time information and the line transmission delay information. At the same time the MN should update the time information and line transmission delay information to the designated TN.
- The MN can select periodic time synchronization or single time synchronization operation according to system requirements and configuration.

#### 6.6.4.3.2 Time query interaction process

The time query service provides the function of querying the time synchronization state and synchronization quality of the TN. The time query service uses the C/S communication model to interact. The time application of the MN acts as a client and requests time synchronization state and synchronization quality information from time application at the TN as a server. The MN sends a time service notification message to trigger the time query service. For the procedure, see Figure 16.



**Figure 16 – Time query process**

See Figure 16. The time query is initiated by the MN:

- The MN queries the time of the last synchronization of the corresponding TN and the deviation value during synchronization.
- After receiving the query, the TN replies to the local record information of the MN, and the MN determines the quality and deviation range of the synchronization of the TN based on this.
- The MN can query other terminal information of corresponding TN, such as the average value, root mean square value, time span, etc. of deviation values of the synchronization.
- The TN replies to the corresponding content of the MN based on the content of the query.

## 6.6.5 Resource ASE

### 6.6.5.1 General

The resource ASE of the Type 28 AL protocol defines the service interface definition and parameter settings of the channel resources required for AP interaction on the network. The resource ASE performs channel resource allocation and usage based on the service interface provided by the Type 28 DLL, and thus implements the operations of applying, establishing, updating, and releasing channel resources.

Channel resources include fixed bandwidth channel resources and variable bandwidth channel resources. The fixed bandwidth channel resources are created during Type 28 configuration or network initialization, and cannot be updated and released afterwards; variable bandwidth channel resources are created and applied by users when needed. The size of variable bandwidth channel resources and resources usage time is determined by the user at the time of application, and users can dynamically update and release the resource channel during use. The MN is responsible for the allocation and management of all channel resources of the Type 28 network.

The resource ASE implements the AP interaction by using the confirmed C/S communication model. When the resource AP interacts, the server application is always at the MN, while the client application can be at the MN or the TN. The client application sends a resource application request, and server application responds to the resource application request, sends the response data, allocates the resource channel or rejects the resource application request, while the client application acknowledges receipt of the data and sends a confirmation message to the server application, and then completes the local resource AP interaction.

### 6.6.5.2 Interface

```
ALS-RESOURCE-Proc (
    Dst_NetworkID,
    Src_NetworkID,
    ProcessID,
    OptionResLst,
    ALS_Result
)
```

Table 19 defines resource service interface parameters.

**Table 19 – Resource service interface parameters**

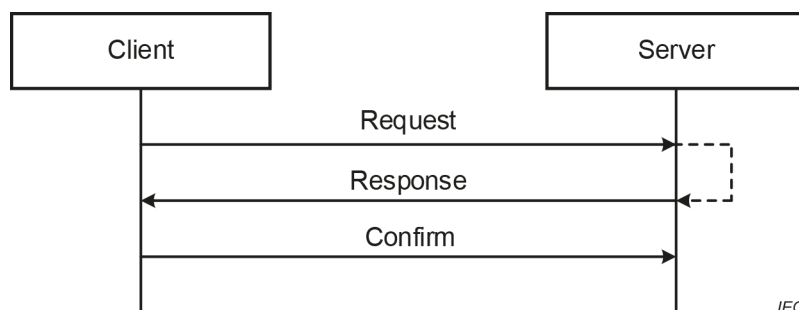
| Parameter name | Data type | Input/output |
|----------------|-----------|--------------|
| Dst_NetworkID  | UINT32    | IN           |
| Src_NetworkID  | UINT32    | IN           |
| ProcessID      | UINT16    | IN           |
| OptionResLst   | STRUCT    | IN OUT       |
| ALS_Result     | UINT32    | OUT          |

- Dst\_NetworkID: NETWORKID of destination device or module.
- Src\_NetworkID: NETWORKID of source device or module.
- ProcessID: Resource service process ID which is defined by bit, as follows:
  - Application (bit 0): Apply for resources
  - Release (bit 1): Release resources
  - Update (bit 2): Update Resources
  - Confirm (bit 3): Confirm Resources
  - Write (bit 4): Write resource configuration
  - Read (bit 5): Read resource configuration
  - Reject (bit 6): Reject resource service
  - Assignment (bit 7): Assign Resource Configuration.
- OptionResLst: Resource application related option information list, including the following information:
  - ResType: UINT8 type, resource type; 0x00: fixed bandwidth resource; 0x01: variable bandwidth resource;
  - UsingCnt: UINT16 type, the number of times the user continuously uses the channel resource. The default value is 1;
  - BWidth: UINT16 type, bandwidth resource size, the unit is octet;
  - LoopTime: UINT16 type, the period of channel resource usage;
  - ResourceID: UINT32 type, ID information of the resource.
- ALS\_Result: return value of resource service. See Table 15 for details of return value.

### 6.6.5.3 Interaction process

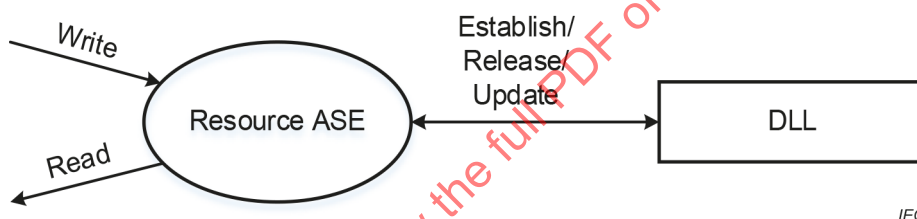
The resource AP interaction is based on the C/S communication model in Type 28 network as shown in Figure 17. The client application requests for bandwidth resources to the server application, and then the server application responds to the request and sends the allocable resources to the client application. The client application acknowledges receipt of the bandwidth resource allocation, sends an acknowledgment message to the server application, and finishes the AP interaction of the resource AP. If the server application fails to allocate bandwidth resources, the client application determines whether it needs to repeatedly apply for bandwidth

resources by parsing the response message content of the server; if the client application fails to send the confirmation message or the server application and does not receive the confirmation message from the client application, the server application automatically reclaims resources that have been allocated to the client application.



**Figure 17 – Resource AP interaction based on C/S communication model**

Resource ASE provides the function of writing and reading services based on the DLL service interface to implement local bandwidth resource allocation. AL user can use Write or Read service function to trigger the resource application as shown in Figure 18. The corresponding channel resource can be established, updated or released in DLL layer.



**Figure 18 – Resource ASE local service function diagram**

## 6.6.6 Addressing ASE

### 6.6.6.1 General

#### 6.6.6.1.1 Overview

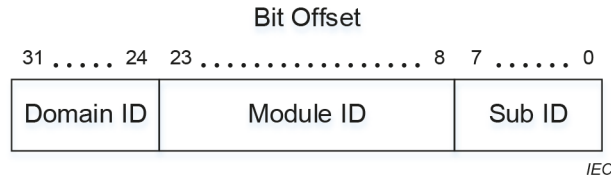
The Type 28 AL user identifies the device or module by NETWORKID. The AL user can also address the device and module based on IP. In the Type 28 network device, there is MAC mapping table, which saves the mapping relationship between NETWORKID and MAC of device or module (MAC uniquely corresponds to the DLL node address NodeID), as well as the mapping relationship between NETWORKID and the IP of device or module stored in IP mapping table. The Type 28 AL user implements the interoperation between the Type 28 network and IP network based on the processing of the mapping tables.

The addressing ASE defines a service interface and parameters based on the NETWORKID, and the AP sets and acquires the NETWORKID of the device or module where the interactive object is located based on the addressing ASE. Addressing ASE provides MAC mapping tables and IP mapping tables with maintenance service functions such as adding, deleting, modifying and querying to meet AP interaction requirements based on IP in the Type 28 network environment.

#### 6.6.6.1.2 NETWORKID

NETWORKID is a 4-octets value (see Figure 19). The Type 28 AL protocol specification supports up to eight virtual bus domains. The NETWORKID value is unique for the device or module identified in a specified domain, and the Module ID is unicast generally. When the

Module ID is a multicast type, it indicates a group of devices or module. Each device or module in the specified domain belongs to a maximum of 16 multicast groups.



**Figure 19 – NETWORKID identification field structure**

- Domain ID: Domain ID, 8 bits, identifies the bus domain to which the corresponding node belongs. This field has the following defined values:
  - 0x00: The device does not belong to any bus domain;
  - 0x01 to 0x08: Identifies the domain ID of a bus domain;
  - 0x09 to 0xFE: The domain ID range is reserved;
  - 0xFF broadcast type domain ID, identifying all bus domains.
- Module ID: Module ID, 16 bits, identifies the number of the device or module; for this field, the following defined values exist:
  - 0x0000: Initialization ID of this device or this module;
  - 0x0001 to 0xDFFF: Identifies valid ID of device or module;
  - 0xE000 to 0xE0FF: Identifies the multicast group number to which the device or module belongs in this bus domain;
  - 0xFFFF: ID of the broadcast type, identifying all devices or modules in the domain.
- Sub ID: sub-module ID, 8 bits, identifies the sub-device or sub-module number. Valid value in range from 0x00 to 0xFE. If the device or module does not have a molecular device or sub-module, the field is 0x00. When the field is 0xFF, it refers to all sub-devices and sub-modules.

A mapping table of NETWORKID and MAC (hereinafter referred to as MAC mapping table) exists on each device or module in Type 28 network. See Table 9 for details of the mapping table. The entry is created when the device or module is initialized. When the network is initialized and the device or module dynamically joins the network, the corresponding entry is created or updated.

#### 6.6.6.1.3 IP address mapping service

For the Type 28 network device, the mapping between the NETWORKID and the IPv4 or IPv6 address is implemented by maintaining the IP address mapping table, to meet the interoperation requirements between the Type 28 network and the IP network. For the contents of this mapping table, see Table 10. NETWORKID in the entry is a keyword. If the device or module is not assigned a NETWORKID, the corresponding entry should not be created. In the mapping table, all nodes in the network should be maintained. In the MN, it shall maintain the whole entries of the entire network. Each TN only maintains its own entries. The methods for adding, deleting, and modifying the mapping table are as follows:

- a) When the release of static configuration is finished, the system creates a corresponding entry based on the configuration parameters, but status bit of the corresponding entry is invalid.
- b) The MN notifies the TN to start running, and the status bit of updated corresponding entry is valid.
- c) If there is a dynamically access node, when the node applies for a valid NETWORKID, it creates a table entry and invalidates the status bit. When the node starts to run officially, the corresponding entry status bit is valid.

- d) If there is a dynamically offline node, the status bit of the related entry is invalid. If the timeout period is expired, the corresponding entry should be deleted. If the corresponding node is online again, the valid status bit should be restored. If the corresponding MAC address or NETWORKID changes when it is online again, the corresponding entry should be deleted. After the entry is re-created, the status bit is reset.

Addressing ASE defines service interfaces and parameters based on NETWORKID, implements NETWORKID setting and acquisition services for devices or modules based on addressing ASE, and implements maintenance functions such as creation, deletion, and modification of MAC mapping tables and IP mapping tables based on NETWORKID.

#### 6.6.6.2 Interface

ALS-ADDRESSING-Proc (

Dst\_NetworkID,  
Src\_NetworkID,  
ProcessID,  
OptionParaLst,  
ALS\_Result

)

Table 20 defines the addressing service interface parameters.

**Table 20 – Addressing service interface parameters**

| Parameter name | Data type | Input/output |
|----------------|-----------|--------------|
| Dst_NetworkID  | UINT32    | IN           |
| Src_NetworkID  | UINT32    | IN           |
| ProcessID      | UINT16    | IN           |
| OptionParaLst  | STRUCT    | IN OUT       |
| ALS_Result     | UINT32    | OUT          |

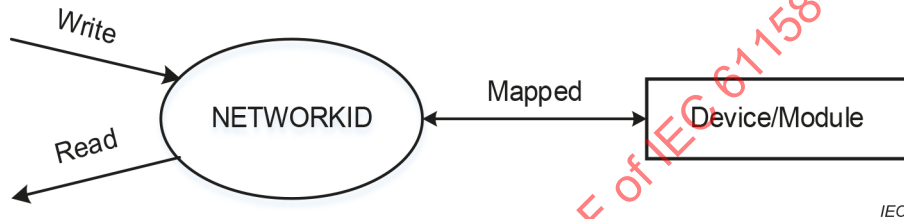
- Dst\_NetworkID: NETWORKID of the destination device or module.
- Src\_NetworkID: NETWORKID of the source device or module.
- ProcessID: service processing ID, including the setting and acquisition of device or module's NETWORKID, and the operation of the MAC mapping table and the IP mapping table, as follows:
  - 0x0000: Set NETWORKID;
  - 0x0001: Get NETWORKID;
  - 0x0002: Add a MAC mapping table entry;
  - 0x0003: Delete the MAC mapping table entry;
  - 0x0004: Modify the MAC mapping table entry;
  - 0x0005: Query MAC mapping table entries;
  - 0x0006: Add an IP mapping table entry;
  - 0x0007: Delete the IP mapping table entry;
  - 0x0008: Modify the IP mapping table entry;
  - 0x0009: Query the IP mapping table entry.
- OptionParaLst: list of parameter information related to the addressing service. The specific content of the parameter is different according to different processing IDs:
  - Para\_Cnt\_In: UINT8 type value, an entry for setting or getting the configuration parameter;
  - Para\_Cnt\_out: UINT8 type value, the configuration data entry actually obtained;

- NetworkCfgPara: list of configuration parameter information. See Table 8 for details about related configuration parameters;
  - AddressingState: Address the status of the service node, 0 is normal, and other values correspond to error codes;
  - DIDIPMapTable: Data structure information corresponding to the mapping table. see Table 10 for related member description;
- ALS\_Result: Address service return value, and see Table 15 for details of return value.

### 6.6.6.3 Interaction process

#### 6.6.6.3.1 NETWORKID interaction process

The addressing ASE implements a Write service function and a Read service function for the NETWORKID of the device or module based on the C/S communication model. The NETWORKID is mapped to a determined register or entry on the device or module. Figure 20 shows the NETWORKID interaction process diagram.

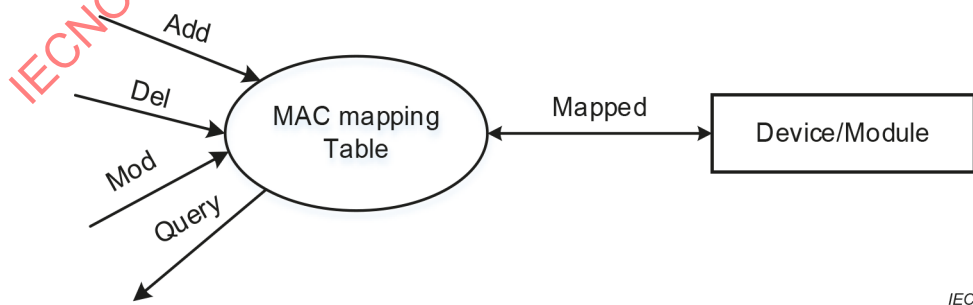


**Figure 20 – NETWORKID interaction process diagram**

During the initialization and operation of the Type 28 network, the client application initiates a request of setting or getting NETWORKID service, and server application responds to the client application request to perform a corresponding write and read operation, and returns the result of the operation status to the client application; If the operation fails, the corresponding error code and status value will be contained in the returned result.

#### 6.6.6.3.2 MAC mapping table interaction process

The C/S communication model is used in the Type 28 network to implement related operations on the MAC mapping table. The client application triggers an operation request for the MAC mapping table based on the addressed ASE interface, and the server application receives the request and implements corresponding processing of the specified MAC mapping table entry on the device or module as shown in Figure 21.

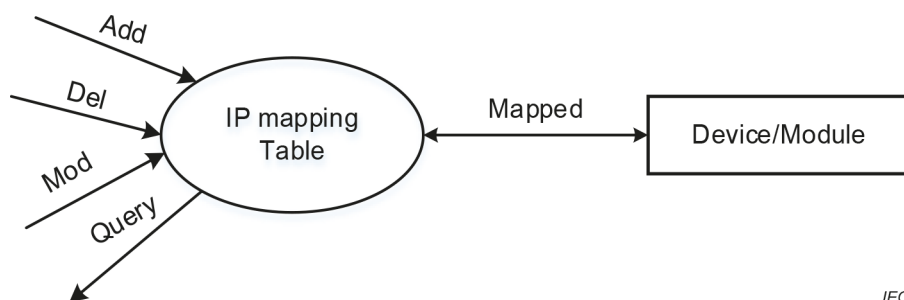


**Figure 21 – Diagram of MAC mapping table service function**

#### 6.6.6.3.3 IP and NETWORKID interaction process

In the Type 28 network, the C/S communication model is used to implement the operation of the IP mapping table. The client application triggers the addition, deletion, modification and query service request of the IP mapping table based on the addressed ASE interface, and the

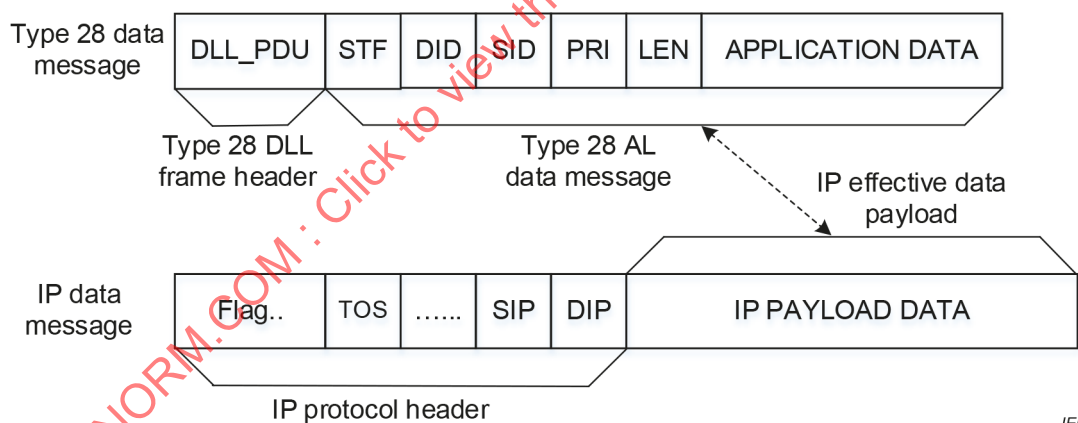
server application receives the request, and implements corresponding processing of the specified IP mapping entries on the device or module as shown in Figure 22.



**Figure 22 – IP mapping table service function diagram**

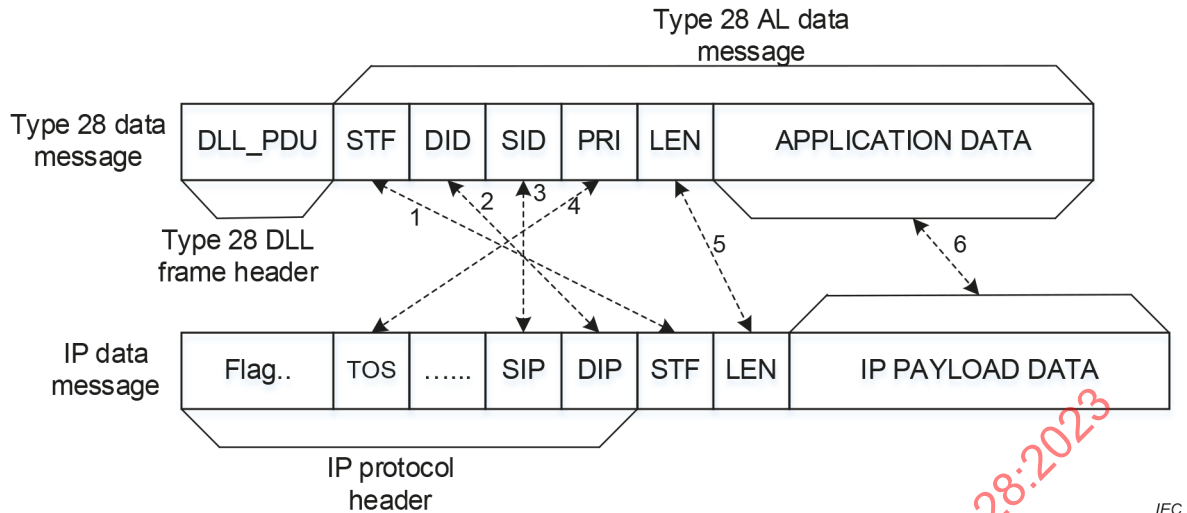
Based on the IP address mapping table, the data intercommunication between the Type 28 network and the IP network can be realized. There are three ways to convert data messages between AL of Type 28 network and IP network:

- a) The Type 28 AL data message is directly transmitted as the effective payload part of the IP data message. The IP-based address mapping table is used to determine the Type 28 AL data message and the IP protocol message to be converted, and the Type 28 AL data message is directly encapsulated into the IP protocol data message as the effective payload of the IP protocol, or the effective payload data in the IP protocol data message is directly extracted and transmitted as a Type 28 AL data message. In Figure 23, this data transmission method is simple and reliable, but this conversion method will completely expose the Type 28 AL information to IP users, which has potential security risks and needs to carry redundant data and consume more bandwidth resources.



**Figure 23 – Type 28 AL data message mapped to IP payload data payload**

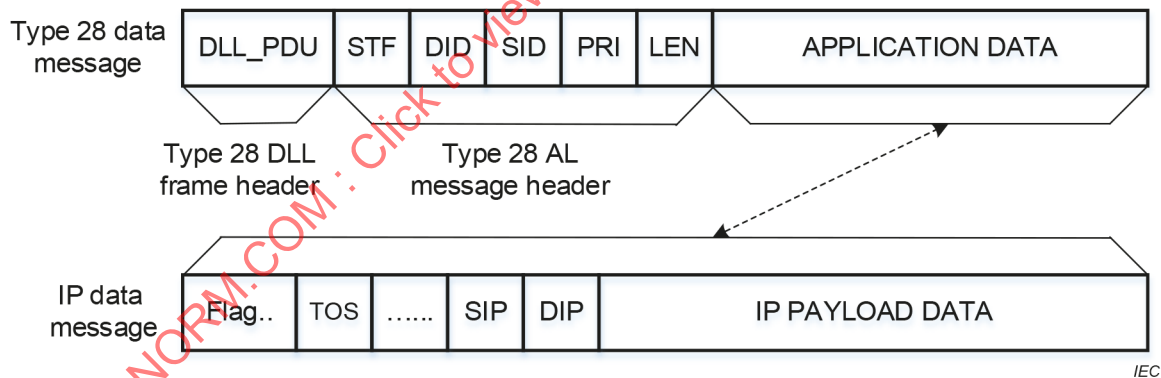
- b) Converting the Type 28 AL protocol message header into IP protocol message header data, and then transmitting the valid data payload in the Type 28 AL data packet as the payload of the IP message. In Figure 24, based on IP, the mapping table implements mapping conversion between the SID and DID, and PRI fields in the Type 28 protocol header, and the SIP, DIP, and TOS fields in the IP protocol header (see 2, 3 and 4 in Figure 24), and retains the Type 28 AL protocol. The type field is unchanged; the priority and length fields are compressed and placed in the IP protocol header and IP option fields (see 1 and 5 in Figure 24). The effective data payload in the Type 28 AL protocol message should be transmitted as IP payload data (see 6 in Figure 24). This method consumes more computing resources when converting mappings, but can effectively utilize bandwidth resources.



IEC

**Figure 24 – Type 28 AL protocol header and IP protocol header mapping**

- c) When the IP data message is carried in the Type 28 network, the entire IP data message is transmitted as part of the effective data payload of the Type 28 AL data message. For the non-IP related AP interaction, the transmission of the IP data message by the Type 28 network does not care about the specific content of the IP data message header. The entire IP data message is encapsulated and transmitted to the peer terminal, and then it is processed after the parsing is completed. For the IP related AP interaction, the Type 28 network needs to parse the IP protocol header before transmitting the IP data messages, determine the corresponding Type 28 AL protocol message header information based on the IP mapping table, and then encapsulate the data into effective data payload for transmission. Figure 25 shows how an IP data message is mapped to valid data of Type 28 AL data message.



IEC

**Figure 25 – IP data message is mapped to valid data of Type 28 AL data message**

## 6.6.7 Management ASE

### 6.6.7.1 General

#### 6.6.7.1.1 Overview

The management ASE provides network parameter configuration and acquisition based on NETWORKID, as well as service functions such as diagnostic maintenance and log management for the network. For network configuration parameters and log files, the management ASE also provides corresponding file import or export service functions. The management ASE defines the interfaces and parameters of the above service functions.

### 6.6.7.1.2 Network

The Type 28 network configuration parameters shall be configured and confirmed before the network works normally. The network configuration parameters are sent to all devices or modules through the service interface. The MN is responsible for sending configuration parameters to all nodes and getting the network configuration parameters of specified devices or module in the network. The import and export service functions of the configuration file are also provided at the MN.

### 6.6.7.1.3 Diagnosis

For Type 28 application, the MN is responsible for diagnosing the operating status of all nodes and the entire network. The TN responds based on the diagnostic command and response objects are the remote MNs or local AL service users. The content of diagnosis includes related device status, module status, link status, and network running status, etc. The diagnostic object can be a specific device or module or current bus network domain, or can be a group of devices or modules.

### 6.6.7.1.4 Logging

The normal or abnormal information of the device or module during the running process is stored in the memory of the device or the module. The remote MN or local AL user obtains the stored log information through the log service. The log object entities in the form of files or data variables are stored on the device or module. The user program can set the update operation of the log object or event trigger mode periodically.

### 6.6.7.2 Interface

ALS-MANAGEMENT-Proc (

Dst\_NetworkID,  
Src\_NetworkID,  
ProcessID,  
OptionMgtInfoLst,  
NetworkCfgPara,  
ALS\_Result

)

Table 21 defines the management service interface parameters.

**Table 21 – Management service interface parameters**

| Parameter name   | Data type | Input/Output |
|------------------|-----------|--------------|
| Dst_NetworkID    | UINT32    | IN           |
| Src_NetworkID    | UINT32    | IN           |
| ProcessID        | UINT16    | IN           |
| OptionMgtInfoLst | STRUCT    | IN OUT       |
| NetworkCfgPara   | STRUCT    | IN           |
| ALS_Result       | UINT32    | OUT          |

- Dst\_NetworkID: NETWORKID of the destination device or module.
- Src\_NetworkID: NETWORKID of the source device or module.
- ProcessID: Manages service processing ID. Different ID values correspond to different processing operations:
  - 0x0000: Set network configuration parameters;
  - 0x0001: Get network configuration parameters;
  - 0x0002: Import the network configuration file;

- 0x0003: Export network configuration file;
  - 0x0004: Diagnose network operation status;
  - 0x0005: Diagnose node operation status;
  - 0x0006: Request to write log;
  - 0x0007: Request to get the log;
  - 0x0008: Log file export.
- OptionMgtInfoLst: Manages the service information list. Depending on the processing ID, the effective content of the option list is different, including the following information:
- Cfg\_Cnt\_In: UINT8 type value, the number of network devices that need to be set or got;
  - Cfg\_Cnt\_Out: UINT8 type value, the number of network devices are set or got actually;
  - Cfg\_Set\_Stat: UINT8 type value, set or get the status information of the network configuration parameters, the default value is 0x00, that is, the setting can always be successful, and if the setting fails, the corresponding error code is returned;
  - Node\_Cur\_State: UINT8 type value, the current running state value of the node;
  - Node\_Prev\_State: UINT8 type value, the previous running state value of the node;
  - Node\_Next\_State: UINT8 type value, the next running state value of the node;
  - NetRun\_State: UINT8 type array, network running status value;
  - Path\_Name: VISIBLESTRING type, the path name of the configuration file or log file;
  - File\_Name: VISIBLESTRING type, configuration file name with suffix name or log file name;
  - RuningCnt: UINT16 type value, the number of device nodes in the running state;
  - FailedCnt: UINT16 type value, number of device nodes in the failed state;
  - InitdCnt: UINT16 type value, the number of device nodes in the initialization state;
  - PausedCnt: UINT16 type value, the number of device nodes in the suspended state;
  - MaxTimeOut: UINT32 type value, maximum delay information of network;
  - LogType: UINT8 type value, log type.
- NetworkCfgPara: A list of configuration parameter information that needs to be set or obtained. If the MN is set for all nodes, information list contains configuration parameter information for all nodes. See Table 8 for detailed information.
- ALS\_Result: Manages the service return value. See Table 15 for the specific return value.

### 6.6.7.3 Interaction process

#### 6.6.7.3.1 Network configuration process

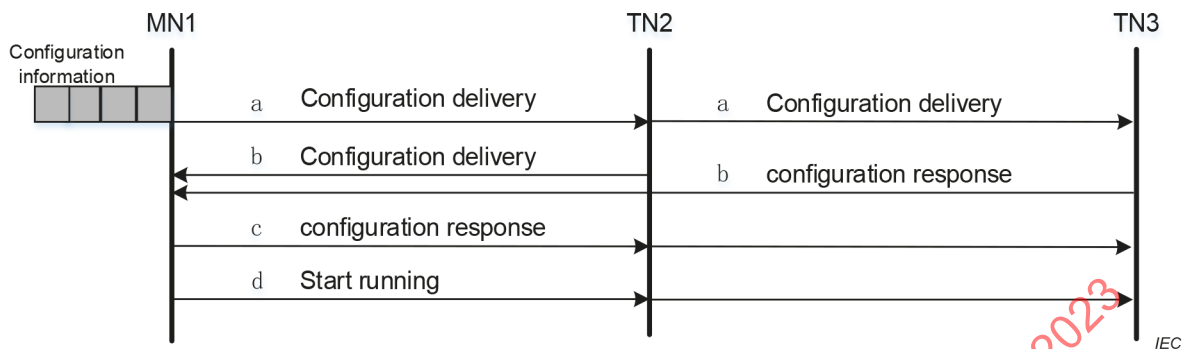
##### 6.6.7.3.1.1 General

The fieldbus network configuration parameters of the device or module are saved in the MN, where the configuration to all stations through static allocation is broadcast. All TNs obtain their own configuration information. For the normal running network, the configured TN is allowed to access or exit the network. The network configuration AP interaction adopts the C/S communication model. The system initialization static configuration phase always has a configuration request of the TN by default, and the server application sends the configuration to all client applications.

##### 6.6.7.3.1.2 Initialization process of static configuration

Before the network is running, network parameters such as NETWORKID need to be configured for all devices or modules in the network. The network completes the relevant configuration through static configuration before the normal operation. The MN sends network configuration

parameters and receives the TN response and confirmation of all the TNs to finish network initialization. See Figure 26 for details of process:



**Figure 26 – Static configuration initialization process**

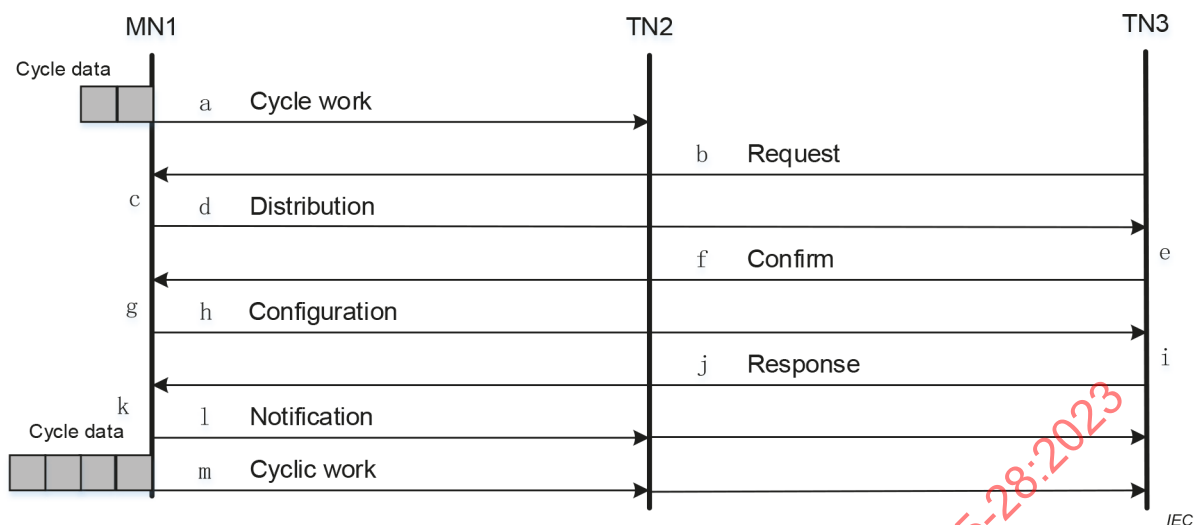
The static configuration initialization process assumes that the network configuration parameters of all nodes have been confirmed correctly, and the configuration file already exists on the MN storage area.

- After all node devices except Management Node 1 (MN1) device in the network are powered on and initialized, they wait for the configuration message. MN1 sends the configuration to all TNs and waits for confirmation.
- TNs receive the configuration parameters to complete the checksum configuration and deliver the configuration. All configured nodes should send the configuration response message to the MN1, and switch to the specified state based on the configuration parameter information.
- The MN1 confirms that the network configuration is successful, and sends a work notification message to all nodes and notifies all nodes to start operating normally.
- The network starts to work normally based on the set parameters.

The static configuration initialization process depends on whether the user-specified configuration parameters are reasonable and correct. If there are errors in the parameters set by the user, it will affect the initialization process and the normal operation of the network.

#### 6.6.7.3.1.3 Join the network interaction process

Figure 27 shows the specific processes for devices or modules joining a stable network dynamically.



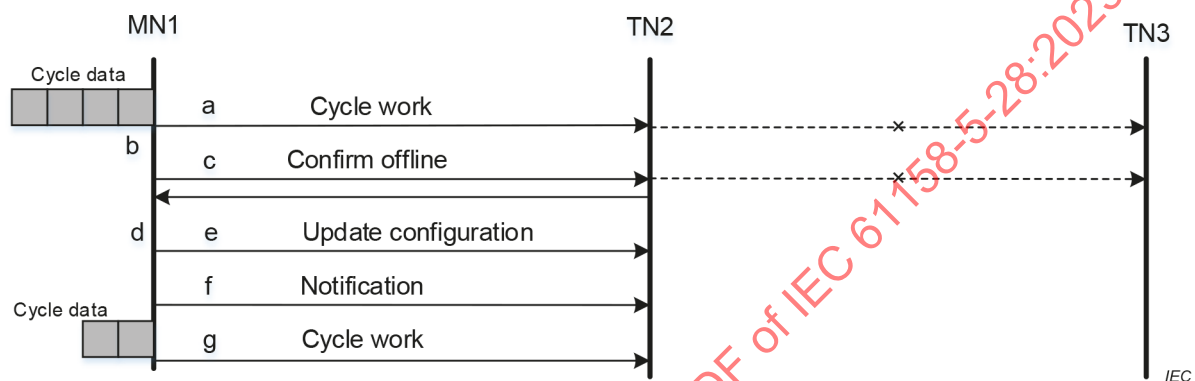
**Figure 27 – Join the network interaction process**

In Figure 27, before the Terminal Node (TN3) dynamically accesses the normal running Type 28 network, the effective configuration parameter information of TN3 is ready on the Management Node 1 (MN1).

- a) The bus is in normal working condition and the data is processed cyclically according to the specified cycle.
- b) After TN3 is powered on and initialized, the protocol stack starts to work, and periodically sends NETWORKID requesting broadcast message to the network until the message of the NETWORKID resource allocation of MN1 is received.
- c) After receiving the access request from TN3, MN1 completes the parameter checksum and assigns the determined NETWORKID to TN3.
- d) MN1 broadcasts the pre-assigned NETWORKID to the Type 28 network. Only the newly added TN3 should receive and parse the broadcast message, while other nodes that are in normal operation will ignore this message.
- e) TN3 receives the data message sent by MN1, and parses the assigned NETWORKID and sets the local node. If it fails, it should send the error result to MN1 and reapply.
- f) When TN3 is successfully set up, it will send a confirmed response data message to MN1, and apply for network configuration.
- g) MN1 receives the confirmation response of TN3, updates the local NETWORKID and MAC mapping table, confirms that the ID is used normally, and obtains the configuration parameter information of TN3 that joins dynamically, and prepares to send to TN3.
- h) MN1 sends the configuration parameter information of the dynamic joining node in broadcast mode. If the dynamically joining node makes related nodes update configuration information, they should update the configuration information synchronously. In addition, the command code is sent in the transmit data message to notify TN3 whether it needs to respond.
- i) After receiving the configuration, TN3 finishes the configuration data parsing and setting, and continues to finish the initialization process of the node until it is running.
- j) TN3 sends the confirmation response data message to the bus in broadcast mode. In addition to MN1, the relevant TNs also perform synchronization processing after receiving the confirmation message in the network.
- k) The MN1 receives the finished and confirmed response data message which means dynamically joining to the node configuration is finished.
- l) MN1 sends a work notification message to all nodes including TN3.
- m) Based on the above process, TN3 joins in the network successfully, and MN1 updates the cycle data that needs to be processed, and the network continues to process the bus data according to the specified cycle.

#### 6.6.7.3.1.4 Exit network interaction process

There are two situations in which a device or module on a TN exits the network: passively or actively exit. In the case where the device or the module exits the network due to abnormal power failure or cable disconnection, the TN management application user cannot actively notify the MN management application user, and the MN management application user passively perceives the offline state of the device or module on the TN and updates the network state and configuration. The TN should send a message when exiting the network because of the network configuration or other requirements. In this case, the MN management application user receives device or module's message of exiting the network on the corresponding TN from the management application user. This method is called normal exiting. See Figure 28 and Figure 29 for the interactive process of exiting the network.

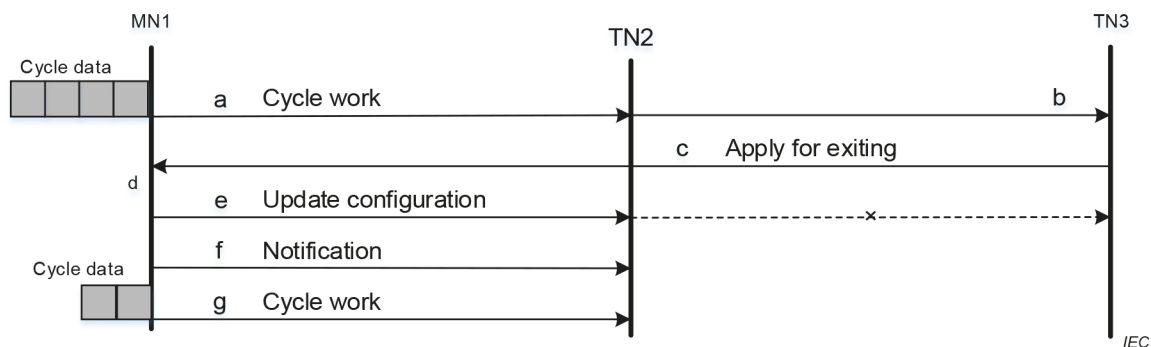


**Figure 28 – Passively exit the network interaction process**

Figure 28 shows the process to passively exiting:

- The bus is in the normal periodic data processing working state. At this time, Terminal Node 3 (TN3) has abnormally left the network, and the cyclic data sent by Management Node 1 (MN1) still contains the data that needs to be processed by TNs.
- If MN1 does not receive the data of the corresponding TN within the set period, TN3 is considered to be faulty and is ready to update the network state information.
- The handshake processing phase in the next cycle confirms that the response of TN3 that has left the network is not received, and the TN2 that has not left the network shall respond to MN1.
- MN1 protocol stack deletes the related items of TN3, releases relevant resources, and confirms the update of the network state and configuration information.
- MN1 notifies the TN2 and other nodes that have not left the network to update the network state and network configuration information.
- MN1 sends a work notification message based on the changed network configuration, and the TN2 receives the protocol data message and refreshes the working state machine to continue processing the bus data.
- MN1 cycles the data based on the update cycle and the network continues to cycle periodically.

Figure 29 shows the interactive process of actively exiting.



**Figure 29 – Actively exit network interaction process**

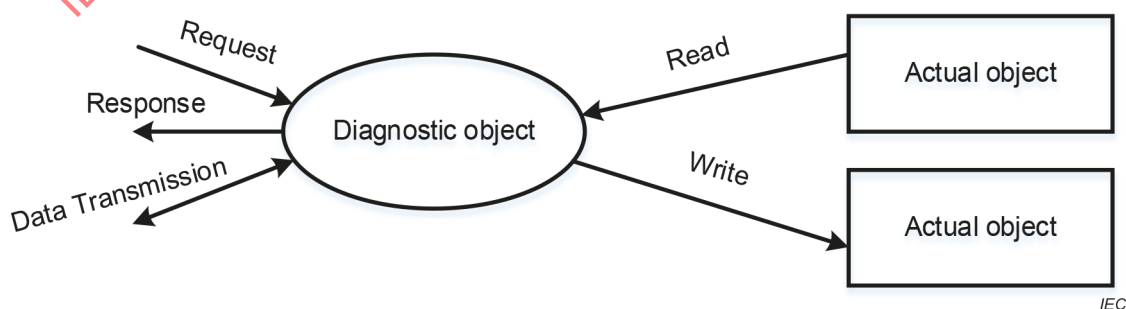
Unlike the passive exiting network processing procedure, the TN that actively exits the network should send an online exit protocol data message to the MN to request for exiting. Figure 29 shows the interaction process:

- The network is in normal operation and all nodes are cycled according to the set cycle.
- Terminal Node 3 (TN3) is ready to actively apply for exiting the current network based on the new configuration notification or external switch action.
- The TN3 sends a message including exiting apply to the Management Node 1 (MN1).
- MN1 receives the request from TN3, releases the corresponding resources, and updates related entries and configuration information.
- MN1 sends the latest configuration message to all active TNs. If the TN is in the exiting state, it will not do any processing even if it receives the broadcast message.
- The processing of step f) and step g) is consistent with the corresponding items in Figure 28.

### 6.6.7.3.2 Diagnostic process

The diagnostic service initiated by the AL user should read and write the actual diagnosis object. The user initiates a request for diagnosing the diagnosis object and responds to the diagnosis object, and then data is transmitted between the diagnosis object and user. At last, it will confirm that the diagnosis information for the user is complete, who should obtain the whole diagnostic data. The diagnostic service function is implemented based on C/S communication model. When the client requests the diagnostic data of the server, the server responds to the diagnostic request of the client and sends the diagnostic information to the client based on the specific diagnostic requirements.

Figure 30 shows the diagram of diagnostic object mapping.



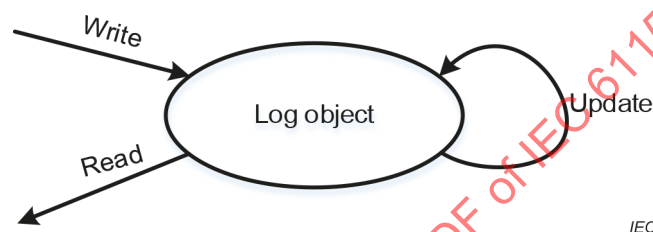
**Figure 30 – Diagram of diagnostic object mapping**

### 6.6.7.3.3 Logging process

During the process of AP interaction, the log information formed during the process of AP interaction exists in the form of files or data variables. The user can read or write the local or remote log object, while the user program sets the state machine to update the log object in a periodical or event-triggered manner.

The log interaction process adopts the C/S communication model or the P/S communication model to perform interactive processing. When the C/S communication model is adopted, the log object serves as a server to receive the client's reading and writing operation request, implements the acquisition, and writes processing of log information respectively. When the P/S communication model is adopted, the log publisher provides log information to the log subscriber and receives log events or interruption triggers log class information in the local log object to implement the reading or writing operation.

Figure 31 shows the diagram of logging process.



**Figure 31 – Diagram of the logging process**

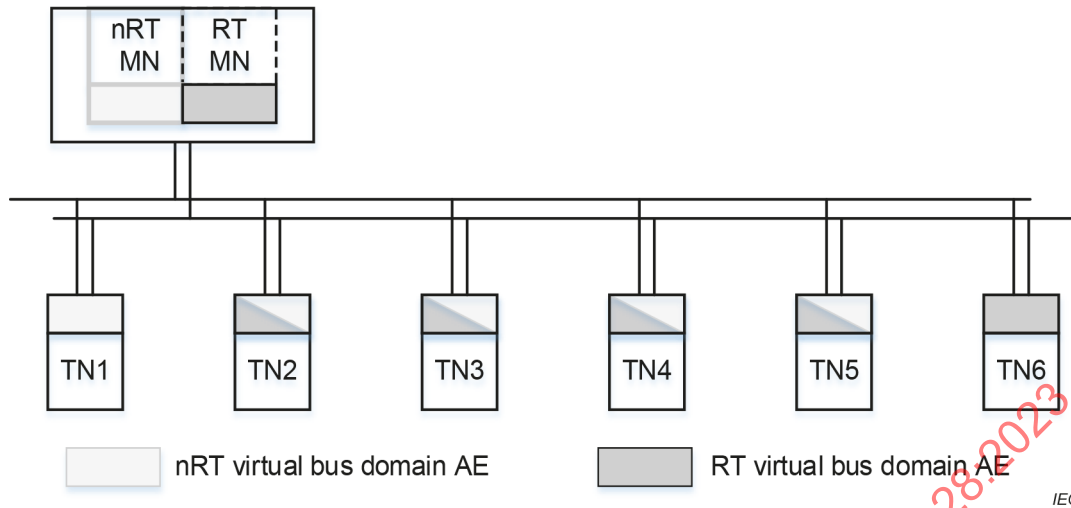
According to events occurred in the network, the log objects are classified into the following types:

- Ordinary log: The log data generated during normal operation of the system does not affect the system operation.
- Warning log: The alarm log generated by the device or module during operation that exceeds the operating parameter threshold.
- Error log: The error log generated by the device or module during operation has a limited impact on production activities based on application interaction.
- Emergency log: A serious error or abnormal log occurred during the operation of the device or module which is usually written to the permanent storage area.

## 6.6.8 Virtualization ASE

### 6.6.8.1 General

Since the Type 28 AL protocol can provide bus virtualization service, it can divide different devices or module areas on the same physical bus into different virtual bus domains through service interfaces defined by the virtualization ASE. The devices and modules in each virtual bus domain operate independently, and the virtual bus domains are mutually isolated from each other to meet the functional requirements of independent management of each virtual bus domain. In the Type 28 network, the virtual application service can at least provide the function of distinguishing between the RT bus domain and the nRT bus domain. Figure 32 shows a network topology diagram of two RT and nRT virtual buses based on the same physical bus. In the network, the nRT virtual bus domain implements nRT AP interaction, processes nRT data application services, and implements RT AP interaction in the RT virtual bus domain to process RT data application services.



**Figure 32 – A physical bus divided into two virtual bus domain diagrams**

Virtualization ASE defines interfaces and parameters of virtual application service, and provides service functions such as configuration, loading, running, suspending, and resetting for AE virtualization based on the virtualization ASE service interface.

#### 6.6.8.2 Interface

```
ALS-VIRTUALIZATION-Proc (
    Domain_ID,
    ProcessID,
    OptVirtualInfo,
    ALS_Result
)
```

Table 22 defines the virtual service interface parameters.

**Table 22 – Virtual service interface parameters**

| Parameter name | Data type | Input/output |
|----------------|-----------|--------------|
| DomainID       | UINT16    | IN           |
| ProcessID      | UINT16    | IN           |
| OptVirtualInfo | STRUCT    | IN OUT       |
| ALS_Result     | UINT32    | OUT          |

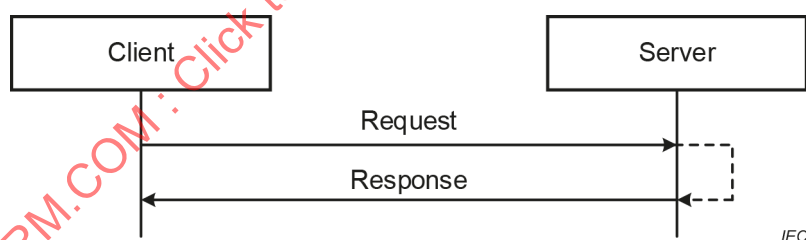
- DomainID: unique domain ID corresponding to the virtual bus.
- ProcessID: represents virtual bus service processing ID based on the bit bits, as follows:
  - Config (bit 0): Configure the virtual bus;
  - Load (bit 1): The virtual configuration is loaded into the device or module run parameters;
  - Startup (bit 2): Run a virtual bus based on a virtualized configuration;
  - Suspend (bit 3): Suspend the specified virtual bus operation, and the process data is not lost;
  - Reset (bit 4): Reload configuration parameters and restart specified virtual bus;
  - Stop (bit 5): Stop the operation of the virtual bus, the process data will be lost;
  - Unload (bit 6): Virtual bus configuration unload, bus restored to virtualized pre-configuration state.

- OptVirtualInfo: Information about the options related to the virtualization service parameters, including the following information:
  - VirtualID\_Cnt: UINT16 type, the number of virtual bus members;
  - VirtualID\_Array: UINT32 type array, virtual bus device or module NETWORKID array, array size determined by VirtualID\_Cnt;
  - ResourceID\_Cnt: The number of resource channels used by the corresponding virtual bus. The resource channel is created and maintained by the resource ASE;
  - ResourceID\_Array: UINT32 type, ID information of resource channels that need to be applied, released, and updated;
  - Delay\_Time: UINT16 value, the delay time of the virtual bus processing, the actual value should be an integer multiple of the bus cycle;
  - NetworkCfgPara: For the network configuration parameters of the virtual network, see Table 8 for the description of the data structure NETWORK\_CFG\_PARAMETERS.
- ALS\_Result: The value returned by the virtual service. See Table 15 for the description of the specific return value.

### 6.6.8.3 Interaction process

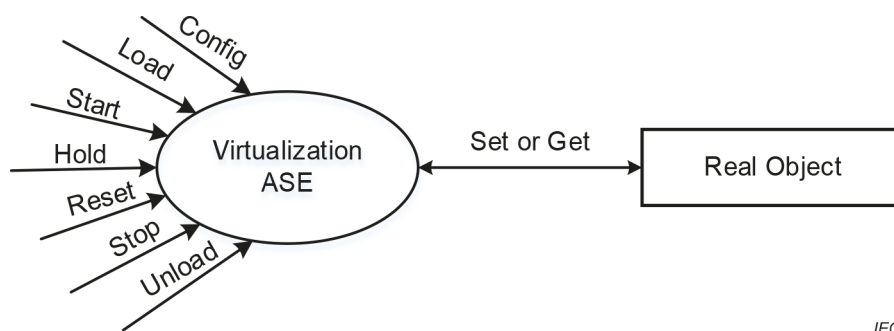
In Type 28 network, the virtualization ASE divides different devices or modules on the same physical bus into different virtual bus domains and allocates different configuration resource channels to the corresponding virtual bus domains, so that devices or modules of different virtualized bus domains can run independently for AP interaction processing.

The AP interaction based on the virtualization ASE adopts the C/S communication model, and the client application initiates the virtual application service request, and the server application responds to the request, implements the processing of the local variables and data based on the request, and then confirms to send the response data to the client application according to the requested content. In Figure 33, if the client application request fails or the server application responds incorrectly, the corresponding error information should be included in the message sent by the server application to the client application.



**Figure 33 – Diagram of virtualized AP interaction based on C/S communication model**

The local implementation of virtualization ASE is the operation of the virtual AE. The virtual AE sets or gets the actual object such as variables, entries or registers on the device or module to implement the specific operation (as shown in Figure 34) through the virtualization ASE service. The interface implements the virtual bus application for the operation of the device or module on the network and implements the specific AP interactive service function through the RT data ASE and the nRT data ASE based on the virtual bus.



**Figure 34 – Virtualization ASE local implementation diagram**

See Annex D for an example of a virtualized bus based on virtualization ASE.

## 6.7 Application service

### 6.7.1 Overview

Application services defined by the Type 28 AL include read/write services for RT periodic data, RT acyclic data, and nRT data, as well as read/write services for configuration and alarm data.

### 6.7.2 Read service

The AL ASE reads the value of the RT or nRT data object through the read service, and Table 23 describes the parameters of the service.

**Table 23 – Read service parameters**

| Parameter name | Req | Ind   | Rsp | Cnf   |
|----------------|-----|-------|-----|-------|
| SEP            | M   | M (=) | M   | M (=) |
| Type           | M   | M (=) |     |       |
| Index          | M   | M (=) |     |       |
| Length         | M   | M (=) |     |       |
| Data           |     |       | M   | M (=) |
| ErrCode        |     |       | M   | M     |

Parameter description:

- SEP: TN information of read service, including data buffer address and AR identifier.
- Type: Message type of read service, indicating the specific operation type of the read service performed by the service caller.
- Index: Use this parameter on the destination device to address the desired data object.
- Length: This parameter indicates the number of octets of the data object that should be read.
- Data: Data object value read by the service operation. When the read service operation succeeds, it is Length octets.
- ErrCode: Read service operation code value. When the read service succeeds, the error value is 0; if the read service fails, the corresponding error code is returned, see Table 25.

### 6.7.3 Write service

The AL ASE writes the value of the RT or nRT data object through write service. Table 24 describes the parameters of the service.

**Table 24 – Write service parameters**

| Parameter Name | Req | Ind   | Rsp | Cnf   |
|----------------|-----|-------|-----|-------|
| SEP            | M   | M (=) | M   | M (=) |
| Type           | M   | M (=) |     |       |
| Index          | M   | M (=) |     |       |
| Length         | M   | M (=) |     |       |
| Data           | M   | M (=) |     |       |
| ErrCode        |     |       | M   | M     |

Parameter description:

- SEP: TN information of write service, including data buffer address and AR identifier.
- Type: Message type of write service, indicating the specific operation type of the write service performed by the service caller.
- Index: Use this parameter on the destination device to address the desired data object.
- Length: This parameter indicates the number of octets of the data object that should be written.
- Data: Data object value read by the service operation. The length of the data object is Length octets.
- ErrCode: Write service operation code value. When the write service succeeds, the error value is 0x0000; if the write service fails, the corresponding error code is returned, see Table 25.

**Table 25 – Application service errcode**

| ErrCode name    | ErrCode value | ErrCode description          |
|-----------------|---------------|------------------------------|
| SUCCESS         | 0x0000        | Service success              |
| ReadErr         | 0x0001        | Read service failed          |
| WriteErr        | 0x0002        | Write service failed         |
| ModuleErr       | 0x0010        | Wrong module                 |
| VerConflict     | 0x0020        | Version conflict             |
| NotSupport      | 0x0040        | Unsupported service          |
| InvalidPara     | 0x0080        | Invalid parameter            |
| LengthErr       | 0x0100        | Wrong length                 |
| TypeConflict    | 0x0200        | Type conflict                |
| StateConflict   | 0x0400        | State conflict               |
| AccessDenied    | 0x0800        | Access denied                |
| NoResource      | 0x1000        | No resource                  |
| InvalidResource | 0x2000        | Resource accessed is invalid |

## Annex A (informative)

### Example of service data mapping DTU message

Refer to the DTU message format. The definition of the field content reflects the characteristics of the object (NETWORKID in the AL protocol header), data and method. For example, a temperature data reset operation is performed on a sensor device on the bus, and a module of the temperature sensor on the device is confirmed, and the data execution method of the module is a write command value represented by 4 octets. The corresponding value is 0x00001001, and the temperature value to be set has a length of 10 octets. The actual value is 0x00000000000000000064, and the corresponding RT DTU message content is as shown in Figure A.1.

| Octet Offset | Bit Offset             |   |   |   |   |   |   |   |            |
|--------------|------------------------|---|---|---|---|---|---|---|------------|
|              | 7                      | 6 | 5 | 4 | 3 | 2 | 1 | 0 |            |
| 0 to 1       | 0x04                   |   |   |   |   |   |   |   | Method_Len |
| 2 to 3       | 0x0A                   |   |   |   |   |   |   |   | Data_Len   |
| 4 to 7       | 0x00001001             |   |   |   |   |   |   |   | Method     |
| 8 to 17      | 0x00000000000000000064 |   |   |   |   |   |   |   | Data       |

IEC

Figure A.1 – Data transmission unit message content example

## Annex B (informative)

### Example of OPC UA data model and Type 28 data model mapping

For the data modelling of sensor business based on OPC UA, see the left part of Figure B.1, the sensor device itself has attribute data, type, device name and address. etc. At the same time, data is generated during the operation of the device, mainly dynamic data, and the device has at least two execution commands: start, stop. The data model instance based on OPC UA can be mapped to the corresponding Type 28 data model as shown in Figure B.1, the device attribute can be instantiated by the Module class, and the data generated during the running process and data of the device itself can be instantiated through the Data class, and the command words supported by the device can be instantiated based on the Method class. In this way, the data under the OPC UA data model can be completely mapped to the Type 28 data model, and the data communication with the OPC UA can be realized in the Type 28 network.

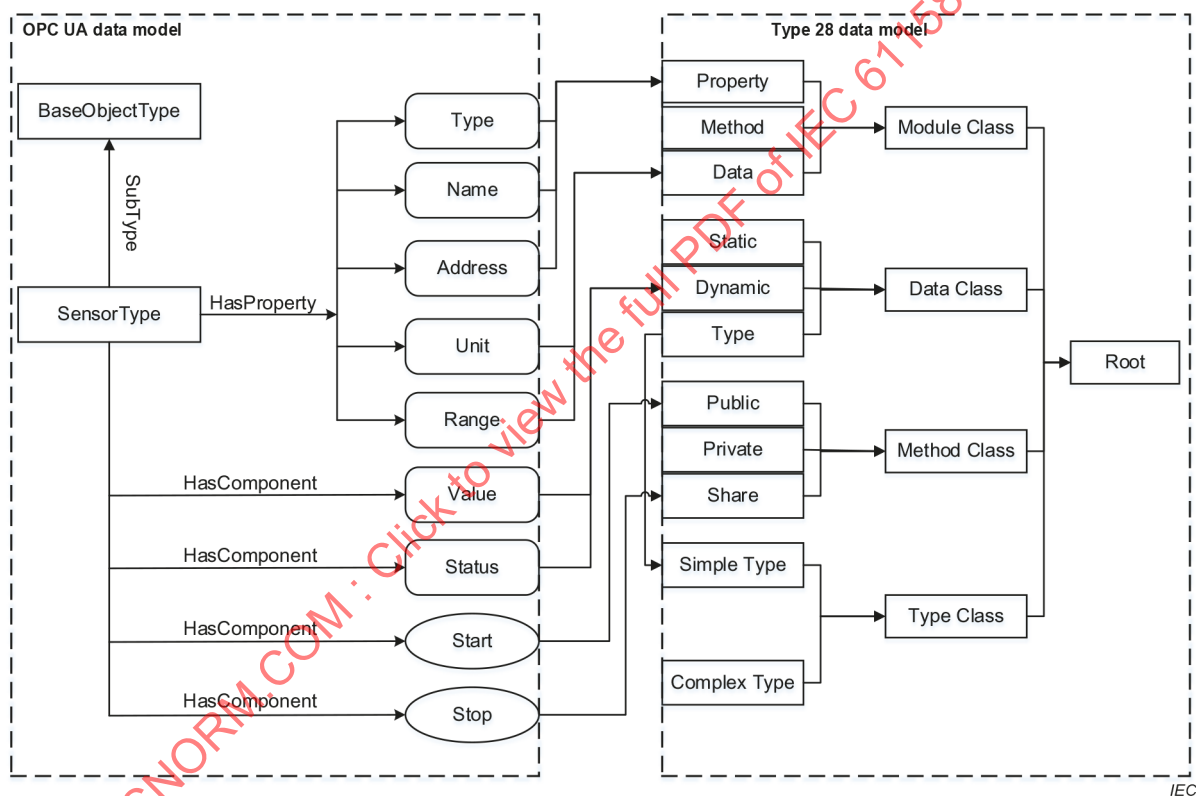
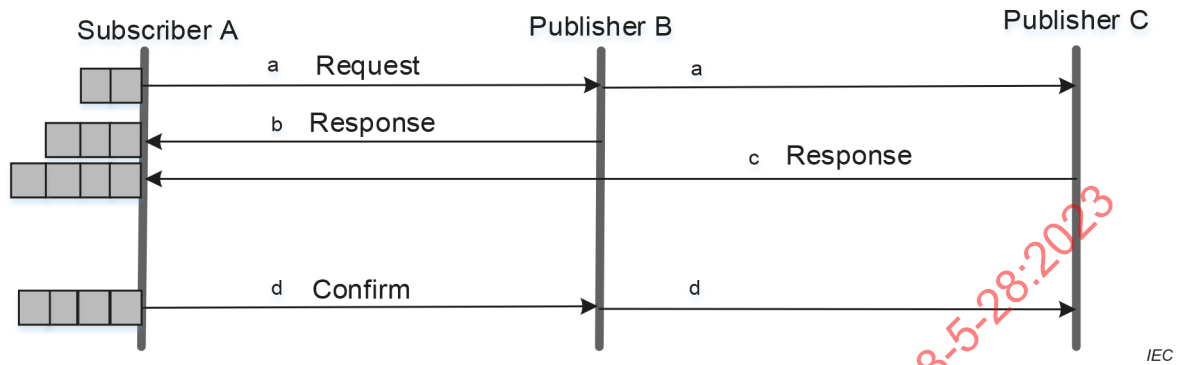


Figure B.1 – OPC UA data model and mapping of Type 28 data model

## Annex C (informative)

### Example of RTA service processing



**Figure C.1 – Example of RTA service processing**

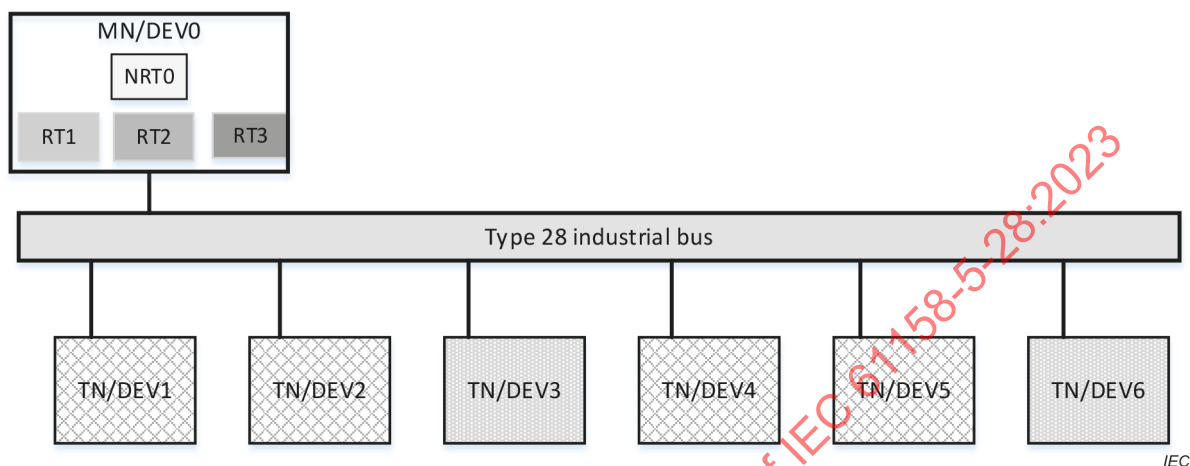
In Figure C.1, the procedure description of digital annotation is as follows:

- Subscriber A initiates the RTA data request (Request);
- Publisher B response synchronously to subscriber A;
- Publisher C response asynchronously to subscriber A;
- The subscriber confirms the request to the data and sends a confirmation message to publisher B and publisher C.

## Annex D (informative)

### Type 28 virtualization solution example

Figure D.1 shows the diagram of an industrial bus field network structure.

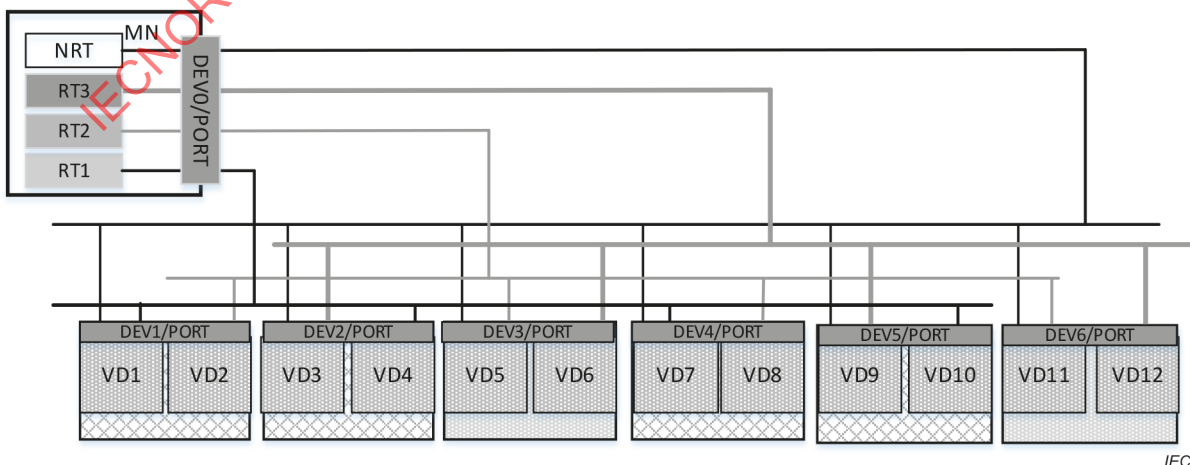


**Figure D.1 – Diagram of Type 28 industrial bus service deployment**

Description:

- DEVx: physical device x
- RTx: RT Service Logic Module x
- NRT0: nRT service logic module 0

In Figure D.1, there are 7 physical devices in the industrial field bus environment. For the bus control device DEV0, three RT application bus domains and one nRT application bus domain can be deployed according to the logical service function. The physical devices of the four bus domains are based on the same physical medium connected to each other in a bus topology. For all physical devices of the end node, the logical function module can be virtualized according to the logical service function. The extension of the industrial network topology based on the virtual bus scheme is described as follows:

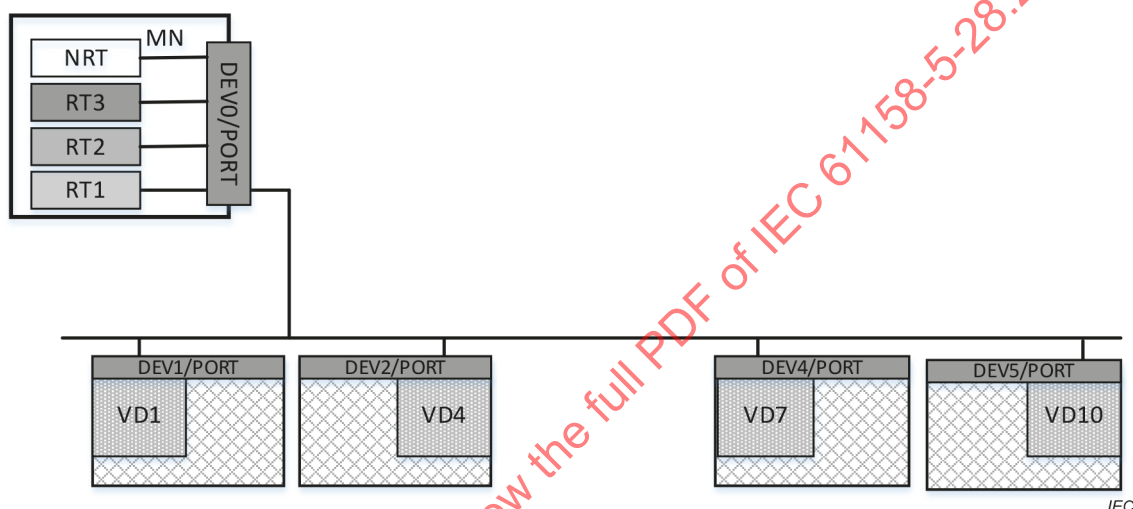


**Figure D.2 – Diagram of networking topology based on the virtualized bus**

Description:

- VDx: Virtualization Logic Function Module
- PORT: Interface of the device based on physical media access to the network

As shown in Figure D.2, for the TN device DEV1, two virtualization function modules VD1 and VD2 are distinguished according to logical services, and, for the field devices DEV2, DEV3, DEV4, DEV5, and DEV6, there are corresponding virtualized function modules. The MN and the TN are divided into different virtualized bus domains according to the logical service. The actual physical device external interface form is unchanged. Figure D.3 shows the actual topology form of the unified physical bus is unchanged. The virtual bus based on logical function service division implements whole bus communication and service data transmission functions. Figure D.3 is a diagram of a virtualized network topology based on the logical service function block RT1.



**Figure D.3 – Virtual network topology based on logical business function RT1**

In terms of data forwarding layer, the virtualization example implementation is described as follows:

- a) Data access mode: Data access is implemented by independent or shared buffers according to logical service requirements between different virtual devices on the same physical device. For example, for the field station DEV1, the virtual devices VD1 and VD2 can access nRT data in a shared buffer mode, and when VD1 and VD2 correspond to different virtual bus services, the independent buffer mode is used.
- b) The NETWORKID scheme follows the following rules:
  - Rule 1: Different virtual networks correspond to different domain IDs;
  - Rule 2: the module IDs corresponding to the same physical node are consistent;
  - Rule 3: The virtual device modules on the same physical device have different submodule IDs;
  - Rule 4: There can be multiple NETWORKID for the same physical device module;
  - Rule 5: nRT services use a separate virtual bus, based on physical device unique addressing, as shown in Figure D.2. The relevant node NETWORKID is set as follows in Table D.1.

**Table D.1 – Virtual bus environment node NETWORKID list**

| Device or module | NETWORKID  | Description                                                                                    |
|------------------|------------|------------------------------------------------------------------------------------------------|
| RT1              | 0x01000000 | RT virtual bus 1 corresponds to the NETWORKID of the virtual master node                       |
| RT2              | 0x02000000 | RT virtual bus 2 corresponds to the NETWORKID of the virtual MN                                |
| RT3              | 0x03000000 | RT virtual bus 3 corresponds to the NETWORKID of the virtual MN                                |
| NRT              | 0x0F000000 | NRT virtual bus corresponds to virtual MN NETWORKID                                            |
| DEV1             | 0x0F000100 | Physical device 1 corresponds to the NETWORKID of the nRT virtual bus                          |
| VD1              | 0x01000101 | Virtual device 1 on physical device 1 corresponds to the NETWORKID of virtual bus RT1          |
| VD2              | 0x02000102 | Virtual device 2 on physical device 1 corresponds to the NETWORKID of virtual bus RT2          |
| DEV2             | 0x0F000200 | Physical device 2 corresponds to the NETWORKID of the nRT virtual bus                          |
| VD4              | 0x01000202 | Virtual device 4 on the physical device 2 corresponds to the NETWORKID of the virtual bus RT1  |
| DEV4             | 0x0F000400 | Physical device 4 corresponds to the NETWORKID of the nRT virtual bus                          |
| VD7              | 0x01000401 | Virtual device 7 on the physical device 4 corresponds to the NETWORKID of the virtual bus RT1. |
| DEV5             | 0x0F000500 | Physical device 5 corresponds to the NETWORKID of the nRT virtual bus                          |
| VD10             | 0x01000501 | Virtual device 10 on the physical device 5 corresponds to the NETWORKID of the virtual bus RT1 |

## Bibliography

IEC 61158-2, *Industrial communication networks – Fieldbus specifications – Part 2: Physical layer specification and service definition*

IEC 61158-3-28, *Industrial communication networks – Fieldbus specifications – Part 3-28: Data-link layer service definition – Type 28 elements*

IEC 61158-6 (all parts), *Industrial communication networks – Fieldbus specifications – Part 6-x: Application layer protocol specification – Type x elements*

IEC 61784-1 (all parts), *Industrial networks – Profiles – Part 1-x: Fieldbus profiles – CPF x*

IEC 61784-2 (all parts), *Industrial networks – Profiles – Part 2-x: Additional real-time fieldbus profiles based on ISO/IEC/IEEE 8802-3 – CPF x*

---

IECNORM.COM : Click to view the full PDF of IEC 61158-5-28:2023

IECNORM.COM : Click to view the full PDF of IEC 61158-5-28:2023

## SOMMAIRE

|                                                                                 |     |
|---------------------------------------------------------------------------------|-----|
| AVANT-PROPOS .....                                                              | 72  |
| INTRODUCTION.....                                                               | 74  |
| 1 Domaine d'application .....                                                   | 75  |
| 1.1 Vue d'ensemble .....                                                        | 75  |
| 1.2 Spécifications .....                                                        | 76  |
| 1.3 Conformité .....                                                            | 76  |
| 2 Références normatives .....                                                   | 76  |
| 3 Termes, définitions, symboles, abréviations et conventions .....              | 77  |
| 3.1 Termes et définitions référencés .....                                      | 77  |
| 3.1.1 Termes de l'ISO/IEC 7498-1 .....                                          | 77  |
| 3.1.2 Termes de l'ISO/IEC 8822 .....                                            | 78  |
| 3.1.3 Termes de l'ISO/IEC 9545 .....                                            | 78  |
| 3.1.4 Termes de l'ISO/IEC 8824 .....                                            | 78  |
| 3.2 Termes et définitions supplémentaires pour la présente IEC 61158-5-28 ..... | 78  |
| 3.3 Abréviations et symboles .....                                              | 79  |
| 3.4 Conventions.....                                                            | 80  |
| 3.4.1 Vue d'ensemble .....                                                      | 80  |
| 3.4.2 Conventions pour les définitions de classes .....                         | 80  |
| 3.4.3 Conventions pour les définitions de service .....                         | 82  |
| 4 Concepts .....                                                                | 83  |
| 5 ASE de type de données .....                                                  | 83  |
| 5.1 Généralités .....                                                           | 83  |
| 5.1.1 Vue d'ensemble .....                                                      | 83  |
| 5.1.2 Vue d'ensemble des types de base .....                                    | 84  |
| 5.1.3 Vue d'ensemble des types de longueurs fixes .....                         | 84  |
| 5.1.4 Vue d'ensemble des types construits .....                                 | 84  |
| 5.2 Types de données définis d'AL .....                                         | 85  |
| 5.2.1 Types de longueur fixe .....                                              | 85  |
| 5.2.2 Type de chaîne .....                                                      | 87  |
| 5.2.3 Type de structure .....                                                   | 88  |
| 6 Spécification du modèle de communication de type 28 .....                     | 93  |
| 6.1 Généralités .....                                                           | 93  |
| 6.2 Pile de protocole pour le bus de terrain de type 28 .....                   | 94  |
| 6.3 Modélisation de la correspondance des données .....                         | 95  |
| 6.4 Vue d'ensemble du modèle de communication de type 28.....                   | 97  |
| 6.4.1 Généralités .....                                                         | 97  |
| 6.4.2 P/S .....                                                                 | 97  |
| 6.4.3 C/S .....                                                                 | 99  |
| 6.5 Table d'informations de gestion relatives à l'AL .....                      | 99  |
| 6.5.1 Généralités .....                                                         | 99  |
| 6.5.2 Table d'informations de gestion du temps .....                            | 100 |
| 6.5.3 Table d'informations de gestion de réseau .....                           | 101 |
| 6.6 ASE .....                                                                   | 102 |
| 6.6.1 Vue d'ensemble .....                                                      | 102 |
| 6.6.2 ASE de données RT .....                                                   | 103 |
| 6.6.3 ASE de données nRT .....                                                  | 105 |

|                                                                                                                            |                            |     |
|----------------------------------------------------------------------------------------------------------------------------|----------------------------|-----|
| 6.6.4                                                                                                                      | ASE de temps.....          | 108 |
| 6.6.5                                                                                                                      | ASE de ressource.....      | 111 |
| 6.6.6                                                                                                                      | ASE d'adressage.....       | 113 |
| 6.6.7                                                                                                                      | ASE de gestion.....        | 119 |
| 6.6.8                                                                                                                      | ASE de virtualisation..... | 127 |
| 6.7                                                                                                                        | Service d'application..... | 129 |
| 6.7.1                                                                                                                      | Vue d'ensemble.....        | 129 |
| 6.7.2                                                                                                                      | Service de lecture.....    | 129 |
| 6.7.3                                                                                                                      | Service d'écriture.....    | 130 |
| Annexe A (informative) Exemple de message DTU de correspondance de données de service.....                                 |                            | 132 |
| Annexe B (informative) Exemple de correspondance de modèle de données OPC UA et de modèles de données de type 28.....      |                            | 133 |
| Annexe C (informative) Exemple de traitement de service RTA.....                                                           |                            | 134 |
| Annexe D (informative) Exemple de solution de virtualisation de type 28.....                                               |                            | 135 |
| Bibliographie.....                                                                                                         |                            | 138 |
| Figure 1 – Exemple de hiérarchie de classe de type de données.....                                                         |                            | 83  |
| Figure 2 – Type de données BITMAP.....                                                                                     |                            | 88  |
| Figure 3 – AL dans l'architecture de la pile de protocole de bus de terrain.....                                           |                            | 94  |
| Figure 4 – Architecture de protocole AL de type 28.....                                                                    |                            | 94  |
| Figure 5 – Structure de l'objet de modélisation de correspondance de données.....                                          |                            | 95  |
| Figure 6 – Schéma des tampons de données de mise en correspondance et des variables de données de service.....             |                            | 96  |
| Figure 7 – Schéma du modèle P/S.....                                                                                       |                            | 98  |
| Figure 8 – Modèle P/S du mode PUSH.....                                                                                    |                            | 98  |
| Figure 9 – Modèle P/S du mode PULL.....                                                                                    |                            | 98  |
| Figure 10 – Schéma du modèle de communication C/S.....                                                                     |                            | 99  |
| Figure 11 – Modèle de transmission de données RT.....                                                                      |                            | 105 |
| Figure 12 – Schéma du modèle de réponse de demande de données nRT.....                                                     |                            | 107 |
| Figure 13 – Schéma d'interaction AP de données nRT basé sur le modèle de communication C/S.....                            |                            | 107 |
| Figure 14 – Schéma d'interaction AP de données nRT basé sur le modèle de communication P/S.....                            |                            | 108 |
| Figure 15 – Processus d'interaction d'application de synchronisation temporelle.....                                       |                            | 110 |
| Figure 16 – Processus d'interrogation de temps.....                                                                        |                            | 111 |
| Figure 17 – Interaction de l'AP de ressource sur la base du modèle de communication C/S.....                               |                            | 113 |
| Figure 18 – Schéma de la fonction de service local d'ASE de ressource.....                                                 |                            | 113 |
| Figure 19 – Structure du champ d'identification NETWORKID.....                                                             |                            | 114 |
| Figure 20 – Schéma du processus d'interaction de NETWORKID.....                                                            |                            | 116 |
| Figure 21 – Schéma de la fonction de service de table de correspondance MAC.....                                           |                            | 117 |
| Figure 22 – Schéma de la fonction de service de table de correspondance IP.....                                            |                            | 117 |
| Figure 23 – Message de données AL de type 28 mis en correspondance avec la charge utile de données de charge utile IP..... |                            | 118 |
| Figure 24 – Correspondance d'en-tête de protocole AL type 28 et d'en-tête de protocole IP.....                             |                            | 118 |

|                                                                                                                                  |     |
|----------------------------------------------------------------------------------------------------------------------------------|-----|
| Figure 25 – Le message de données IP est mis en correspondance avec des données valides du message de données AL de type 28..... | 119 |
| Figure 26 – Processus d'initialisation de configuration statique.....                                                            | 122 |
| Figure 27 – Processus d'interaction de rattachement au réseau.....                                                               | 123 |
| Figure 28 – Processus d'interaction de sortie passive du réseau .....                                                            | 124 |
| Figure 29 – Processus d'interaction de sortie active du réseau .....                                                             | 125 |
| Figure 30 – Schéma de correspondance d'objets de diagnostic .....                                                                | 126 |
| Figure 31 – Schéma du processus de journalisation .....                                                                          | 126 |
| Figure 32 – Schémas d'un bus physique divisé en deux domaines de bus virtuels .....                                              | 127 |
| Figure 33 – Schéma d'interaction AP virtuel basé sur le modèle de communication C/S.....                                         | 129 |
| Figure 34 – Schéma de mise en œuvre locale de l'ASE de virtualisation.....                                                       | 129 |
| Figure A.1 – Exemple de contenu de message d'unité de transmission de données.....                                               | 132 |
| Figure B.1 – Modèle de données OPC UA et mise en correspondance du modèle de données de type 28 .....                            | 133 |
| Figure C.1 – Exemple de traitement de service RTA .....                                                                          | 134 |
| Figure D.1 – Schéma du déploiement du service de bus industriel de type 28.....                                                  | 135 |
| Figure D.2 – Schéma de topologie de mise en réseau basé sur le bus virtualisé .....                                              | 135 |
| Figure D.3 – Topologie de réseau virtuel basée sur la fonction d'activité logique RT1.....                                       | 136 |
| Tableau 1 – Codage des données de type TIMEV.....                                                                                | 85  |
| Tableau 2 – Codage du type TIMEDATE.....                                                                                         | 86  |
| Tableau 3 – Codage du type TIMEOFDAY .....                                                                                       | 86  |
| Tableau 4 – Codage du type TIMEDIFFER.....                                                                                       | 87  |
| Tableau 5 – Codage des données de type VISIBLESTRING .....                                                                       | 87  |
| Tableau 6 – Description de la structure d'information de temps TIMEINFO_S .....                                                  | 88  |
| Tableau 7 – Description de la structure d'information de l'option d'horloge<br>CLOCK_OPTION_INFO_S.....                          | 89  |
| Tableau 8 – Description de la structure des paramètres de configuration du réseau<br>NETWORK_CFG_PARA_S .....                    | 90  |
| Tableau 9 – Description de la table de correspondance MAC<br>NETWORKID_MAC_MAP_TABLE_S.....                                      | 91  |
| Tableau 10 – Description de la structure de table de correspondance IP<br>IP_MAP_TABLE_S .....                                   | 92  |
| Tableau 11 – Liste des paramètres de la table de correspondance<br>des données de service .....                                  | 97  |
| Tableau 12 – Table d'informations de gestion du temps.....                                                                       | 100 |
| Tableau 13 – Table d'informations de gestion de réseau.....                                                                      | 101 |
| Tableau 14 – Modèles de communication pris en charge par les ASE AL de type 28.....                                              | 102 |
| Tableau 15 – Liste des valeurs de retour de l'interface de service d'application .....                                           | 103 |
| Tableau 16 – Paramètres de l'interface de service de données RT .....                                                            | 103 |
| Tableau 17 – Paramètres de l'interface de service de données nRT .....                                                           | 106 |
| Tableau 18 – Paramètres de l'interface du service de cadencement .....                                                           | 109 |
| Tableau 19 – Paramètres de l'interface de service de ressource.....                                                              | 112 |
| Tableau 20 – Paramètres de l'interface de service d'adressage .....                                                              | 115 |
| Tableau 21 – Paramètres de l'interface de service de gestion.....                                                                | 120 |

|                                                                                  |     |
|----------------------------------------------------------------------------------|-----|
| Tableau 22 – Paramètres de l'interface de service virtuel .....                  | 128 |
| Tableau 23 – Paramètres du service de lecture .....                              | 130 |
| Tableau 24 – Paramètres du service d'écriture .....                              | 130 |
| Tableau 25 – ErrCode du service d'application .....                              | 131 |
| Tableau D.1 – Liste de NETWORKID du nœud de l'environnement de bus virtuel ..... | 137 |

IECNORM.COM : Click to view the full PDF of IEC 61158-5-28:2023

## COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

**RÉSEAUX DE COMMUNICATION INDUSTRIELS –  
SPÉCIFICATIONS DES BUS DE TERRAIN –****Partie 5-28: Définition des services de la couche application –  
Éléments de type 28****AVANT-PROPOS**

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

L'attention est attirée sur le fait que l'utilisation du type de protocole associé est restreinte par les détenteurs des droits de propriété intellectuelle. En tout état de cause, l'engagement de renonciation partielle aux droits de propriété intellectuelle pris par les détenteurs de ces droits autorise l'utilisation d'un type de protocole de couche avec les autres protocoles de couche du même type, ou dans des combinaisons avec d'autres types autorisés explicitement par les détenteurs des droits de propriété intellectuelle pour ce type.

NOTE Les combinaisons de types de protocoles sont spécifiées dans la série IEC 61784-1 et la série IEC 61784-2.

L'IEC 61158-5-28 a été établie par le sous-comité 65C: Réseaux industriels, du comité d'études 65 de l'IEC: Mesure, commande et automation dans les processus industriels. Il s'agit d'une Norme internationale.

Le texte de cette Norme internationale est issu des documents suivants:

| Projet        | Rapport de vote |
|---------------|-----------------|
| 65C/1206/FDIS | 65C/1235/RVD    |

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

La version française de la norme n'a pas été soumise au vote.

La langue employée pour l'élaboration de cette Norme internationale est l'anglais.

Ce document a été rédigé selon les Directives ISO/IEC, Partie 2, il a été développé selon les Directives ISO/IEC, Partie 1 et les Directives ISO/IEC, Supplément IEC, disponibles sous [www.iec.ch/members\\_experts/refdocs](http://www.iec.ch/members_experts/refdocs). Les principaux types de documents développés par l'IEC sont décrits plus en détail sous [www.iec.ch/publications](http://www.iec.ch/publications).

Une liste de toutes les parties de la série IEC 61158, publiées sous le titre général *Réseaux de communication industriels – Spécifications des bus de terrain*, se trouve sur le site web de l'IEC.

Le comité a décidé que le contenu de ce document ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous [webstore.iec.ch](http://webstore.iec.ch) dans les données relatives au document recherché. À cette date, le document sera:

- reconduit,
- supprimé,
- remplacé par une édition révisée, ou
- amendé.

## INTRODUCTION

Le présent document fait partie d'une série élaborée pour faciliter l'interconnexion des composants de systèmes d'automatisation. Il est lié aux autres normes de la série telle que définie par le modèle de référence de bus de terrain "à trois couches" décrit dans l'IEC 61158-1.

Le service d'application est assuré par le protocole d'application qui utilise les services disponibles à partir de la couche liaison de données ou d'une autre couche immédiatement inférieure. Le présent document définit les caractéristiques de services d'application pouvant être exploitées par les applications de bus de terrain et/ou la gestion système.

Dans l'ensemble des normes de bus de terrain, le terme "service" désigne la capacité abstraite fournie par une couche du modèle de référence de base de l'OSI à la couche immédiatement supérieure. Ainsi, le service de couche application défini dans le présent document est un service d'architecture conceptuel, indépendant des services d'administration et de mise en œuvre.

IECNORM.COM : Click to view the full PDF of IEC 61158-5-28:2023

## RÉSEAUX DE COMMUNICATION INDUSTRIELS – SPÉCIFICATIONS DES BUS DE TERRAIN –

### Partie 5-28: Définition des services de la couche application – Éléments de type 28

#### 1 Domaine d'application

##### 1.1 Vue d'ensemble

La couche application de bus de terrain (FAL, Fieldbus Application Layer) fournit aux programmes d'utilisateur un moyen d'accéder à l'environnement de communication du bus de terrain. A cet égard, la FAL peut être considérée comme une "fenêtre entre programmes d'application correspondants".

La présente partie de l'IEC 61158 donne les éléments communs visant à assurer les communications de messagerie de base prioritaires et non prioritaires entre les programmes d'application d'un environnement d'automatisation et d'un matériel spécifique au bus de terrain de type 28. Le terme "prioritaire" est utilisé pour indiquer la présence d'une fenêtre temporelle, dans laquelle il est exigé de réaliser une ou plusieurs actions spécifiées selon un niveau défini de certitude.

Le présent document définit de manière abstraite le service visible de l'extérieur fourni par les différents types de la couche application de bus de terrain en ce qui concerne:

- un modèle abstrait pour définir des ressources (objets) d'application capables d'être manipulées par les utilisateurs par l'intermédiaire de l'utilisation du service FAL;
- les actions et événements liés aux primitives du service;
- les paramètres associés à chaque action et événement de primitive, ainsi que la forme prise par ces paramètres; et
- l'interrelation entre ces actions et événements, et leurs séquences valides.

Le but du présent document est de définir les services fournis à:

- l'utilisateur de FAL à la frontière entre l'utilisateur et la couche application du modèle de référence de bus de terrain; et
- la gestion système au niveau de la frontière entre la couche application et la gestion système selon le modèle de référence de bus de terrain.

Le présent document spécifie la structure et les services de la couche application des bus de terrain de l'IEC, conformément au modèle de référence de base de l'OSI (ISO/IEC 7498-1) et à la structure de la couche application de l'OSI (ISO/IEC 9545).

Les services et protocoles FAL sont fournis par des entités d'application FAL présentes dans les processus d'application. Une entité AE de couche FAL se compose d'un ensemble d'éléments de service d'application (ASE) orientés objet et d'une entité de gestion de couche (LME) qui gère l'entité AE. Les éléments de service d'application proposent des services de communication opérant sur un ensemble de classes d'objets de processus d'application (APO) connexes. L'un des éléments de service d'application FAL est un ASE de gestion qui fournit un ensemble commun de services de gestion des instances de classes FAL.

Bien que ces services spécifient, du point de vue des applications, la manière dont les demandes et les réponses sont données et délivrées, ils ne précisent pas ce qu'il est nécessaire que les applications demandeuses et répondantes en fassent. En d'autres termes,

les aspects liés au comportement des applications ne sont pas précisés; seule est spécifiée une définition des demandes et réponses qu'elles peuvent envoyer/recevoir. Cela permet d'assurer une plus grande souplesse aux utilisateurs FAL quant à la normalisation du comportement de ce type d'objet. Outre ces services, le présent document définit également certains services de soutien donnant accès à la couche FAL dans un but de commande de certains aspects de son fonctionnement.

## 1.2 Spécifications

L'objectif principal du présent document est de spécifier les caractéristiques des services conceptuels de couche application qui sont adaptées à des communications prioritaires et donc complètent le modèle de référence de base de l'OSI en guidant le développement des protocoles de couche application pour les communications prioritaires.

Un objectif secondaire est de fournir des chemins de migration à partir de protocoles de communication industriels préexistants. C'est ce dernier objectif qui donne naissance à la diversité des services normalisés comme les divers types de l'IEC 61158-6 et les protocoles correspondants normalisés dans la série IEC 61158-6.

Le présent document peut être utilisé comme la base pour les interfaces de programmation d'applications formelles. Néanmoins, il n'est pas une interface de programmation formelle et il est nécessaire pour toute interface de ce type de traiter de questions de mise en œuvre qui ne sont pas couvertes par le présent document, y compris:

- les tailles et l'ordonnancement des octets pour les divers paramètres de service à plusieurs octets; et
- la corrélation de primitives appariées "request-confirm" (demande et confirmation) ou "indication-response" (indication et réponse).

## 1.3 Conformité

Le présent document ne spécifie aucune mise en œuvre ou aucun produit individuel, de même qu'il ne restreint nullement les mises en œuvre des entités de couche application dans les systèmes d'automatisation industriels.

Il n'y a pas de conformité des équipements à la présente norme de définition des services de couche application. La conformité est au contraire obtenue par une mise en œuvre de protocoles conformes de couche application qui satisfont aux services de couche application de type 28 définis dans le présent document.

## 2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

NOTE Toutes les parties de la série IEC 61158, ainsi que de la série IEC 61784-1 et de la série IEC 61784-2, font l'objet d'une maintenance simultanée. Les références croisées à ces documents dans le texte se rapportent par conséquent aux éditions datées dans la présente liste de références normatives.

IEC 61158-1:2023, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 1: Vue d'ensemble et recommandations pour les séries IEC 61158 et IEC 61784*

IEC 61158-4-28:2023, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 4-28: Spécification du protocole de la couche liaison de données – Éléments de type 28*

IEC 61158-6-28:2023, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 6-28: Spécification du protocole de la couche application – Eléments de type 28*

ISO/IEC 7498-1:1994, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Modèle de référence de base: Le modèle de base*

ISO/IEC 7498-3:1997, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Modèle de référence de base: Dénomination et adressage*

ISO/IEC 8822, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Définition du service de présentation*

ISO/IEC 8824-1, *Technologies de l'information – Notation de syntaxe abstraite numéro un (ASN.1) – Partie 1: Spécification de la notation de base*

ISO/IEC 9545, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Structure de la couche application*

ISO/IEC 10731:1994, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Modèle de référence de base – Conventions pour la définition des services OSI*

ISO/IEC/IEEE 8802-3:2021, *Technologies de l'information – Télécommunications et échange entre systèmes informatiques – Exigences pour les réseaux locaux et métropolitains – Partie 3: Norme pour Ethernet*

IETF RFC 791, J. Postel, *Internet Protocol*, septembre 1981, disponible à l'adresse <https://www.rfc-editor.org/info/rfc791> (révisée le 18/02/2022)

IETF RFC 2460, S. Deering and R. Hinden, *"Internet Protocol, Version 6 (IPv6) Specification"*, décembre 1998, disponible à l'adresse <https://www.rfc-editor.org/info/rfc2460> (révisée le 18/02/2022)

### 3 Termes, définitions, symboles, abréviations et conventions

Pour les besoins du présent document, les termes, définitions, abréviations et conventions suivants s'appliquent.

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse <https://www.electropedia.org/>
- ISO Online browsing platform: disponible à l'adresse <https://www.iso.org/obp>

#### 3.1 Termes et définitions référencés

##### 3.1.1 Termes de l'ISO/IEC 7498-1

Pour les besoins du présent document, les termes suivants donnés dans l'ISO/IEC 7498-1 s'appliquent:

- entité d'application;
- processus d'application;
- unité de données de protocole d'application;
- élément de service d'application;
- invocation d'entité d'application;

- invocation de processus d'application;
- transaction d'application;
- système ouvert réel;
- syntaxe de transfert.

### 3.1.2 Termes de l'ISO/IEC 8822

Pour les besoins du présent document, les termes suivants donnés dans l'ISO/IEC 8822 s'appliquent:

- syntaxe abstraite;
- contexte de présentation.

### 3.1.3 Termes de l'ISO/IEC 9545

Pour les besoins du présent document, les termes suivants donnés dans l'ISO/IEC 9545 s'appliquent:

- invocation d'entités d'application;
- type d'entité d'application;
- invocation de processus d'application;
- type de processus d'application;
- élément de service d'application.

### 3.1.4 Termes de l'ISO/IEC 8824

Pour les besoins du présent document, les termes suivants donnés dans l'ISO/IEC 8824 s'appliquent:

- identificateur d'objet;
- type.

## 3.2 Termes et définitions supplémentaires pour la présente IEC 61158-5-28

Les termes et définitions suivants s'appliquent au présent document.

### 3.2.1

#### **interrogation d'horloge**

obtenir des informations d'horloge de la part du nœud de gestion ou de l'appareil nœud terminal

### 3.2.2

#### **synchronisation d'horloge**

étalonnage de l'horloge de l'appareil nœud terminal

### 3.2.3

#### **appareil de commande**

appareil qui commande tous les appareils de terrain pour les opérations logiques, le cadencement, les calculs, etc.

### 3.2.4

#### **cyclique**

événements qui se répètent de manière régulière et répétitive

### 3.2.5

#### **temps cyclique**

temps de traitement cyclique d'un appareil ou d'un module

**3.2.6****appareil de terrain**

entité physique qui est connectée au processus ou au matériel d'usine et qui possède au moins un élément de signalisation qui communique avec un ou plusieurs autres éléments de signalisation par le biais d'un câble

**3.2.7****nœud de gestion**

appareil d'allocation et de gestion des ressources de communication physique de réseau de type 28

**3.2.8****données non-RT**

données insensibles aux exigences déterministes du temps

**3.2.9****données RT**

données sensibles aux exigences déterministes du temps

**3.2.10****nœud terminal**

appareil dans un réseau de type 28 qui communique sur la base des ressources de communication physique allouées

**3.3 Abréviations et symboles**

|                 |                                                                                  |
|-----------------|----------------------------------------------------------------------------------|
| <b>AL</b>       | Application Layer (couche application)                                           |
| <b>ALCE</b>     | Application Layer Clock Entity (entité horloge de la couche application)         |
| <b>ALDE</b>     | Application Layer Data Entity (entité données de la couche application)          |
| <b>ALME</b>     | Application Layer Management Entity (entité de gestion de la couche application) |
| <b>ALS</b>      | Application Layer Service (service de la couche application)                     |
| <b>AE</b>       | Application Entity (entité d'application)                                        |
| <b>AP</b>       | Application Process (processus d'application)                                    |
| <b>APO</b>      | AP Object (objet AP)                                                             |
| <b>AR</b>       | Application Relationship (relation d'application)                                |
| <b>ASE</b>      | Application Service Elements (éléments de service d'application)                 |
| <b>C/S</b>      | Client/Server (Client/Serveur)                                                   |
| <b>DL-</b>      | Data-Link layer (couche liaison de données [utilisé comme préfixe])              |
| <b>DLCE</b>     | Data-Link Clock Entity (entité horloge de liaison de données)                    |
| <b>DLDE</b>     | Data-Link Data Entity (entité de données de liaison de données)                  |
| <b>DLE</b>      | Data-Link Entity (entité de liaison de données)                                  |
| <b>DLL</b>      | Data-Link Layer (couche liaison de données)                                      |
| <b>DLS</b>      | Data Link Service (service de liaison de données)                                |
| <b>DLP</b>      | Data-Link Protocol (protocole de liaison de données)                             |
| <b>DLS-user</b> | Data-Link Service user (utilisateur de service de liaison de données)            |
| <b>DTC</b>      | Data Transmission Channel (canal de transmission de données)                     |
| <b>DTS</b>      | Data Transmission Service (service de transmission de données)                   |
| <b>DTU</b>      | Data Transmission Unit (unité de transmission de données)                        |
| <b>DLSDU</b>    | Data-Link Service Data Unit (unité de données de service de liaison de données)  |

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| <b>DLPDU</b>  | Data-Link Protocol Data Unit (unité de données de protocole de liaison de données) |
| <b>MAC</b>    | Medium Access Control (contrôle d'accès au support)                                |
| <b>MN</b>     | Management Node (nœud de gestion)                                                  |
| <b>NodeID</b> | Node Identifier (identificateur de nœud)                                           |
| <b>nRT</b>    | non-Real-Time (temps différé)                                                      |
| <b>PhL</b>    | Physical Layer (couche physique)                                                   |
| <b>RT</b>     | Real-Time (temps réel)                                                             |
| <b>RTA</b>    | Real-Time Acyclic (temps réel acyclique)                                           |
| <b>RTC</b>    | Real-Time Cyclic (temps réel cyclique)                                             |
| <b>TN</b>     | Terminal Node (nœud terminal)                                                      |

### 3.4 Conventions

#### 3.4.1 Vue d'ensemble

La couche application de bus de terrain (FAL) se compose d'un ensemble d'éléments ASE orientés objet. Chaque ASE est spécifié dans un paragraphe distinct. Chaque spécification d'ASE est constituée de deux parties, à savoir sa spécification de classe et sa spécification de service.

La spécification de classe définit les attributs de la classe. La spécification de service définit les services fournis par l'ASE.

#### 3.4.2 Conventions pour les définitions de classes

Les définitions de classes sont décrites à l'aide de modèles. Chaque modèle est constitué d'une liste d'attributs de la classe. La forme générale du modèle est représentée ci-dessous:

| ASE FAL:             |     |                         | Nom de l'ASE                   |
|----------------------|-----|-------------------------|--------------------------------|
| <b>CLASS:</b>        |     | <b>Nom de la classe</b> |                                |
| <b>CLASS ID</b>      |     |                         | <b>N°</b>                      |
| <b>PARENT CLASS:</b> |     |                         | <b>Nom de la classe parent</b> |
| <b>ATTRIBUTS:</b>    |     |                         |                                |
| 1                    | (f) | Key Attribute:          | identificateur numérique       |
| 2                    | (f) | Key Attribute:          | Nom                            |
| 3                    | (o) | Attribute:              | nom (valeurs) de l'attribut    |
| 4                    | (o) | Attribute:              | nom (valeurs) de l'attribut    |
| 4.1                  | (s) | Attribute:              | nom (valeurs) de l'attribut    |
| 4.2                  | (s) | Attribute:              | nom (valeurs) de l'attribut    |
| 4.3                  | (s) | Attribute:              | nom (valeurs) de l'attribut    |
| 5                    | (c) | Constraint:             | expression de contrainte       |
| 5.1                  | (o) | Attribute:              | nom (valeurs) de l'attribut    |
| 5.2                  | (f) | Attribute:              | nom (valeurs) de l'attribut    |
| 6                    | (o) | Attribute:              | nom (valeurs) de l'attribut    |
| 6.1                  | (s) | Attribute:              | nom (valeurs) de l'attribut    |
| 6.2                  | (s) | Attribute:              | nom (valeurs) de l'attribut    |
| <b>SERVICES:</b>     |     |                         |                                |
| 1                    | (f) | OpsService:             | nom du service                 |
| 2                    | (c) | Constraint:             | expression de contrainte       |
| 3.1                  | (f) | OpsService:             | nom du service                 |
| 3                    | (o) | MgtService:             | nom du service                 |

- (1) La rubrique "ASE FAL:" est le nom de l'élément ASE de la couche FAL (ASE FAL) qui fournit les services pour la classe spécifiée.
- (2) La rubrique "CLASS:" est le nom de la classe spécifiée. Tous les objets définis à l'aide de ce modèle sont une instance de cette classe. La classe peut être spécifiée par le présent document ou par un utilisateur du présent document.
- (3) La rubrique "CLASS ID:" est un numéro qui identifie la classe spécifiée. Ce numéro est unique au sein de l'ASE FAL qui fournit les services pour cette classe. Lorsqu'il est qualifié par l'identité de son ASE FAL, il identifie sans ambiguïté la classe relevant du domaine d'application de la FAL. La valeur "NULL" indique que la classe ne peut pas être instanciée. Les ID de classe compris entre 1 et 99, 240 et 767 sont réservés par le présent document pour identifier les classes normalisées. Les ID de classe compris entre 100 et 199, 768 et 1 279 sont attribués pour identifier les classes définies par l'utilisateur.
- (4) La rubrique "PARENT CLASS:" est le nom de la classe parent pour la classe spécifiée. Tous les attributs définis pour la classe parent et hérités par celle-ci sont hérités pour la classe définie, et n'ont donc pas à être redéfinis dans le modèle pour cette classe.

NOTE La classe parent "TOP" indique que la classe définie est une définition de classe initiale. La classe parent "TOP" est utilisée comme point de départ à partir duquel toutes les autres classes sont définies. L'usage de "TOP" est réservé pour les classes définies par le présent document.

- (5) L'étiquette "ATTRIBUTES" (ATTRIBUTS) indique que les entrées suivantes sont des attributs définis pour la classe:
  - a) chacune des entrées d'attribut contient un numéro de ligne dans la colonne 1; un indicateur obligatoire (o) / facultatif (f) / conditionnel (c) / sélecteur (s) dans la colonne 2; une étiquette de type d'attribut dans la colonne 3; un nom ou une expression conditionnelle dans la colonne 4; et, facultativement, une liste de valeurs énumérées dans la colonne 5. Dans la colonne suivant la liste de valeurs, la valeur par défaut pour l'attribut peut être spécifiée;
  - b) les objets sont normalement identifiés par un identificateur numérique et/ou par un nom d'objet. Dans les modèles de classe, ces attributs clés sont définis sous l'attribut clé;
  - c) le numéro de ligne définit la séquence et le niveau d'imbrication de la ligne. Chaque niveau d'imbrication est identifié par un point. L'imbrication est utilisée pour spécifier:
    - i) des champs d'un attribut structuré (4.1, 4.2, 4.3);
    - ii) des attributs conditionnés à un énoncé de contrainte (5). Les attributs peuvent être obligatoires (5.1) ou facultatifs (5.2) si la contrainte est vraie. Tous les attributs facultatifs n'exigent pas des énoncés de contraintes comme l'impose l'attribut défini en (5.2);
    - iii) les champs de sélection d'un attribut de type choix (6.1 et 6.2).
- (6) L'étiquette "SERVICES" indique que les entrées suivantes sont des services définis pour la classe:
  - a) un (o) dans la colonne 2 indique que le service est obligatoire pour la classe, alors qu'un (f) indique qu'il est facultatif. Un (c) dans cette colonne indique que le service est conditionnel. Lorsque tous les services définis pour une classe le sont comme étant facultatifs, au moins un est à sélectionner quand une instance de la classe est définie;
  - b) l'étiquette "OpsService" désigne un service opérationnel (1);
  - c) l'étiquette "MgtService" désigne un service de gestion (2);
  - d) le numéro de ligne définit la séquence et le niveau d'imbrication de la ligne. Chaque niveau d'imbrication est identifié par un point. L'imbrication dans la liste de services sert à spécifier des services conditionnés à un énoncé de contrainte.

### 3.4.3 Conventions pour les définitions de service

#### 3.4.3.1 Généralités

Le modèle de service, les primitives de service et les schémas de temps-séquence utilisés sont des descriptions totalement abstraites; ils ne constituent pas une spécification pour une mise en œuvre.

#### 3.4.3.2 Paramètres de service

Les primitives de service sont utilisées pour représenter les interactions entre utilisateur de service et fournisseur de service (ISO/IEC 10731). Elles acheminent des paramètres qui indiquent des informations disponibles dans l'interaction d'utilisateur ou de fournisseur. Dans n'importe quelle interface particulière, tous les paramètres ne doivent pas être énoncés de façon explicite.

Les spécifications de service selon le présent document utilisent un format de tableau pour décrire les paramètres de composants des primitives du service d'ASE. Les paramètres qui s'appliquent à chaque groupe de primitives de service sont consignés en tableaux. Chaque tableau comporte jusqu'à cinq colonnes pour:

- a) le nom de paramètre;
- b) la primitive "request";
- c) la primitive "indication";
- d) la primitive "response"; et
- e) la primitive "confirm".

Un paramètre (ou un composant de celui-ci) est énuméré dans chaque ligne de chaque tableau. Dans les colonnes appropriées de la primitive de service, un code est utilisé pour spécifier le type d'usage du paramètre sur la primitive spécifiée dans la colonne:

- O Le paramètre est obligatoire pour la primitive;
- U Le paramètre est une option de l'utilisateur et peut être fourni ou non, en fonction de l'utilisation dynamique de l'utilisateur de service. Lorsqu'il n'est pas fourni, une valeur par défaut est supposée pour le paramètre;
- (vide) Le paramètre n'est jamais présent;
- S Le paramètre est un élément sélectionné.

Certaines entrées sont en plus qualifiées par des éléments entre parenthèses. Ceux-ci peuvent être:

- a) une contrainte spécifique au paramètre:  
"(") indique que le paramètre équivaut du point de vue de la sémantique au paramètre dans la primitive de service située immédiatement à sa gauche dans le tableau;
- b) une indication qu'une certaine note s'applique à l'article:  
"(n)" indique que la note "n" suivante contient des informations supplémentaires relatives au paramètre et à son utilisation.

#### 3.4.3.3 Procédures de service

Les procédures sont définies en termes des

- interactions entre entités d'application par l'échange d'unités de données de protocole d'application de bus de terrain, et
- interactions entre un fournisseur de service de couche application et un utilisateur de service de couche application dans le même système par l'invocation de primitives de service de couche application.

Ces procédures sont applicables à des instances de communication entre systèmes qui prennent en charge des services de communications à contrainte temporelle au sein de la couche Application de bus de terrain.

## 4 Concepts

Les concepts et modèles fondamentaux utilisés pour décrire le service de couche application dans le présent document suivent l'IEC 61158-1, Article 9.

## 5 ASE de type de données

### 5.1 Généralités

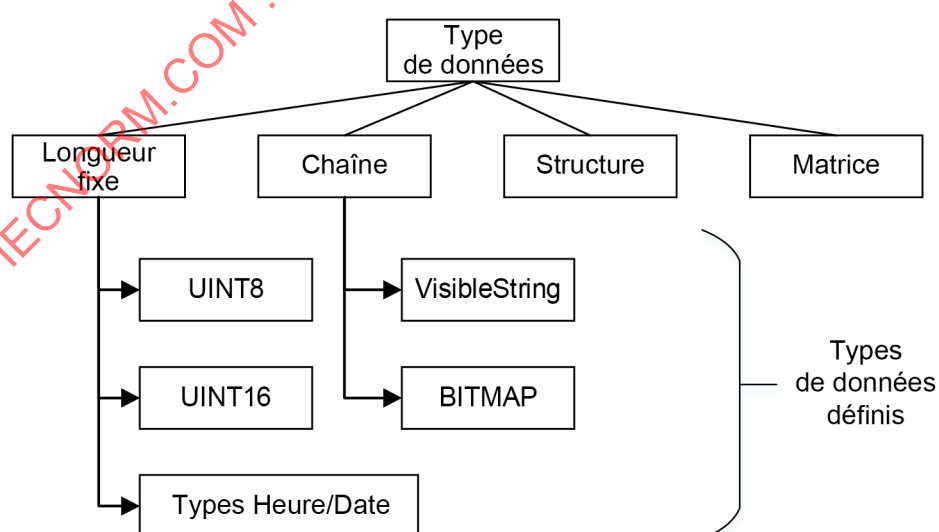
#### 5.1.1 Vue d'ensemble

Dans la FAL de type 28 de communication, l'ASE de type de données satisfait à l'IEC 61158-1, Article 10, et les autres types de données spécifiques au protocole ne sont pas définis dans le présent document.

Les types de données de bus de terrain précisent la syntaxe indépendante de la machine pour les données transportées par les services de chaque couche. La couche application de bus de terrain prend en charge la définition et le transfert des types de données de base et construits.

Les types de base sont des types de données atomiques qui ne peuvent pas être décomposés en types plus élémentaires. Les types de données construits sont composés de types de base et d'autres types construits. Leur complexité et la profondeur de l'imbrication ne sont pas contraintes par le présent document.

Les types de données sont définis sous la forme d'instances de classe de type de données, comme cela est représenté à la Figure 1. Seul un sous-ensemble des types de données définis dans l'Article 5 est représenté à la Figure 1. La définition de nouveaux types est réalisée en fournissant un nouvel ID numérique et en fournissant des valeurs aux attributs définis pour la classe de type de données.



IEC

Figure 1 – Exemple de hiérarchie de classe de type de données

Les définitions de type de données à la Figure 1 sont représentées sous la forme d'une structure classe/format/instance commençant par la classe de type de données intitulée "type de données". Les formats des types de données sont définis par la classe de type de données et représentés à la Figure 1.

Les classes de données de base permettent de définir les types de données de longueur fixe et de chaîne binaire. Les types normalisés issus de l'ISO/IEC 8824 sont appelés types de données simples. D'autres types de données de base normalisés sont destinés spécifiquement aux applications de bus de terrain et sont appelés types spécifiques.

Les types construits spécifiés dans le présent document sont des chaînes, des matrices et des structures. Aucun type normalisé n'est défini pour les matrices et les structures.

### **5.1.2 Vue d'ensemble des types de base**

La plupart des types de base sont définis à partir d'un ensemble de types ISO/IEC 8824 (types simples). Certains types ISO/IEC 8824 ont été étendus pour une utilisation spécifique au bus de terrain (types spécifiques).

Les types simples sont des types universels ISO/IEC 8824. Ils sont définis dans le présent document pour leur fournir des identificateurs de classe de bus de terrain.

Les types spécifiques sont ceux définis spécifiquement pour l'environnement de bus de terrain. Ils sont définis comme des sous-types de classe simples.

Les types de base présentent une longueur constante. Deux variantes sont définies, l'une pour définir les types de données dont la longueur est un nombre entier d'octets, et l'autre pour définir les types de données dont la longueur est exprimée en nombre de bits.

NOTE Boolean, Integer, OctetString, VisibleString et UniversalTime sont définis dans le présent document pour leur attribuer des identificateurs de classe de bus de terrain. Le présent document ne modifie pas leur définition comme cela est indiqué dans l'ISO/IEC 8824.

### **5.1.3 Vue d'ensemble des types de longueurs fixes**

La longueur des types de longueurs fixes est un nombre entier d'octets.

### **5.1.4 Vue d'ensemble des types construits**

#### **5.1.4.1 Chaînes**

Une chaîne est composée d'un ensemble ordonné, variable en nombre, d'éléments de longueur fixe saisis de manière homogène.

#### **5.1.4.2 Matrices**

Une matrice est composée d'un ensemble ordonné d'éléments saisis de manière homogène. Le type de données des éléments de matrice placés ne fait l'objet d'aucune restriction, mais chaque élément doit être de même type. Une fois défini, le nombre d'éléments d'une matrice peut être modifié ou non.

#### **5.1.4.3 Structures**

Une structure est composée d'un ensemble ordonné d'éléments saisis de manière hétérogène appelés "champs". Comme pour les matrices, le présent document ne limite pas le type de données des champs. Toutefois, il n'est pas nécessaire que les types de champs d'une structure soient identiques.

#### 5.1.4.4 Niveau d'imbrication

Le présent document permet aux matrices et structures de contenir d'autres matrices et structures. Il n'impose aucune restriction quant au nombre de niveaux d'imbrication admis.

Lorsqu'une matrice ou une structure contient des éléments construits, l'accès à l'un d'eux dans son intégralité est assuré. L'accès aux sous-éléments de l'élément construit est également assuré.

## 5.2 Types de données définis d'AL

### 5.2.1 Types de longueur fixe

#### 5.2.1.1 Types Date/Heure

##### 5.2.1.1.1 Type temps (TIMEV)

Le type de données de temps est un entier non signé de 64 bits, qui représente le temps incrémenté de 1 ns. Le Tableau 1 indique le codage.

**Tableau 1 – Codage des données de type TIMEV**

| octet | bit      |          |          |          |          |          |          |          |
|-------|----------|----------|----------|----------|----------|----------|----------|----------|
|       | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |
| 1     | $2^{63}$ | $2^{62}$ | $2^{61}$ | $2^{60}$ | $2^{59}$ | $2^{58}$ | $2^{57}$ | $2^{56}$ |
| 2     | $2^{55}$ | $2^{54}$ | $2^{53}$ | $2^{52}$ | $2^{51}$ | $2^{50}$ | $2^{49}$ | $2^{48}$ |
| 3     | $2^{47}$ | $2^{46}$ | $2^{45}$ | $2^{44}$ | $2^{43}$ | $2^{42}$ | $2^{41}$ | $2^{40}$ |
| 4     | $2^{39}$ | $2^{38}$ | $2^{37}$ | $2^{36}$ | $2^{35}$ | $2^{34}$ | $2^{33}$ | $2^{32}$ |
| 5     | $2^{31}$ | $2^{30}$ | $2^{29}$ | $2^{28}$ | $2^{27}$ | $2^{26}$ | $2^{25}$ | $2^{24}$ |
| 6     | $2^{23}$ | $2^{22}$ | $2^{21}$ | $2^{20}$ | $2^{19}$ | $2^{18}$ | $2^{17}$ | $2^{16}$ |
| 7     | $2^{15}$ | $2^{14}$ | $2^{13}$ | $2^{12}$ | $2^{11}$ | $2^{10}$ | $2^9$    | $2^8$    |
| 8     | $2^7$    | $2^6$    | $2^5$    | $2^4$    | $2^3$    | $2^2$    | $2^1$    | $2^0$    |

#### CLASS:

#### TYPE DE DONNÉES

#### ATTRIBUTS:

|     |                                             |                |
|-----|---------------------------------------------|----------------|
| 1   | Identificateur numérique de type de données | = 16           |
| 2   | Nom de type de données                      | = TIMEV        |
| 3   | Format                                      | = FIXED LENGTH |
| 4.1 | Longueur en octets                          | = 8            |

##### 5.2.1.1.2 Codage du type Date (TIMEDATE)

La longueur du type de données TIMEDATE est de sept octets, indiquant la date et l'heure. Le Tableau 2 représente la signification de chaque octet.

**Tableau 2 – Codage du type TIMEDATE**

| octet | Bit             |                 |                 |                 |                 |                 |                |                | Description                                                                                                                                                        |
|-------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|----------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | 7               | 6               | 5               | 4               | 3               | 2               | 1              | 0              |                                                                                                                                                                    |
| 1     | 2 <sup>15</sup> | 2 <sup>14</sup> | 2 <sup>13</sup> | 2 <sup>12</sup> | 2 <sup>11</sup> | 2 <sup>10</sup> | 2 <sup>9</sup> | 2 <sup>8</sup> | Indique la valeur en millisecondes inférieure à une minute                                                                                                         |
| 2     | 2 <sup>7</sup>  | 2 <sup>6</sup>  | 2 <sup>5</sup>  | 2 <sup>4</sup>  | 2 <sup>3</sup>  | 2 <sup>2</sup>  | 2 <sup>1</sup> | 2 <sup>0</sup> |                                                                                                                                                                    |
| 3     | 2 <sup>7</sup>  | 2 <sup>6</sup>  | 2 <sup>5</sup>  | 2 <sup>4</sup>  | 2 <sup>3</sup>  | 2 <sup>2</sup>  | 2 <sup>1</sup> | 2 <sup>0</sup> | Indique la valeur en minutes inférieure à une heure                                                                                                                |
| 4     | 2 <sup>7</sup>  | 2 <sup>6</sup>  | 2 <sup>5</sup>  | 2 <sup>4</sup>  | 2 <sup>3</sup>  | 2 <sup>2</sup>  | 2 <sup>1</sup> | 2 <sup>0</sup> | Indique les valeurs en heures inférieures à un jour                                                                                                                |
| 5     | 2 <sup>2</sup>  | 2 <sup>1</sup>  | 2 <sup>0</sup>  | 2 <sup>4</sup>  | 2 <sup>3</sup>  | 2 <sup>2</sup>  | 2 <sup>1</sup> | 2 <sup>0</sup> | Les 3 premiers bits signifient le jour de la semaine, et les 5 derniers bits sont le jour du mois en cours.                                                        |
| 6     | 2 <sup>7</sup>  | 2 <sup>6</sup>  | 2 <sup>5</sup>  | 2 <sup>4</sup>  | 2 <sup>3</sup>  | 2 <sup>2</sup>  | 2 <sup>1</sup> | 2 <sup>0</sup> | Indique le mois                                                                                                                                                    |
| 7     | 2 <sup>7</sup>  | 2 <sup>6</sup>  | 2 <sup>5</sup>  | 2 <sup>4</sup>  | 2 <sup>3</sup>  | 2 <sup>2</sup>  | 2 <sup>1</sup> | 2 <sup>0</sup> | Indique l'année. La plage de valeurs de 0 à 50 correspond à une plage de 2 000 à 2 050, et la plage de valeurs de 51 à 99 correspond à une plage de 1 951 à 1 999. |

**CLASS:**

**TYPE DE DONNÉES**

**ATTRIBUTS:**

|     |                                             |                |
|-----|---------------------------------------------|----------------|
| 1   | Identificateur numérique de type de données | = 17           |
| 2   | Nom de type de données                      | = TIMEDATE     |
| 3   | Format                                      | = FIXED LENGTH |
| 4.1 | Longueur en octets                          | = 7            |

#### 5.2.1.1.3 Codage du type Heure du jour (TIMEOFDAY)

La longueur du type de données TIMEOFDAY est de six octets et indique l'heure du jour. Le Tableau 3 représente la signification de chaque octet.

**Tableau 3 – Codage du type TIMEOFDAY**

| octet | Bit             |                 |                 |                 |                 |                 |                 |                 | Description                                                                           |
|-------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|---------------------------------------------------------------------------------------|
|       | 7               | 6               | 5               | 4               | 3               | 2               | 1               | 0               |                                                                                       |
| 1     | 2 <sup>31</sup> | 2 <sup>30</sup> | 2 <sup>29</sup> | 2 <sup>28</sup> | 2 <sup>27</sup> | 2 <sup>26</sup> | 2 <sup>25</sup> | 2 <sup>24</sup> | Indique le nombre de nanosecondes à partir de 0:00 minuit                             |
| 2     | 2 <sup>23</sup> | 2 <sup>22</sup> | 2 <sup>21</sup> | 2 <sup>20</sup> | 2 <sup>19</sup> | 2 <sup>18</sup> | 2 <sup>17</sup> | 2 <sup>16</sup> |                                                                                       |
| 3     | 2 <sup>15</sup> | 2 <sup>14</sup> | 2 <sup>13</sup> | 2 <sup>12</sup> | 2 <sup>11</sup> | 2 <sup>10</sup> | 2 <sup>9</sup>  | 2 <sup>8</sup>  |                                                                                       |
| 4     | 2 <sup>7</sup>  | 2 <sup>6</sup>  | 2 <sup>5</sup>  | 2 <sup>4</sup>  | 2 <sup>3</sup>  | 2 <sup>2</sup>  | 2 <sup>1</sup>  | 2 <sup>0</sup>  |                                                                                       |
| 5     | 2 <sup>15</sup> | 2 <sup>14</sup> | 2 <sup>13</sup> | 2 <sup>12</sup> | 2 <sup>11</sup> | 2 <sup>10</sup> | 2 <sup>9</sup>  | 2 <sup>8</sup>  | Indique le nombre de jours qui ont été comptés depuis le 1 <sup>er</sup> janvier 1984 |
| 6     | 2 <sup>7</sup>  | 2 <sup>6</sup>  | 2 <sup>5</sup>  | 2 <sup>4</sup>  | 2 <sup>3</sup>  | 2 <sup>2</sup>  | 2 <sup>1</sup>  | 2 <sup>0</sup>  |                                                                                       |

**CLASS:**

**TYPE DE DONNÉES**

**ATTRIBUTS:**

|     |                                             |                |
|-----|---------------------------------------------|----------------|
| 1   | Identificateur numérique de type de données | = 18           |
| 2   | Nom de type de données                      | = TIMEOFDAY    |
| 3   | Format                                      | = FIXED LENGTH |
| 4.1 | Longueur en octets                          | = 6            |

#### 5.2.1.1.4 Codage du type Différence de temps (TIMEDIFFER)

La longueur de type de données TIMEDIFFER est de six octets et indique la différence de temps. Le Tableau 4 représente la signification de chaque octet.

**Tableau 4 – Codage du type TIMEDIFFER**

| octet | Bit      |          |          |          |          |          |          |          | Description                                                        |
|-------|----------|----------|----------|----------|----------|----------|----------|----------|--------------------------------------------------------------------|
|       | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |                                                                    |
| 1     | $2^{31}$ | $2^{30}$ | $2^{29}$ | $2^{28}$ | $2^{27}$ | $2^{26}$ | $2^{25}$ | $2^{24}$ | Indique l'écart des valeurs en nanosecondes inférieures à un jour. |
| 2     | $2^{23}$ | $2^{22}$ | $2^{21}$ | $2^{20}$ | $2^{19}$ | $2^{18}$ | $2^{17}$ | $2^{16}$ |                                                                    |
| 3     | $2^{15}$ | $2^{14}$ | $2^{13}$ | $2^{12}$ | $2^{11}$ | $2^{10}$ | $2^9$    | $2^8$    |                                                                    |
| 4     | $2^7$    | $2^6$    | $2^5$    | $2^4$    | $2^3$    | $2^2$    | $2^1$    | $2^0$    |                                                                    |
| 5     | $2^{15}$ | $2^{14}$ | $2^{13}$ | $2^{12}$ | $2^{11}$ | $2^{10}$ | $2^9$    | $2^8$    | Indique le nombre de jours d'écart                                 |
| 6     | $2^7$    | $2^6$    | $2^5$    | $2^4$    | $2^3$    | $2^2$    | $2^1$    | $2^0$    |                                                                    |

#### CLASS:

#### TYPE DE DONNÉES

#### ATTRIBUTS:

|     |                                             |                |
|-----|---------------------------------------------|----------------|
| 1   | Identificateur numérique de type de données | = 19           |
| 2   | Nom de type de données                      | = TIMEDIFFER   |
| 3   | Format                                      | = FIXED LENGTH |
| 4.1 | Longueur en octets                          | = 6            |

### 5.2.2 Type de chaîne

#### 5.2.2.1 Type chaîne visible (VisibleString)

Le type VISIBLESTRING est un ensemble d'octets dans l'ordre. La longueur n'est pas fixe et elle est de 1 octet à N octets. Le Tableau 5 représente le codage de chaîne. Pour les chaînes de N octets, le MSB de l'octet valide le plus élevé de ce type de données est transmis en premier.

**Tableau 5 – Codage des données de type VISIBLESTRING**

| octet | bit        |             |             |             |             |             |             |             |
|-------|------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|       | 7          | 6           | 5           | 4           | 3           | 2           | 1           | 0           |
| 1     | $2^{8N-1}$ | $2^{8N-2}$  | $2^{8N-3}$  | $2^{8N-4}$  | $2^{8N-5}$  | $2^{8N-6}$  | $2^{8N-7}$  | $2^{8N-8}$  |
| 2     | $2^{8N-9}$ | $2^{8N-10}$ | $2^{8N-11}$ | $2^{8N-12}$ | $2^{8N-13}$ | $2^{8N-14}$ | $2^{8N-15}$ | $2^{8N-16}$ |
| ...   | ...        | ...         | ...         | ...         | ...         | ...         | ...         | ...         |
| N     | $2^7$      | $2^6$       | $2^5$       | $2^4$       | $2^3$       | $2^2$       | $2^1$       | $2^0$       |

#### CLASS:

#### TYPE DE DONNÉES

#### ATTRIBUTS:

|     |                                             |                 |
|-----|---------------------------------------------|-----------------|
| 1   | Identificateur numérique de type de données | = 32            |
| 2   | Nom de type de données                      | = VISIBLESTRING |
| 3   | Format                                      | = STRING        |
| 4.1 | Longueur en octets                          | = 1 à n         |

### 5.2.2.2 Codage de type Bitmap (BITMAP)

La valeur de la longueur du type BITMAP varie de 2 octets à 256 octets, y compris les informations relatives à la largeur et à Mask\_Info.

Le Figure 2 représente le codage du type bitmap.

NOTE La largeur est de 1 octet, sa valeur indique le nombre d'octets de Mask\_Info.

NOTE Mask\_Info représenté par bit. La longueur d'octet est la valeur de largeur et la longueur en bits est égale à la largeur × 8.

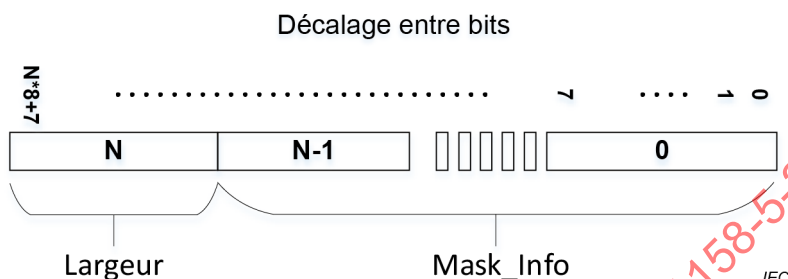


Figure 2 – Type de données BITMAP

CLASS:

TYPE DE DONNÉES

ATTRIBUTS:

|     |                                             |          |
|-----|---------------------------------------------|----------|
| 1   | Identificateur numérique de type de données | = 33     |
| 2   | Nom de type de données                      | = BITMAP |
| 3   | Format                                      | = STRING |
| 4.1 | Longueur en octets                          | = 1 à n  |

### 5.2.3 Type de structure

#### 5.2.3.1 Structure d'information de temps

- Nom de la structure: TIMEINFO\_S
- Nombre de membres de la structure: 5
- Description des membres de la structure: voir le Tableau 6.

Tableau 6 – Description de la structure d'information de temps TIMEINFO\_S

| Nom du membre | Type de données | Description                                                                                                                                                                                                                                                                                                                                                                             |
|---------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Time_fmt      | UINT8           | Identifier les bits valides de trois formats de temps basés respectivement sur les bits, bit0, bit1 et bit2. Déterminer si les valeurs de temps sont valides en fonction de l'heure, de la minute et de la seconde, de la milliseconde et de la microseconde et de la seconde. Lorsque la position correspondante est 1, cela signifie qu'elle est valide, sinon elle n'est pas valide. |
| Time_ns       | TIMEV           | Comptage du temps basé sur les nanosecondes.                                                                                                                                                                                                                                                                                                                                            |
| TimeDate      | TIMEDATE        | Convertir en valeur de date de l'année, mois, jour, minute, minute, seconde et la précision est la milliseconde.                                                                                                                                                                                                                                                                        |
| TimeOfDay     | TIMEOFDAY       | Le nombre de jours basé sur les nanosecondes.                                                                                                                                                                                                                                                                                                                                           |
| TimeDiffer    | TIMEDIFFER      | Valeur de décalage temporel basée sur les nanosecondes.                                                                                                                                                                                                                                                                                                                                 |

**CLASS:****TYPE DE DONNEES****ATTRIBUTS:**

|        |                          |              |
|--------|--------------------------|--------------|
| 2      | Nom de type de données   | = TIMEINFO_S |
| 5      | Format                   | = STRUCTURE  |
| 5.1    | Nombre de champs         | = 5          |
| 5.2    | Liste des champs         |              |
| 5.2.1  | Nom de champ             | = Time_fmt   |
| 5.2.2  | Type de données de champ | UINT8        |
| 5.2.3  | Nom de champ             | = Time_ns    |
| 5.2.4  | Type de données de champ | TIMEV        |
| 5.2.5  | Nom de champ             | = TimeDate   |
| 5.2.6  | Type de données de champ | TIMEDATE     |
| 5.2.7  | Nom de champ             | = TimeOfDay  |
| 5.2.8  | Type de données de champ | TIMEOFDAY    |
| 5.2.9  | Nom de champ             | = TimeDiffer |
| 5.2.10 | Type de données de champ | TIMEDIFFER   |

**5.2.3.2 Structure d'information d'option d'horloge**

- Nom de la structure: CLOCK\_OPTION\_INFO\_S
- Nombre de membres de la structure: 5
- Description des membres de la structure, voir le Tableau 7.

**Tableau 7 – Description de la structure d'information de l'option d'horloge  
CLOCK\_OPTION\_INFO\_S**

| Nom du membre | Type de données | Description                                                                                                            |
|---------------|-----------------|------------------------------------------------------------------------------------------------------------------------|
| Tid_cnt       | UINT8           | Nombre d'horloges sur l'appareil ou le module                                                                          |
| Tid_val       | UINT8           | Valeur de l'ID d'horloge                                                                                               |
| Tid_attr      | UINT8           | L'attribut de champ de commande de l'horloge, 0 est l'horloge de commande (liste), 1 est l'horloge de champ (liste)    |
| Tcd_lst       | UINT8           | Instruction TCD pour l'horloge correspondante, la plage de valeurs de TCD est identique à l'IEC 61158-6-28, Article 4. |
| Tid_errcode   | UINT8           | Une liste d'encodage d'exception d'erreur d'horloge                                                                    |

**CLASS:****TYPE DE DONNEES****ATTRIBUTS:**

|        |                          |                       |
|--------|--------------------------|-----------------------|
| 2      | Nom de type de données   | = CLOCK_OPTION_INFO_S |
| 5      | Format                   | = STRUCTURE           |
| 5.1    | Nombre de champs         | = 5                   |
| 5.2    | Liste des champs         |                       |
| 5.2.1  | Nom de champ             | = Tid_cnt             |
| 5.2.2  | Type de données de champ | UINT8                 |
| 5.2.3  | Nom de champ             | = Tid_val             |
| 5.2.4  | Type de données de champ | UINT8                 |
| 5.2.5  | Nom de champ             | = Tid_attr            |
| 5.2.6  | Type de données de champ | UINT8                 |
| 5.2.7  | Nom de champ             | = Tcd_lst             |
| 5.2.8  | Type de données de champ | UINT8                 |
| 5.2.9  | Nom de champ             | = Tid_errcode         |
| 5.2.10 | Type de données de champ | UINT8                 |

### 5.2.3.3 Structure des paramètres de configuration du réseau

- Nom de la structure: NETWORK\_CFG\_PARA\_S
- Nombre de membres de la structure: 15
- Description des membres de la structure: voir le Tableau 8.

**Tableau 8 – Description de la structure des paramètres de configuration du réseau  
NETWORK\_CFG\_PARA\_S**

| Nom du membre      | Type de données | Description                                                                                                                                                                                                                                                                                                |
|--------------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Network_id         | UINT32          | Identifier le NETWORKID (voir en 6.6.6.1.2) de l'appareil ou du module                                                                                                                                                                                                                                     |
| Ip_switch          | UINT8           | Prise en charge ou non de l'adresse IPv4 (définie dans l'IETF RFC 791) ou IPv6 (définie dans l'IETF RFC 2460) par bit.<br>Bit 0: prise en charge ou non d'IPv4;<br>Bit 1: prise en charge ou non d'IPv6, les autres bits ne sont pas encore utilisés; le bit correspondant = 1 indique la prise en charge. |
| Multicast_id0 à 15 | UINT32          | NETWORKID du type de multidiffusion, groupe de multidiffusion auquel appartient l'appareil ou le module identifié. Chaque appareil ou module appartient à un maximum de 16 groupes de multidiffusion.                                                                                                      |
| Mac_addr           | ARRAY           | Adresse MAC unique (définie dans l'ISO/IEC/IEEE 8802-3) de l'appareil ou du module                                                                                                                                                                                                                         |
| Nodeid             | UINT8           | Adresse ID de nœud de l'appareil                                                                                                                                                                                                                                                                           |
| Dctime             | UINT16          | Temps de cycle de calcul des données                                                                                                                                                                                                                                                                       |
| Bctime             | UINT16          | Temps de cycle du service de bus                                                                                                                                                                                                                                                                           |
| Wctime             | UINT16          | Temps de cycle d'écriture des données de service de bus                                                                                                                                                                                                                                                    |
| Rctime             | UINT16          | Temps de cycle de lecture de données de service de bus                                                                                                                                                                                                                                                     |
| Ctime              | UINT16          | Temps de cycle de concurrence de service de bus                                                                                                                                                                                                                                                            |
| Totcnt             | UINT16          | Nombre de temporisations autorisées par l'appareil ou le module                                                                                                                                                                                                                                            |
| Rt_data_addr       | UINT32          | Adresse de début du tampon de données RT                                                                                                                                                                                                                                                                   |
| Rt_data_size       | UINT32          | Taille du tampon de données RT                                                                                                                                                                                                                                                                             |
| nRt_data_addr      | UINT32          | Adresse de début du tampon nRT                                                                                                                                                                                                                                                                             |
| nRt_data_size      | UINT32          | Taille du tampon de données nRT                                                                                                                                                                                                                                                                            |

#### CLASS:

#### TYPE DE DONNEES

#### ATTRIBUTES:

|        |                          |                                  |
|--------|--------------------------|----------------------------------|
| 2      | Nom de type de données   | = NETWORK_CFG_PARA_S             |
| 5      | Format                   | = STRUCTURE                      |
| 5.1    | Nombre de champs         | = 15                             |
| 5.2    | Liste des champs         |                                  |
| 5.2.1  | Nom de champ             | = Network_id                     |
| 5.2.2  | Type de données de champ | UINT32                           |
| 5.2.3  | Nom de champ             | = Ip_switch                      |
| 5.2.4  | Type de données de champ | UINT8                            |
| 5.2.5  | Nom de champ             | = Multicast_id0 à Multicast_id15 |
| 5.2.6  | Type de données de champ | UINT8                            |
| 5.2.7  | Nom de champ             | = Mac_addr                       |
| 5.2.8  | Type de données de champ | ARRAY                            |
| 5.2.9  | Nom de champ             | = Nodeid                         |
| 5.2.10 | Type de données de champ | UINT8                            |
| 5.2.11 | Nom de champ             | = Dctime                         |

|        |                                      |                 |
|--------|--------------------------------------|-----------------|
| 5.2.12 | Type de données de champ             | UINT16          |
| 5.2.13 | Nom de champ                         | = Bctime        |
| 5.2.14 | Type de données de champ             | UINT16          |
| 5.2.15 | Nom de champ                         | = Wctime        |
| 5.2.16 | Type de données de champ             | UINT16          |
| 5.2.17 | Nom de champ                         | = Rctime        |
| 5.2.18 | Type de données de champ             | UINT16          |
| 5.2.19 | Nom de champ                         | = Ctime         |
| 5.2.20 | Type de données de champ             | UINT16          |
| 5.2.21 | Nom de champ                         | = Totcnt        |
| 5.2.22 | Type de données de champ             | UINT16          |
| 5.2.23 | Nom de champ                         | = Rt_data_addr  |
| 5.2.24 | Type de données de champ             | UINT32          |
| 5.2.25 | Nom de champ                         | = Rt_data_size  |
| 5.2.26 | Type de données de champ             | UINT32          |
| 5.2.27 | Nom de champ                         | = nRt_data_addr |
| 5.2.28 | Type de données de champ             | UINT32          |
| 5.2.29 | Nom de champ                         | = nRt_data_size |
| 5.2.30 | Type de données de champ             | UINT32          |
| 6      | Contrainte                           | Format = ARRAY  |
| 6.1    | Nombre d'éléments de matrice         | 6               |
| 6.2    | Type de données d'élément de matrice | UINT8           |

#### 5.2.3.4 Structure de table de correspondance MAC

- Nom de la structure: NETWORKID\_MAC\_MAP\_TABLE\_S
- Nombre de membres de la structure: 7
- Description des membres de la structure: voir le Tableau 9.

**Tableau 9 – Description de la table de correspondance MAC  
NETWORKID\_MAC\_MAP\_TABLE\_S**

| Nom du membre | Type de données | Description                                                                                                                                                                                                                                                                      |
|---------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Index         | UINT16          | Indice d'entrée                                                                                                                                                                                                                                                                  |
| NodeID        | UINT8           | Adresse du nœud                                                                                                                                                                                                                                                                  |
| RoleID        | UINT8           | Attribut de champ de commande de l'appareil ou du module. 0x00 est l'appareil de commande, 0x01 est l'appareil de terrain, 0x02 est indéterminé. Les autres valeurs sont réservées.                                                                                              |
| IP_Flag       | UINT8           | Le type d'IP pris en charge par l'appareil ou le module.<br>Le bit 0 identifie IPv4 (défini dans l'IETF RFC 791).<br>Le bit 1 identifie IPv6 (défini dans l'IETF RFC 2460).<br>La position de bit 1 correspondante indique la prise en charge.<br>Les autres bits sont réservés. |
| StatusID      | UINT8           | L'ID de l'état de l'appareil ou du module. La valeur 0x00 indique que l'état d'initialisation est OK, et la valeur 0x01 indique l'état de fonctionnement normal. Les autres valeurs sont réservées.                                                                              |
| Network_ID    | UINT32          | ID NETWORKID unique de l'appareil ou du module                                                                                                                                                                                                                                   |
| MacAddr       | ARRAY           | Adresse MAC (définie dans l'ISO/IEC/IEEE 88023-3) de l'appareil ou du module                                                                                                                                                                                                     |

**CLASS:**

**TYPE DE DONNÉES**

**ATTRIBUTS:**

|        |                                      |                             |
|--------|--------------------------------------|-----------------------------|
| 2      | Nom de type de données               | = NETWORKID_MAC_MAP_TABLE_S |
| 5      | Format                               | = STRUCTURE                 |
| 5.1    | Nombre de champs                     | = 7                         |
| 5.2    | Liste des champs                     |                             |
| 5.2.1  | Nom de champ                         | = Index                     |
| 5.2.2  | Type de données de champ             | UINT16                      |
| 5.2.3  | Nom de champ                         | = NodeID                    |
| 5.2.4  | Type de données de champ             | UINT8                       |
| 5.2.5  | Nom de champ                         | = RoleID                    |
| 5.2.6  | Type de données de champ             | UINT8                       |
| 5.2.7  | Nom de champ                         | = IP_Flag                   |
| 5.2.8  | Type de données de champ             | UINT8                       |
| 5.2.9  | Nom de champ                         | = StatusID                  |
| 5.2.10 | Type de données de champ             | UINT8                       |
| 5.2.11 | Nom de champ                         | = Network_ID                |
| 5.2.12 | Type de données de champ             | UINT32                      |
| 5.2.13 | Nom de champ                         | = MacAddr                   |
| 5.2.14 | Type de données de champ             | ARRAY                       |
| 6      | Contrainte                           | Format = ARRAY              |
| 6.1    | Nombre d'éléments de matrice         | = 6                         |
| 6.2    | Type de données d'élément de matrice | UINT8                       |

**5.2.3.5 Structure de table de correspondance**

- Nom de la structure: IP\_MAP\_TABLE\_S
- Nombre de membres de la structure: 9
- Description des membres de la structure: voir le Tableau 10.

**Tableau 10 – Description de la structure de table de correspondance IP  
IP\_MAP\_TABLE\_S**

| Nom du membre | Type de données | Description                                                                                                                                                                                                                     |
|---------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Index         | UINT16          | Indice d'entrée                                                                                                                                                                                                                 |
| Stated        | UINT8           | Valeur qui identifie l'état de l'entrée.<br>0x00 est invalide; 0x01 est valide; 0x02 est en attente.<br>Les autres valeurs sont réservées.                                                                                      |
| Network_id    | UINT32          | NETWORKID unique de l'appareil ou du module                                                                                                                                                                                     |
| ipv4_addr     | UINT32          | Adresse IPv4 (définie dans l'IETF RFC 791).                                                                                                                                                                                     |
| ipv4_mask     | UINT32          | Masque IPv4 (défini dans l'IETF RFC 791).                                                                                                                                                                                       |
| ipv6_addr     | UINT128         | Adresse IPv6 (définie dans l'IETF RFC 2460)                                                                                                                                                                                     |
| ipv6_mask     | UINT128         | Masque IPv6 (défini dans l'IETF RFC 2460)                                                                                                                                                                                       |
| Dmac_addr     | ARRAY           | Adresse MAC physique unique de l'appareil ou du module                                                                                                                                                                          |
| Vmac_addr     | ARRAY           | Adresse MAC virtuelle de l'appareil ou du module. Un appareil physique possède un ou plusieurs sous-appareils ou sous-modules, et chaque sous-appareil ou sous-module correspond de manière unique à une adresse MAC virtuelle. |

**CLASS: TYPE DE DONNÉES****ATTRIBUTS:**

|        |                                      |                  |
|--------|--------------------------------------|------------------|
| 2      | Nom de type de données               | = IP_MAP_TABLE_S |
| 5      | Format                               | = STRUCTURE      |
| 5.1    | Nombre de champs                     | = 9              |
| 5.2    | Liste des champs                     |                  |
| 5.2.1  | Nom de champ                         | = Index          |
| 5.2.2  | Type de données de champ             | UINT16           |
| 5.2.3  | Nom de champ                         | = Stated         |
| 5.2.4  | Type de données de champ             | UINT8            |
| 5.2.5  | Nom de champ                         | = Network_id     |
| 5.2.6  | Type de données de champ             | UINT32           |
| 5.2.7  | Nom de champ                         | = ipv4_addr      |
| 5.2.8  | Type de données de champ             | UINT32           |
| 5.2.9  | Nom de champ                         | = ipv4_mask      |
| 5.2.10 | Type de données de champ             | UINT32           |
| 5.2.11 | Nom de champ                         | = ipv6_addr      |
| 5.2.12 | Type de données de champ             | UINT128          |
| 5.2.13 | Nom de champ                         | = ipv6_mask      |
| 5.2.14 | Type de données de champ             | UINT128          |
| 5.2.15 | Nom de champ                         | = Dmac_addr      |
| 5.2.16 | Type de données de champ             | ARRAY            |
| 5.2.17 | Nom de champ                         | = Vmac_addr      |
| 5.2.18 | Type de données de champ             | ARRAY            |
| 6      | Contrainte                           | Format = ARRAY   |
| 6.1    | Nombre d'éléments de matrice         | 6                |
| 6.2    | Type de données d'élément de matrice | UINT8            |
| 6.3    | Nombre d'éléments de matrice         | 6                |
| 6.4    | Type de données d'élément de matrice | UINT8            |

**6 Spécification du modèle de communication de type 28****6.1 Généralités**

L'AL de type 28 traite les services d'application RT et nRT sur la base de la DLL de type 28 et la PhL, ce qui offre trois types de fonctions de service: service de temps, service de données et service de gestion système. La position de l'AL dans l'architecture de la pile de protocole de type 28 est représentée à la Figure 3.

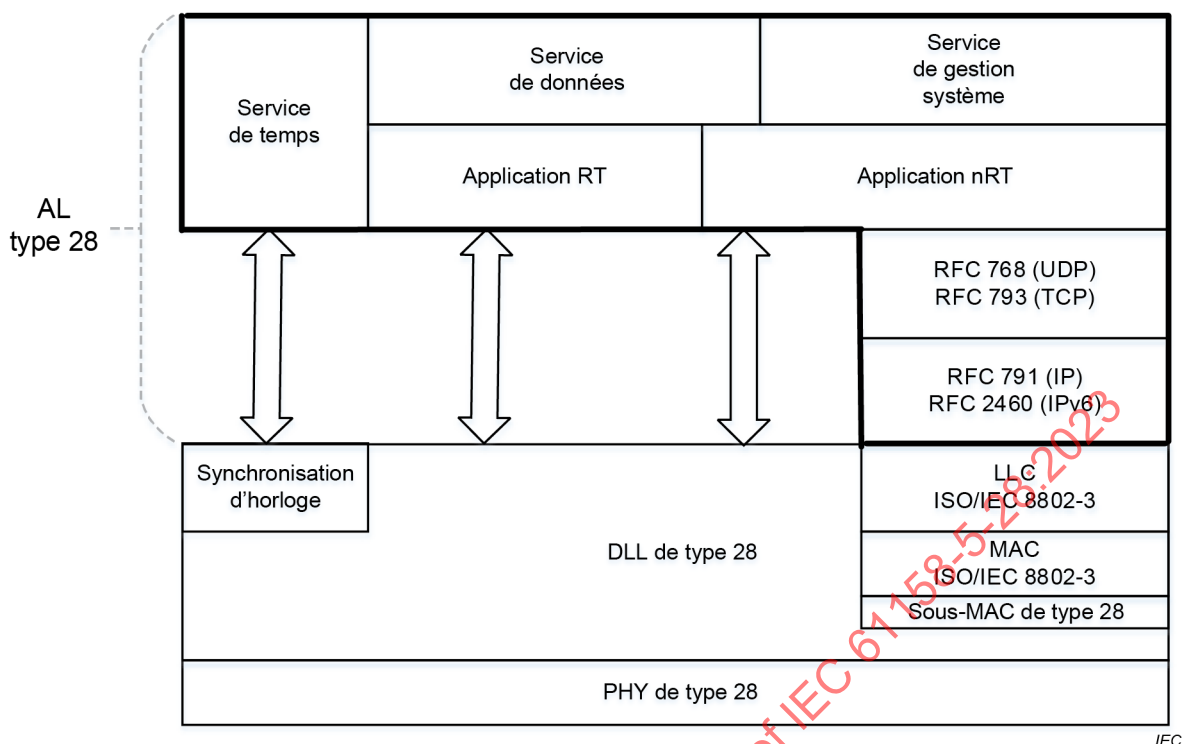


Figure 3 – AL dans l'architecture de la pile de protocole de bus de terrain

## 6.2 Pile de protocole pour le bus de terrain de type 28

La Figure 4 représente l'architecture du protocole AL de type 28.

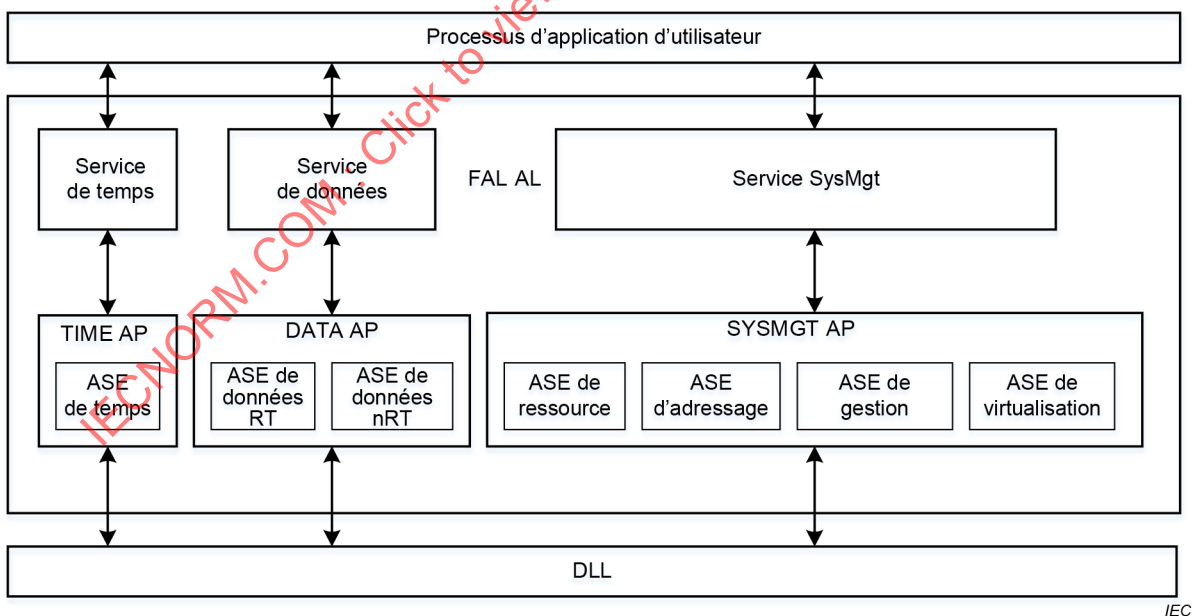


Figure 4 – Architecture de protocole AL de type 28

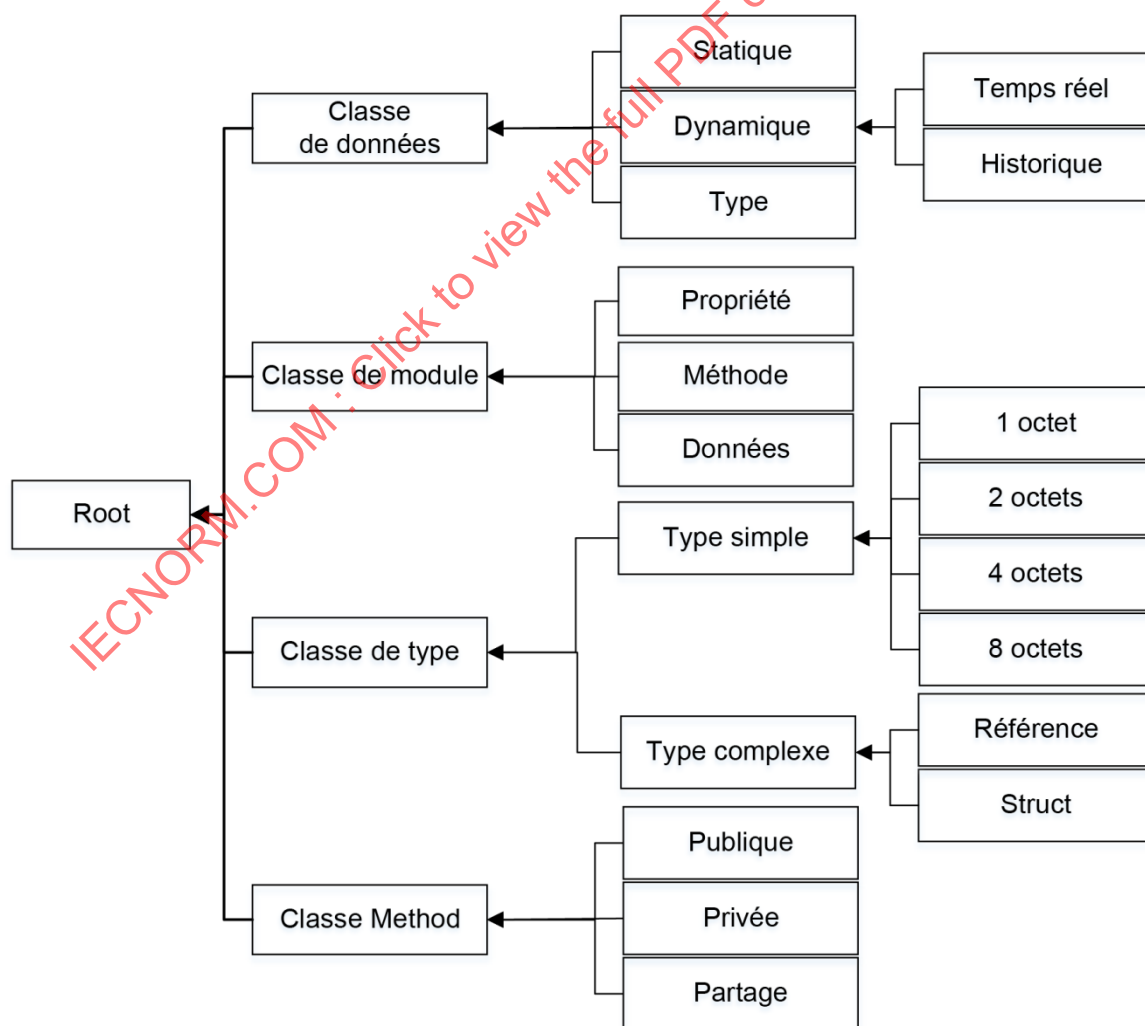
L'AL de type 28 fournit aux utilisateurs un service de temps, un service de données et un service de gestion système, qui sont mis en œuvre par l'élément de service d'application (ASE) du processus d'application (AP) correspondant. Le service de temps met en œuvre la synchronisation temporelle et l'interrogation du temps sur la base de l'ASE de temps. Le service de données traite les données avec différentes exigences relatives au temps réel basées sur l'ASE de données RT et l'ASE de données nRT. Le service de gestion système

fournit au système de réseau un service d'application de ressources, des services d'application d'adressage, des services d'application de gestion et des services d'application de virtualisation. Le protocole AL de type 28 met en œuvre les fonctions de service AL spécifiques par le biais de la DLL de type 28 et des interfaces de service fournies par la couche inférieure.

### 6.3 Modélisation de la correspondance des données

Dans le réseau de type 28, il existe une structure de tampon de données de service sur l'appareil ou le module avec un tampon de données RT ou le tampon de données nRT basé sur le RT du traitement de données de service. Le tampon de données RT distingue le tampon de données cycliques en temps réel (RTC) et le tampon de données acycliques en temps réel (RTA) en fonction des différentes méthodes de traitement cyclique des données. Il existe plusieurs adresses de réception et de transmission pour chaque type de tampon de données. Les données de service relatives à l'AP dans le réseau de type 28 correspondent à la position correspondante dans le tampon de données en utilisant la modélisation de données.

La définition d'APO impliquée dans l'AP du protocole AL de type 28 adopte la méthode orientée objet. Toutes les classes d'APO héritent du nœud racine unique (Root) et la définition de tous les APO comprend quatre classes d'objets, Data Class (classe de données), Module Class (classe de module), Type Class (classe de type) et Method Class (classe de méthode) comme représenté à la Figure 5.



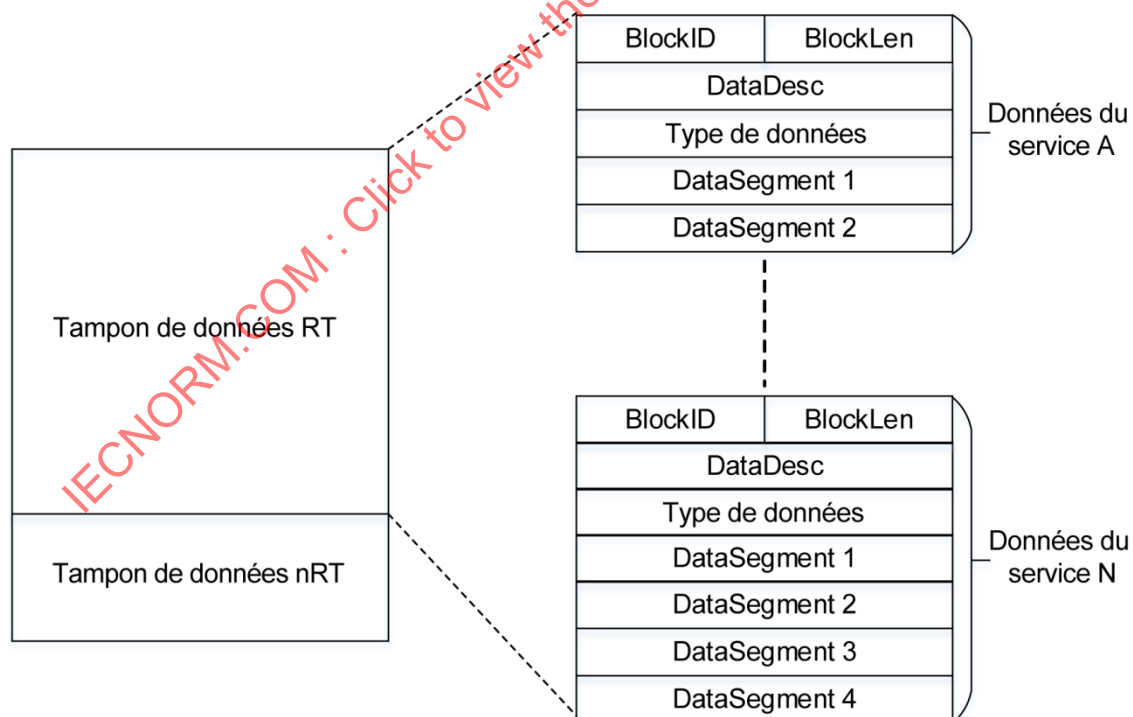
IEC

Figure 5 – Structure de l'objet de modélisation de correspondance de données

Dans l'environnement de réseau de type 28, les appareils ou modules décrivent une interaction AP spécifique à l'aide de données et de méthodes. Le contenu des données comprend les attributs, les fonctionnalités, les types et les données de l'appareil ou du module qui nécessitent d'être traités de manière interactive. Les données sont décrites en type de données simple ou en type de données complexe. Les méthodes peuvent être des méthodes privées, des méthodes publiques ou des méthodes partagées. Lorsque l'appareil ou le module effectue une interaction AP spécifique, il est nécessaire qu'il décrive avec les données et méthodes spécifiques, et il met en correspondance les données tampons avec le message de données réel à transmettre. Ces méthodes de modélisation et les méthodes de transmission de données sont directement liées au format des messages de données, ce qui peut améliorer l'efficacité et les performances de communication du bus (du point de vue de l'encapsulation et de l'analyse des messages de données). Voir l'Annexe A pour un exemple de mise en correspondance des données de service pour les messages DTU.

La méthode de modélisation de mise en correspondance des données fournie par le protocole AL de type 28 convient à différents environnements de bus industriels. Si le dictionnaire d'objets est utilisé pour définir les données d'AL, sur la base de la classe d'objets définie par le protocole d'AL de type 28, différentes données d'objet sontinstanciées séparément, puis mises en correspondance en des messages de données transmis par la communication pour satisfaire aux exigences d'interaction AP. La méthode du modèle de mise en correspondance est parfaitement compatible avec le modèle de données OPC UA. Voir l'Annexe B pour une description de modélisation de données à l'aide d'un capteur à titre d'exemple.

En modélisant les données d'APO dans l'environnement réseau de type 28, les variables de données de service réelles sont représentées par des blocs tampons dans le tampon de données, et chaque bloc a des informations déterministes telles que l'ID de bloc et la longueur de bloc comme représenté à la Figure 6.



IEC

**Figure 6 – Schéma des tampons de données de mise en correspondance et des variables de données de service**

**Description:**

- **BlockID:** L'adresse de début du bloc tampon de données correspondant aux données de service.
- **BlockLen:** Longueur de données valide du bloc tampon de données correspondant aux données de service, c'est-à-dire la longueur totale entre le début du segment de données et la fin du segment de données.
- **DataDesc:** Description de l'association avec le service d'application réel.
- **Type de données:** Détermine la taille du segment de données.
- **DataSegment:** Correspond à la variable de données de service réelle.

Dans le réseau de type 28, il convient que l'interaction AP accède aux données du bloc tampon par le DataID globalement unique. Il existe une table de correspondance de données de service dont la clé est le DataID unique qui existe sur l'appareil ou le module. Le contenu de chaque entrée unique est associé aux données réelles du bloc tampon de données. Les paramètres de la table sont indiqués dans le Tableau 11.

**Tableau 11 – Liste des paramètres de la table de correspondance des données de service**

| Nom de l'entrée | Type de données | Description de l'entrée                                                                                                   |
|-----------------|-----------------|---------------------------------------------------------------------------------------------------------------------------|
| DataID          | UINT16          | ID qui identifie de manière unique les données de service                                                                 |
| BuffType        | UINT8           | Type de tampon, 0x00: RT, 0x04: nRT, les autres valeurs sont réservées.                                                   |
| DataState       | UINT8           | Etat du bloc de données, 0x00: valide, 0x04: en cours d'utilisation, 0x0F: non valide, les autres valeurs sont réservées. |
| BlockID         | UINT32          | ID du bloc tampon de données                                                                                              |
| NetworkID       | UINT32          | NETWORKID de l'appareil ou du module auquel appartiennent les données de service                                          |

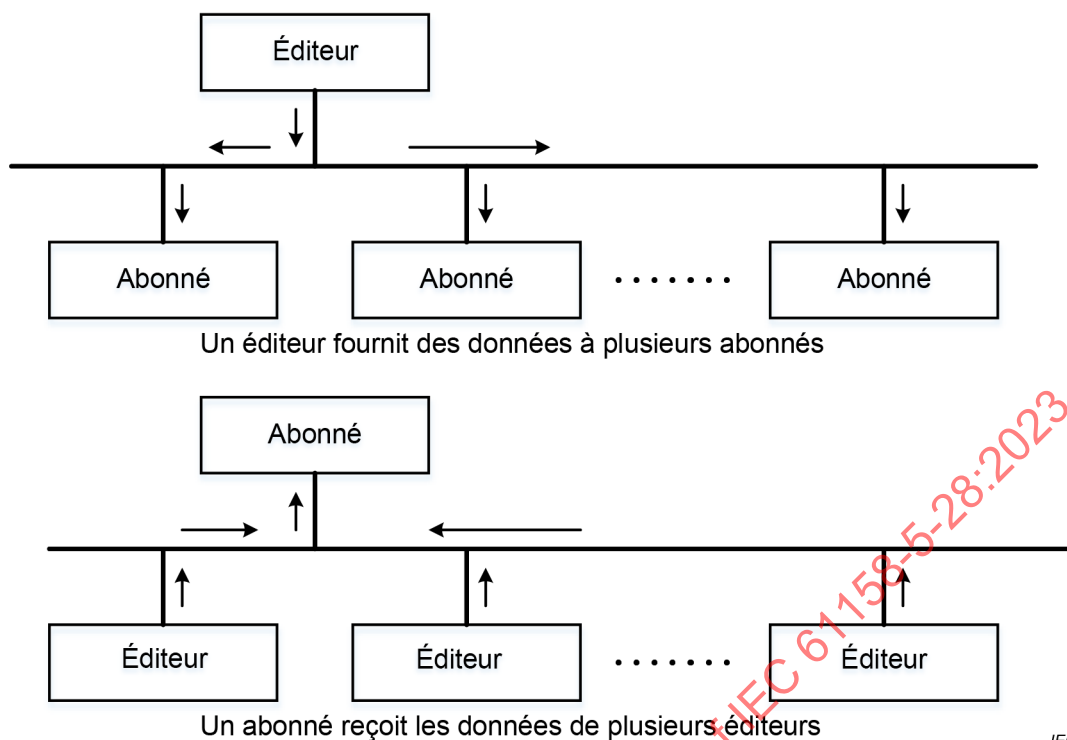
## 6.4 Vue d'ensemble du modèle de communication de type 28

### 6.4.1 Généralités

L'AP du réseau de type 28 met en œuvre les objectifs de la fonction de service déterminée par interaction. Les interactions entre les AP peuvent être basées sur le message de demande/réponse ou sur les données/événements émises par chacun des autres AP. Les modèles de communication de ces méthodes d'interaction sont le modèle éditeur/abonné (P/S) et le modèle client/serveur (C/S).

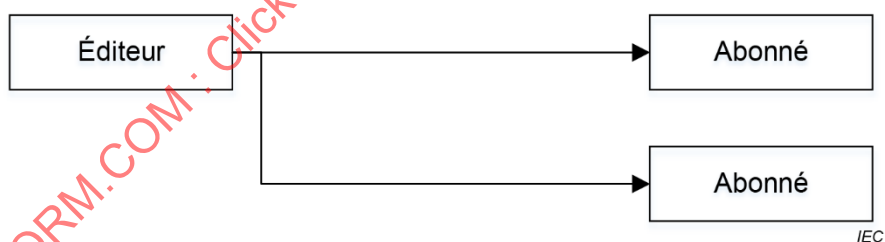
### 6.4.2 P/S

Dans le processus d'interaction AP basé sur le modèle P/S, un éditeur fournit des données à un ou plusieurs abonnés et un abonné reçoit des données fournies par un ou plusieurs éditeurs. Le schéma du modèle P/S est représenté à la Figure 7.

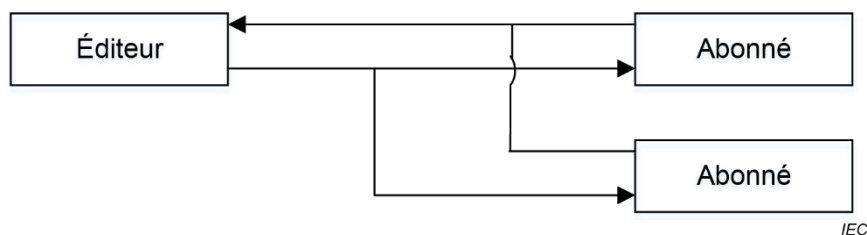


**Figure 7 – Schéma du modèle P/S**

Sur le réseau, l'appareil ou le module fournit des données de service en tant qu'éditeur ou reçoit des données de service en tant qu'abonné en fonction de la demande d'interaction AP. Il convient que l'interaction AP des données RT et des données nRT entre les AP adopte le modèle P/S. Dans le réseau de type 28, le modèle de communication P/S prend en charge à la fois le mode pousser (PUSH) et le mode extraire (PULL) comme représenté à la Figure 8 et à la Figure 9.



**Figure 8 – Modèle P/S du mode PUSH**



**Figure 9 – Modèle P/S du mode PULL**

Pour le mode PUSH, l'éditeur envoie activement des données à l'abonné. Ce mode de communication interactif n'a pas besoin de confirmer si l'abonné reçoit correctement les données. Pour le mode PULL, l'abonné demande activement à l'éditeur d'envoyer des données lorsque des données sont nécessaires, et le service d'application basé sur ce mode est généralement exigé pour être confirmé.

#### 6.4.3 C/S

L'interaction AP basée sur le modèle C/S utilise un flux de données bidirectionnel pour mettre en œuvre la fonction de service déterminée par demande et réponse. Sur la base de ce modèle, il existe deux modes: confirmé et non confirmé. Dans le mode confirmé, il est nécessaire que le serveur confirme l'état et l'achèvement du processus d'interaction conformément à la réponse du client au message de réponse. En mode non confirmé, le serveur confirme l'état et l'achèvement du processus d'interaction après avoir effectué un traitement de réponse par rapport à la demande du client. Le choix du mode confirmé et non confirmé par l'utilisateur dépend du service d'interaction d'application spécifique. Un schéma de modèle de communication C/S est représenté à la Figure 10.

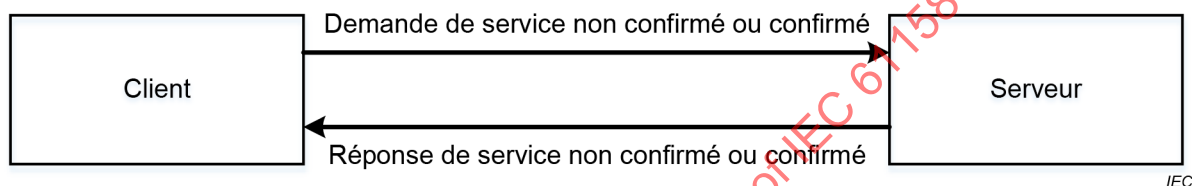


Figure 10 – Schéma du modèle de communication C/S

### 6.5 Table d'informations de gestion relatives à l'AL

#### 6.5.1 Généralités

Les informations de gestion d'AL sont compatibles avec les éléments correspondants du fichier de configuration de gestion système et existent sous la forme de variables logicielles dans l'appareil. Les informations de gestion d'AL incluent les informations de gestion de temps et les informations de gestion de réseau. Il convient que l'utilisateur de gestion système d'AL ou l'utilisateur d'AL accède à l'entrée d'information correspondante par l'intermédiaire de l'interface de service de gestion d'AL.

## 6.5.2 Table d'informations de gestion du temps

Tableau 12 – Table d'informations de gestion du temps

| Indice de paramètre | Nom de paramètre | RW  | Type de données | Description                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------|------------------|-----|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x2000              | time_fmt         | R/W | UINT8           | Différents formats de temps représentés par un bit<br>Bit 0: indique que le format de temps est basé sur l'heure, la minute et la seconde.<br>Bit 1: indique que le format de temps est basé sur les millisecondes et les microsecondes.<br>Bit 2: indique que le format de temps est basé sur l'impulsion de la seconde.<br>Le bit correspondant mis sur 0 est invalide, mis sur 1 est valide<br>Bit 3 au bit 7: réservés |
| 0x2001              | time_ns          | R/W | TIMEV           | Comptage du temps basé sur la nanoseconde                                                                                                                                                                                                                                                                                                                                                                                  |
| 0x2002              | timedate         | R/W | TIMEDATE        | Convertit en valeur de date de l'année, du mois, du jour, de la minute et de la seconde, dont la précision est de l'ordre des millisecondes                                                                                                                                                                                                                                                                                |
| 0x2003              | timeofday        | R/W | TIMEOFDAY       | Nombre de jours basés sur les ns                                                                                                                                                                                                                                                                                                                                                                                           |
| 0x2004              | timediffer       | R/W | TIMEOFDIFF      | La différence de temps basée sur la nanoseconde                                                                                                                                                                                                                                                                                                                                                                            |
| 0x2005              | tid_cnt          | R/W | UINT8           | Nombre d'horloges sur l'appareil ou le module<br>Plage de valeurs: 1 à 8<br>La valeur 0 n'est pas valide.<br>Les autres valeurs sont réservées.                                                                                                                                                                                                                                                                            |
| 0x2006              | tid_val          | R/W | UINT8           | ID d'horloge<br>Plage de valeurs: 0 à 7<br>Les autres valeurs sont réservées.                                                                                                                                                                                                                                                                                                                                              |
| 0x2007              | tid_attr         | R/W | UINT8           | Attribut de champ de commande de l'horloge<br>Plage de valeurs: 0x00 ou 0x01<br>0x00: horloge de commande<br>0x01: horloge de terrain<br>Les autres valeurs sont réservées.                                                                                                                                                                                                                                                |
| 0x2008              | tcd_lst          | R/W | ARRAY           | ARRAY de type UINT8<br>Instruction TCD pour l'horloge correspondante<br>La plage de valeurs est spécifiée dans l'IEC 61158-6-28, Article 4.                                                                                                                                                                                                                                                                                |
| 0x2009 à 0x208F     | réservé          |     |                 | Protocole réservé                                                                                                                                                                                                                                                                                                                                                                                                          |
| 0x2090 à 0x20FF     | réservé          |     |                 | Réserve pour le fabricant d'équipements                                                                                                                                                                                                                                                                                                                                                                                    |

## 6.5.3 Table d'informations de gestion de réseau

Tableau 13 – Table d'informations de gestion de réseau

| Indice de paramètre | Nom de paramètre               | RW  | Type de données | Description                                                                                                                                                                                                                                                                                                         |
|---------------------|--------------------------------|-----|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x2100              | network_id                     | R/W | UINT32          | Indiquer NETWORKID du nœud ou du module<br>Voir en 6.6.6.1.2.                                                                                                                                                                                                                                                       |
| 0x2101              | ip_switch                      | R/W | UINT8           | Prise en charge de l'IPv4 ou de l'IPv6 représentée par le bit<br>Bit 0: Prise en charge ou non de IPv4<br>Bit 1: Prise en charge ou non de IPv6<br>Le bit correspondant mis à 0 indique qu'il n'y a pas de prise en charge, et le bit correspondant mis à 1 indique la prise en charge.<br>Bit 2 au bit 7: réservés |
| 0x2102              | multicast_id0 à multicast_id15 | R/W | UINT32          | NETWORKID avec type de groupe indique le groupe de multidiffusion auquel appartient l'appareil ou le module.<br>Chaque appareil ou module appartient à un maximum de 16 groupes de multidiffusion.                                                                                                                  |
| 0x2103              | mac_addr                       | RW  | UINT64          | Adresse MAC unique de l'appareil ou du module<br>Plage de valeurs: 0 à $(2^{48} - 1)$<br>Les autres valeurs sont réservées.                                                                                                                                                                                         |
| 0x2104              | nodeid                         | RW  | UINT8           | Adresse ID de nœud de l'appareil<br>Plage de valeurs: 0 à 255<br>Voir l'IEC 61158-4-28, Article 4.                                                                                                                                                                                                                  |
| 0x2105              | dctime                         | RW  | UINT16          | Temps de cycle de calcul de données, l'unité est la $\mu$ s<br>Plage de valeurs: 0 à 65 535                                                                                                                                                                                                                         |
| 0x2106              | bctime                         | RW  | UINT16          | Temps de cycle de service du bus, l'unité est la $\mu$ s<br>Plage de valeurs: 0 à 65 535                                                                                                                                                                                                                            |
| 0x2107              | wctime                         | RW  | UINT16          | Temps de cycle d'écriture de données de service de bus, l'unité est la $\mu$ s<br>Plage de valeurs: 0 à 65 535                                                                                                                                                                                                      |
| 0x2108              | rcetime                        | RW  | UINT16          | Temps de cycle de lecture de données de service de bus, l'unité est la $\mu$ s<br>Plage de valeurs: 0 à 65 535                                                                                                                                                                                                      |
| 0x2109              | cctime                         | RW  | UINT16          | Temps de cycle de la concurrence du service de bus, l'unité est la $\mu$ s<br>Plage de valeurs: 0 à 65 535                                                                                                                                                                                                          |
| 0x210A              | totcnt                         | RW  | UINT16          | Nombre de temporisations autorisées par l'appareil ou le module<br>Plage de valeurs: 0 à 65 535                                                                                                                                                                                                                     |
| 0x210B              | rt_data_addr                   | RO  | UINT32          | Adresse de début de tampon de données en temps réel<br>Plage de valeurs: 0x00000000 à 0xFFFFFFFF                                                                                                                                                                                                                    |
| 0x210C              | rt_data_size                   | RW  | UINT32          | Taille du tampon de données en temps réel<br>Plage de valeurs: 0x00000000 à 0xFFFFFFFF                                                                                                                                                                                                                              |
| 0x210D              | nrt_data_addr                  | RW  | UINT32          | Adresse de début de tampon de données en temps différé<br>Plage de valeurs: 0x00000000 à 0xFFFFFFFF                                                                                                                                                                                                                 |

| Indice de paramètre | Nom de paramètre | RW | Type de données | Description                                                                               |
|---------------------|------------------|----|-----------------|-------------------------------------------------------------------------------------------|
| 0x210E              | nrt_data_size    | RW | UINT32          | Taille du tampon de données en temps différé<br>Plage de valeurs: 0x00000000 à 0xFFFFFFFF |
| 0x210F à<br>0x28FF  | réservé          |    |                 | Protocole réservé                                                                         |
| 0x2900 à<br>0x2FFF  | réservé          |    |                 | Réservé au fabricant d'équipements                                                        |

## 6.6 ASE

### 6.6.1 Vue d'ensemble

L'AL de type 28 fournit aux utilisateurs des services temporels, des services de données et des services de gestion système basés sur l'interface de service DLL. L'utilisateur d'AL adresse les appareils ou modules du réseau de type 28 et y accède par l'intermédiaire de NETWORKID. L'AP de temps fournit des fonctions de synchronisation temporelle et de service d'interrogation du temps par le biais de l'ASE de temps. L'AP de données fournit des opérations de lecture et d'écriture de données RT et de données nRT par le biais de l'ASE de données RT et de l'ASE de données nRT. L'AP de gestion système fournit les fonctions de service d'application du système telles que la gestion des ressources, la gestion des adresses, la gestion du système de réseau, la gestion de la virtualisation par le biais de l'ASE de ressources, l'ASE d'adressage, l'ASE de gestion et l'ASE de virtualisation. Ces entités d'application mettent en œuvre des services de transmission de données spécifiques au moyen de modèles de communication C/S ou P/S. Le Tableau 14 représente la description du modèle de communication utilisé par les ASE AL de type 28.

**Tableau 14 – Modèles de communication pris en charge par les ASE AL de type 28**

| ASE                        | C/S                    | P/S                    |
|----------------------------|------------------------|------------------------|
| Données RT cycliques       | Pas de prise en charge | Prise en charge        |
| Données RT acycliques      | Prise en charge        | Prise en charge        |
| Données nRT                | Prise en charge        | Prise en charge        |
| Synchronisation temporelle | Pas de prise en charge | Prise en charge        |
| Interrogation de temps     | Prise en charge        | Pas de prise en charge |
| Ressource                  | Prise en charge        | Prise en charge        |
| Adressage                  | Prise en charge        | Pas de prise en charge |
| Gestion                    | Prise en charge        | Prise en charge        |
| Virtualisation             | Prise en charge        | Pas de prise en charge |

La valeur de retour du service AL est indiquée dans le Tableau 15. L'utilisateur d'AL détermine si le service réussit et la raison de la défaillance en fonction de la valeur de retour.

**Tableau 15 – Liste des valeurs de retour de l'interface de service d'application**

| Nom incorrect      | Codage incorrect | Description incorrecte                                                         |
|--------------------|------------------|--------------------------------------------------------------------------------|
| SUCCESS            | 0x00000000       | Traitement de service réussi                                                   |
| FAILED             | 0x00000001       | Échec du traitement de service                                                 |
| ERR_NOTSUPPORT     | 0x00000002       | L'appareil ou le module ne prend pas en charge                                 |
| ERR_NOTINITIALIZED | 0x00000003       | L'appareil ou le module n'a pas été initialisé                                 |
| ERR_NOTEXIST       | 0x00000004       | L'appareil ou le module n'existe pas, le service n'est pas disponible          |
| ERR_NOTREADY       | 0x00000005       | L'initialisation de l'appareil se termine, mais aucun état valide n'est défini |
| ERR_INVALIDPARA    | 0x00000006       | Informations de paramètre non valides                                          |
| ERR_UNKNOWN        | 0x00000007       | Erreur inconnue                                                                |
| ERR_TIMEOUT        | 0x00000008       | Temporisation de traitement de service                                         |

## 6.6.2 ASE de données RT

### 6.6.2.1 Généralités

L'ASE de données RT définit les interfaces de service et les paramètres d'interface qui prennent en charge les services d'application RT. Le service d'application RT traite les données RT de manière cyclique ou acyclique. Les données RTC sont lues ou écrites de manière cyclique selon le temps de cycle spécifié. Le traitement des données RTA n'a pas de temps de cycle fixe, mais il convient qu'il soit achevé dans le délai spécifié lorsqu'il s'est produit une fois.

Les données RTC sont transmises en fonction du modèle de communication P/S et les données RTA sont transmises en fonction du modèle de communication P/S ou du modèle de communication C/S. Avant de traiter le service d'application RT, il est nécessaire que l'utilisateur préconfigure la ressource de canal en fonction de l'ASE de la ressource, et la ressource de canal sera occupée par l'utilisateur tout le temps. Si la ressource de canal pour la transmission de données RT est modifiée, il convient de redémarrer le système pour satisfaire à la dernière exigence du service de données RT.

### 6.6.2.2 Interface

ALS-RT-DATA-Proc (

Dst\_NetworkID,  
Src\_NetworkID,  
ProcessID,  
DataID,  
OptionRTLst,  
ALS\_Result

)

Le Tableau 16 définit les paramètres d'interface de service de données RT.

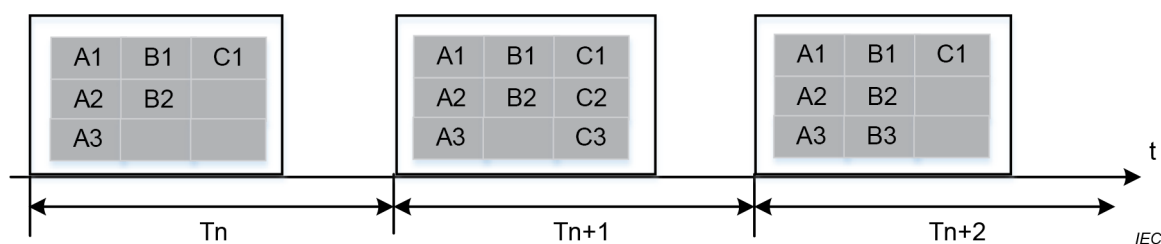
**Tableau 16 – Paramètres de l'interface de service de données RT**

| Nom de paramètre | Type de données | Entrée/sortie |
|------------------|-----------------|---------------|
| Dst_NetworkID    | UINT32          | Entrée        |
| Src_NetworkID    | UINT32          | Entrée        |
| ProcessID        | UINT16          | Entrée        |
| DataID           | UINT32          | E/S           |
| OptionRTLst      | STRUCT          | E/S           |
| ALS_Result       | UINT32          | Sortie        |

- Dst\_NetworkID: NETWORKID de l'appareil ou du module de destination.
- Src\_NetworkID: NETWORKID de l'appareil ou du module source.
- ProcessID: ID du mode de traitement des données RT, qui identifie la méthode de traitement de données RT basée sur des bits et si la position de bit correspondante est 1, exécute le traitement correspondant, défini comme suit:
  - mise à jour (bit 0): ): Mise à jour des données après comparaison du contenu des données en fonction de la valeur du bit;
  - réinitialisation (bit 1): Restaure les données dans le contenu de la zone de données à la valeur par défaut;
  - effacement (bit 2): Efface les données de la zone de données spécifiée et fixe l'opération nulle;
  - lecture (bit 3): Lit les données de la zone de données spécifiée et ne supprime pas le contenu de données correspondant;
  - lecture effacement (bit 4): Lit les données de la zone de données spécifiée et supprime les données de la zone de données correspondante;
  - écriture (bit 5): Écrit des données directement dans la zone de données spécifiée et écrase les données d'origine;
  - écriture effacement (bit 6): Efface l'opération pour la zone de données spécifiée, puis écrit de nouvelles données;
  - interrogation (bit 7): Interroge l'état des données;
  - demande (bit 8): Demande des données;
  - réponse (bit 9): Répond à la demande, données de réponse;
  - confirmation (bit 10): Confirme l'état de transmission de données;
  - envoi (bit 11): Envoie des données à l'appareil ou au module de destination spécifié;
  - réception (bit 12): Reçoit les données envoyées par un appareil ou un module source spécifié;
- DataID: ID d'encodage de données de service, ID unique des données de service spécifiques. L'ID peut être associé à la zone de données correspondant au tampon de données. Pour les données RTC et les données RTA, le DataID correspondant est différent.
- OptionRTLst: Une liste d'options d'informations de traitement de données RT, y compris l'état des informations de données après traitement, comme décrit ci-dessous:
  - StateInfo: Valeur de type UINT16, informations d'état de la zone de données;
  - DataSize: Valeur de type UINT16, taille des données valides exigée pour la lecture et l'écriture;
  - RetCode: Valeur de type UINT32, encodage du résultat du traitement des informations des données RT, la valeur par défaut est 0.
- ALS\_Result: La valeur que le service de données RT retourne, comme spécifié dans le Tableau 15.

### 6.6.2.3 Processus d'interaction

Le traitement des données RT est déterministe et opportun. Le caractère déterministe signifie qu'il convient de traiter les données à un certain moment, tandis que le caractère opportun signifie que le temps exigé pour que les données soient transmises de l'appareil ou du module source à l'appareil ou au module de destination est attendu. Il est nécessaire de traiter les données RTC de manière cyclique dans un cycle fixe, alors que les données RTA n'ont pas besoin d'être traitées de manière cyclique dans le cycle fixe.



**Figure 11 – Modèle de transmission de données RT**

Comme représenté à la Figure 11, les blocs de données de service A1, A2, A3, B1, B2 et C1 sont toujours traités de manière cyclique dans chaque période fixe, comme  $T_n$ ,  $T_{n+1}$ ,  $T_{n+2}$ , ce sont des interactions de données RTC. Les blocs de données de service B3, C2 et C3 ne sont pas traités dans chaque cycle. Lorsque la transmission est exigée, le traitement RT est achevé dans une certaine période, c'est-à-dire les interactions de données RTC.

Étant donné que les données RTA sont généralement des informations de commande, de configuration, d'alarme ou d'exception, elles ne sont pas traitées de manière cyclique. Voir l'Annexe C pour un exemple de procédure de traitement de service RTA. Lorsque le service d'application de données RTA adopte le modèle de communication C/S, l'application client envoie une demande au serveur. L'application serveur acquiert la demande de traitement correspondant et envoie les données de réponse à l'application cliente conformément au contenu demandé dans un délai spécifique. L'application cliente obtient le contenu des données de réponse envoyées par le serveur et confirme l'achèvement du service interactif.

### 6.6.3 ASE de données nRT

#### 6.6.3.1 Généralités

Dans le réseau de type 28, pour les données de service d'application nRT, le temps critique n'est pas nécessaire et la longueur du message n'est pas fixe. Avant d'effectuer la transmission de données nRT, l'ASE de la ressource est utilisé pour appliquer la ressource de canal pour la transmission au MN. Il convient de spécifier les durées maximales de ces ressources de canal appliquées à l'utilisateur.

L'ASE de données nRT définit les interfaces et les paramètres associés pour le traitement des données nRT, et fournit également aux utilisateurs des opérations telles que la lecture et l'écriture des données nRT. L'utilisateur utilise l'ASE de ressource pour demander des ressources de canal pour le traitement des données nRT, et il est nécessaire que l'utilisateur spécifie les informations de ressource de canal dans le traitement des données nRT.

Lorsque l'AP de données nRT interagit sur la base du modèle de communication C/S, l'application client envoie une demande de lecture ou d'écriture à l'application serveur, et l'application serveur effectue un traitement de réponse. Lorsque l'AP de données nRT interagit sur la base du modèle de communication P/S, les données sont transmises par l'intermédiaire des ressources de canal par défaut au repos attribuées par le système. Le système planifie les ressources de canal en fonction de la priorité de service d'application nRT préaffectée, et attribue de préférence les ressources de canal inactif aux utilisateurs de service d'application nRT avec une priorité élevée.

### 6.6.3.2 Interface

```
ALS-NRT-DATA-Proc (
    Dst_NetworkID,
    Src_NetworkID,
    ProcessID,
    DataID,
    OptionNRTLst,
    ALS_Result
)
```

Le Tableau 17 définit les paramètres de l'interface de service de données nRT.

**Tableau 17 – Paramètres de l'interface de service de données nRT**

| Nom de paramètre | Type de données | Entrée/sortie |
|------------------|-----------------|---------------|
| Dst_NetworkID    | UINT32          | Entrée        |
| Src_NetworkID    | UINT32          | Entrée        |
| ProcessID        | UINT16          | Entrée        |
| DataID           | UINT32          | Entrée        |
| OptionNRTLst     | STRUCT          | E/S           |
| ALS_Result       | UINT32          | Sortie        |

- Dst\_NetworkID: NETWORKID de l'appareil ou du module de destination.
- Src\_NetworkID: NETWORKID de l'appareil ou du module source.
- ProcessID: ID de mode de traitement des données nRT. Le mode de traitement des données est représenté par un bit, le bit 15 position 1 indiquant que les données sont transmises en fonction de la pile de protocole TCP/IP et que les informations IP sont obtenues par le NETWORKID. Le bit 15 position 0 indique que le traitement est basé sur la couche DLL de type 28, et voir la description du ProcessID décrite en 6.6.2.2 pour la description du bit restant.
- DataID: ID de données de service, qui est associé à l'adresse correspondante du tampon de données. Pour les données nRT correspondant à l'interface de service, il est possible de choisir de ne pas les associer au tampon. Dans ce cas, l'ID est invalide.
- OptionNRTLst: Une liste de paramètres facultatifs, comprenant les informations suivantes:
  - Opt\_Cmd: Valeur de type UINT8, mot de commande facultatif, qui identifie si le service a besoin de confirmation; 0x00: le service n'a pas besoin d'être confirmé; 0x01: il est nécessaire de confirmer le service;
  - DataSize: Valeur de type UINT16, la taille effective des données qui nécessitent d'être lues et écrites;
  - DataAddr: Adresse de la zone de stockage des données de type UINT32, cette valeur est valide lorsque le DataID est invalide;
  - Data\_Flag: Valeur de type UINT8, qui identifie l'opération de libération après lecture et écriture du bloc de données nRT; 0x00: pas de traitement; 0x01: libération claire; 0x02: pas de libération claire.
- ALS\_Result: Le service de données nRT traite la valeur de retour. Voir le Tableau 15 pour les valeurs de retour spécifiques.

### 6.6.3.3 Processus d'interaction

Dans le réseau de type 28, lorsque l'interaction AP de données nRT utilise le modèle de communication C/S, l'interface de service de données nRT fournit un mode confirmé et un mode non confirmé pour satisfaire aux exigences de différents services. Dans le mode confirmé, le client envoie un message confirmé au serveur après réception et traitement corrects du message de réponse envoyé par le serveur. Dans le mode non confirmé, il convient que le client n'envoie pas le message confirmé au serveur après réception et traitement corrects du message de réponse envoyé par le serveur. Dans le processus de mise en œuvre du service d'application de données nRT en utilisant le modèle de communication C/S, après que l'application client a initié la demande, elle attend la réponse asynchrone de l'application serveur, et la durée d'intervalle asynchrone est généralement un ou plusieurs cycles de bus comme représenté à la Figure 12. La durée de l'intervalle est liée au cycle de bus configuré et à la taille spécifique des données nRT qu'il est nécessaire de transmettre.

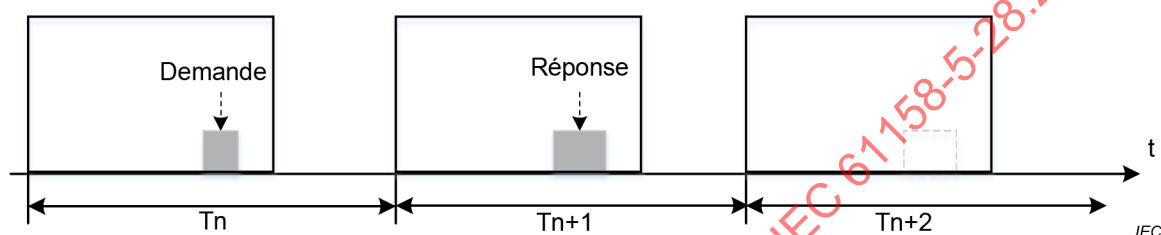


Figure 12 – Schéma du modèle de réponse de demande de données nRT

Lorsque l'interaction AP de données nRT adopte le modèle de communication C/S, l'application client initie une demande de service de données nRT. L'application serveur répond à la demande d'application cliente, complète le traitement des données localement ou envoie les données de réponse à l'application client distante. En fonction du contenu spécifique différent de la demande d'application cliente, le service de transmission de données basé sur la demande et la réponse est continuellement mené à bien entre l'application serveur et l'application cliente.

La Figure 13 représente l'interaction AP de données nRT basée sur le modèle de communication C/S.

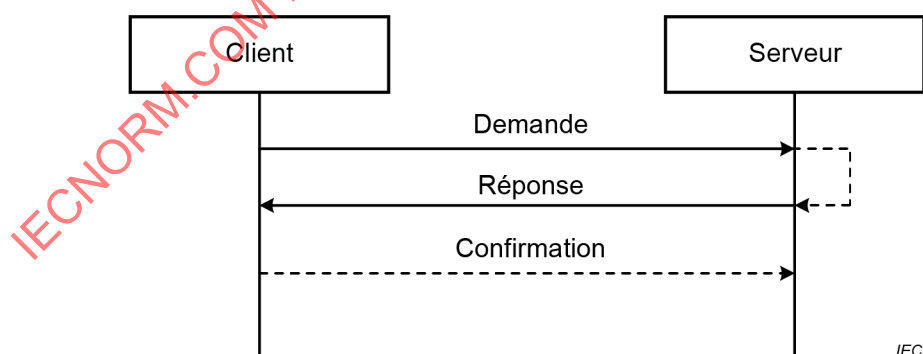


Figure 13 – Schéma d'interaction AP de données nRT basé sur le modèle de communication C/S

Lorsque l'interaction AP de données nRT adopte le modèle P/S, elle planifie les ressources de canal en fonction de la priorité du service d'application de données nRT et envoie des données nRT par le biais du canal par défaut inactif. Selon les caractéristiques des données nRT, les utilisateurs du service d'application de données nRT utiliseront différents canaux par défaut et différentes périodes d'occupation. Un schéma d'interaction AP de données nRT basé sur le modèle de communication P/S est présenté à la Figure 14, où le service nRT 1 et le service nRT 2 peuvent tous deux utiliser un canal par défaut pour la transmission de données, et le service nRT 3 a une priorité plus élevée et a besoin d'utiliser le canal par défaut deux fois pour mener à bien la transmission de données correspondante.

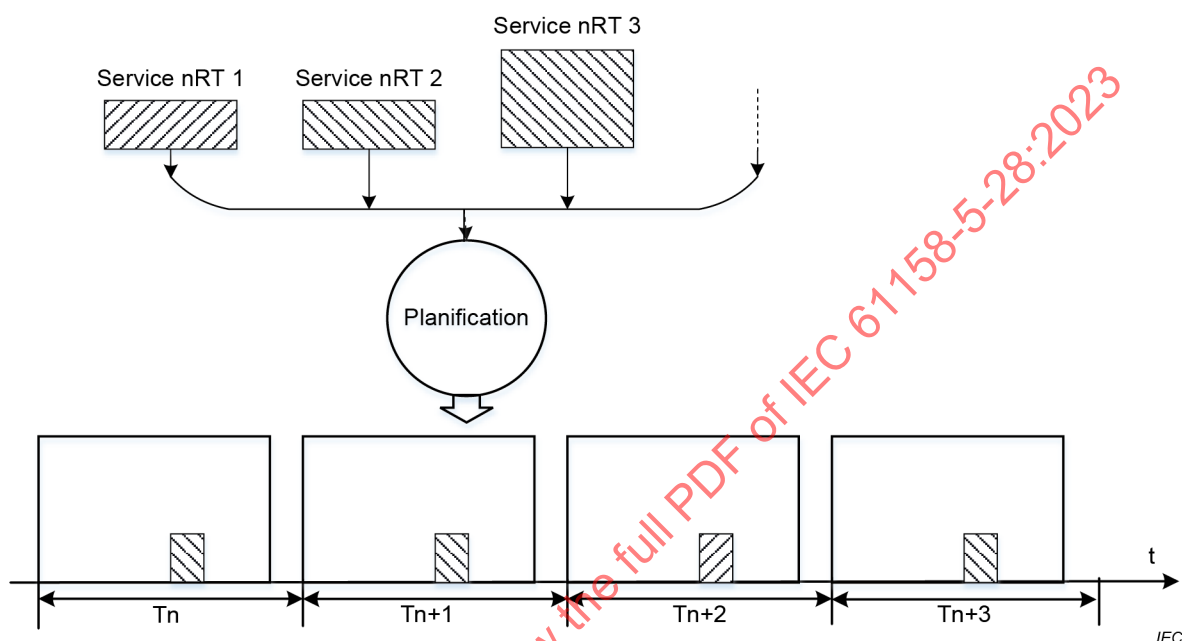


Figure 14 – Schéma d'interaction AP de données nRT basé sur le modèle de communication P/S

## 6.6.4 ASE de temps

### 6.6.4.1 Généralités

L'ASE de temps définit l'interface et les paramètres du service de temps. Les appareils et les modules dans le réseau de type 28 mettent en œuvre les fonctions de synchronisation temporelle et de service d'interrogation du temps sur la base de l'ASE de temps.

Sur le réseau, l'utilisateur d'AL déclenche le service de synchronisation temporelle au niveau du MN. Le MN envoie un message comprenant les informations relatives au temps de propagation entre le TN et tous les TN, puis le TN analyse les informations relatives au temps de retard à partir du message envoyé par le MN, et règle l'heure de la station pour terminer la synchronisation.

Le service d'interrogation de temps fournit une interrogation pour l'état de synchronisation temporelle du temps du TN. L'objet du service d'interrogation sur MN est n'importe quel nœud du réseau, à l'exception du nœud à proprement parler sur TN.

### 6.6.4.2 Interface

ALS-TIMING-Proc (

Dst\_NetworkID,  
Src\_NetworkID,  
ProcessID,  
OptionClkInfo,  
TimeInfo,  
ALS\_Result

)

Le Tableau 18 définit les paramètres de l'interface de service de cadencement.

**Tableau 18 – Paramètres de l'interface du service de cadencement**

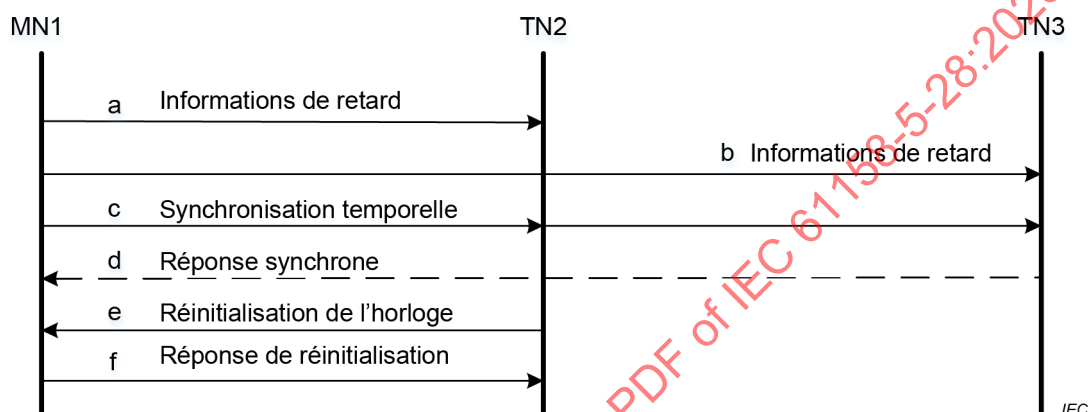
| Nom de paramètre | Type de données | Entrée/sortie |
|------------------|-----------------|---------------|
| Dst_NetworkID    | UINT32          | Entrée        |
| Src_NetworkID    | UINT32          | Entrée        |
| ProcessID        | UINT16          | Entrée        |
| OptionClkInfo    | STRUCT          | E/S           |
| TimeInfo         | STRUCT          | E/S           |
| ALS_Result       | UINT32          | Sortie        |

- Dst\_NetworkID: NETWORKID de l'appareil ou du module de destination.
- Src\_NetworkID: NETWORKID de l'appareil ou du module source.
- ProcessID: ID de traitement de service de temps:
  - 0x0000: Demande de synchronisation temporelle;
  - 0x0001: Réponse de synchronisation temporelle;
  - 0x0002: Exception de synchronisation temporelle;
  - 0x0004: Demande d'interrogation de temps;
  - 0x0008: Réponse d'interrogation de temps;
  - 0x0010: Erreur d'interrogation de temps.
- OptionClkInfo: Informations d'option de temps. Lors de la mise en œuvre du service de synchronisation temporelle, ce paramètre est utilisé comme paramètre d'entrée et spécifie la liste d'informations d'appareil d'horloge qu'il est nécessaire de définir. Lorsque ce paramètre est utilisé comme paramètre de sortie, il doit spécifier les informations de sortie définies. Si le réglage réussit, il émet l'état actuel, la valeur actuelle et l'attribut actuel du temps de sortie; si le réglage échoue, le code d'état d'erreur correspondant est émis. Voir le Tableau 7 pour la description de la structure CLOCK\_OPTION\_INFO\_S. Lorsque la fonction de service d'interrogation de temps est mise en œuvre, ce paramètre est un paramètre de sortie.
- TimeInfo: Informations de temps qu'il est nécessaire de définir pour le temps. Voir le Tableau 6 pour la description de la structure TIMEINFO\_S.
- ALS\_Result: Valeur de retour de traitement de service de temps. Voir le Tableau 15 pour plus de détails sur la valeur de retour.

### 6.6.4.3 Processus d'interaction

#### 6.6.4.3.1 Processus d'interaction de synchronisation temporelle

Le MN envoie ses propres informations de temps et le retard de transmission mesuré entre les nœuds correspondants à chaque TN, et le TN synchronise le temps et met à jour les informations de temps locales en fonction des informations de temps du MN. Le processus d'interaction de synchronisation temporelle adopte le modèle de communication P/S et l'application temporelle du MN fait office d'éditeur et envoie périodiquement des informations de temps à l'application temporelle du TN, mais la période de libération est définie par le MN. Si une défaillance ou une erreur de synchronisation survient dans l'application de temps client, une opération de réinitialisation du temps peut être demandée à l'application de temps du MN.



**Figure 15 – Processus d'interaction d'application de synchronisation temporelle**

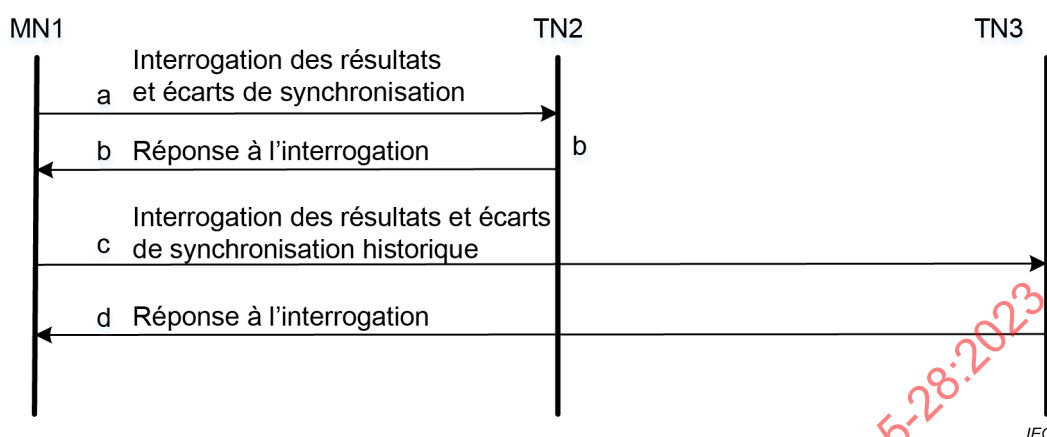
Voir la Figure 15. La synchronisation temporelle est initiée par le MN:

- le MN envoie la valeur de retard de transmission du MN correspondant au TN;
- le MN initie la synchronisation temporelle, envoie la valeur de temps au TN par l'intermédiaire du message de notification de service de temps, et demande au TN de répondre au résultat de synchronisation;
- après réception des informations de temps dans le message de notification de service de temps envoyé par le MN, le TN met en œuvre la synchronisation temporelle locale par l'intermédiaire de l'interface DLL locale et détermine s'il convient de répondre au résultat de synchronisation conformément aux exigences du MN;
- si une erreur se produit sur le TN pendant la synchronisation temporelle, le message d'erreur de réponse doit être transmis au MN;
- si une erreur se produit sur le TN au cours de la synchronisation temporelle, le TN peut envoyer activement une demande de réinitialisation du temps au MN et demander au MN de mettre à jour les informations de temps et les informations de retard de transmission de ligne. Dans le même temps, il convient que le MN mette à jour les informations de temps et les informations de retard de transmission de ligne au TN désigné;
- le MN peut choisir une synchronisation temporelle périodique ou une opération de synchronisation temporelle unique conformément aux exigences du système et à la configuration.

#### 6.6.4.3.2 Processus d'interaction d'interrogation de temps

Le service d'interrogation de temps fournit la fonction d'interroger l'état de synchronisation temporelle et la qualité de synchronisation du TN. Le service d'interrogation de temps utilise le modèle de communication C/S pour interagir. L'application de temps du MN fait office de client et demande l'état de synchronisation temporelle et des informations de qualité de synchronisation à partir de l'application de temps au TN en tant que serveur. Le MN envoie un

message de notification de service de temps pour déclencher le service d'interrogation de temps. Pour la procédure, voir la Figure 16.



**Figure 16 – Processus d'interrogation de temps**

Voir la Figure 16. L'interrogation de temps initiée par le MN:

- le MN interroge le moment de la dernière synchronisation du TN correspondant et la valeur d'écart pendant la synchronisation;
- après réception de l'interrogation, le TN répond aux informations d'enregistrement local du MN, et le MN détermine la qualité et la plage d'écarts de la synchronisation du TN sur la base de celles-ci;
- le MN peut interroger d'autres informations de terminal du TN correspondant, comme la valeur moyenne, la valeur efficace, l'intervalle de temps, etc., des valeurs d'écart de synchronisation;
- le TN répond au contenu correspondant du MN en fonction du contenu de l'interrogation.

## 6.6.5 ASE de ressource

### 6.6.5.1 Généralités

L'ASE de ressource du protocole AL de type 28 définit la définition de l'interface de service et les paramètres des ressources de canal exigés pour l'interaction AP sur le réseau. L'ASE de la ressource effectue l'attribution et l'utilisation des ressources de canal en se basant sur l'interface de service fournie par la DLL de type 28, et met ainsi en œuvre les opérations d'application, d'établissement, de mise à jour et de libération des ressources de canal.

Les ressources du canal comprennent les ressources de canal à largeur de bande fixe et les ressources de canal à largeur de bande variable. Les ressources de canal à largeur de bande fixe sont créées pendant la configuration de type 28 ou l'initialisation du réseau, et ne peuvent pas être mises à jour et publiées par la suite; les ressources de canal à largeur de bande variable sont créées et appliquées par les utilisateurs, si nécessaire. La taille des ressources et le temps d'utilisation des ressources du canal à largeur de bande variable sont déterminés par l'utilisateur au moment de l'application, et les utilisateurs peuvent mettre à jour et libérer dynamiquement le canal de ressource pendant l'utilisation. Le MN est chargé de l'attribution et de la gestion de toutes les ressources de canal du réseau de type 28.

L'ASE de la ressource met en œuvre l'interaction AP en utilisant le modèle de communication C/S confirmé. Lorsque la ressource AP interagit, l'application serveur est toujours au niveau du MN, tandis que l'application cliente peut être au niveau du MN ou du TN. L'application client envoie une demande d'application de ressource et l'application serveur répond à la demande d'application de ressource, envoie les données de réponse, attribue le canal de ressource ou rejette la demande d'application de ressource, tandis que l'application client