

PUBLICLY AVAILABLE SPECIFICATION

PRE-STANDARD

Industrial communication networks – Broadband fieldbus specification – Autbus

IECNORM.COM : Click to view the full PDF of IEC PAS 63256:2020



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2020 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.

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

Tel.: +41 22 919 02 11
info@iec.ch
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

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 It also gives information on projects, replaced and withdrawn publications.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

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

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

Electropedia - www.electropedia.org

The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

67 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IECNORM.COM : Click to view the full text of IEC 60326:2020

PUBLICLY AVAILABLE SPECIFICATION

PRE-STANDARD

Industrial communication networks – Broadband fieldbus specification – Autbus

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 00.000

ISBN 978-2-8322-7749-2

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

FOREWORD.....	11
1 Scope.....	12
2 Normative references	12
3 Terms and definitions, abbreviated terms and symbols	12
3.1 Terms and definitions.....	12
3.2 Abbreviated terms.....	14
3.3 Symbols.....	16
4 Data type.....	16
4.1 Overview.....	16
4.2 Basic data type	16
4.2.1 Unsigned type coding (UINT).....	16
4.2.2 Time type coding (TIMEV)	17
4.2.3 Date type coding (DATE2000)	17
4.2.4 Time of day type coding (TIMEOFDAY)	18
4.2.5 Time difference type coding (TIMEDIFFER).....	18
4.2.6 String type coding (STRING)	18
4.2.7 Bitmap type coding (BITMAP).....	19
4.3 Complex type.....	19
4.3.1 Struct type coding (STRUCT).....	19
4.3.2 Array type coding (ARRAY)	19
5 AUTOBUS overview	19
5.1 Overview.....	19
5.2 AUTOBUS protocol stack architecture	20
5.3 Device type.....	22
5.3.1 Overview	22
5.3.2 Control Device.....	22
5.3.3 Field Device	22
5.3.4 Management Node	22
5.3.5 Terminal Node.....	22
5.4 Network topology	22
6 System management.....	23
6.1 Overview.....	23
6.2 System management process	23
6.3 System management information table.....	24
6.3.1 PhL related management information table.....	24
6.3.2 DLL related management information table.....	31
6.3.3 AL related management information table.....	34
7 Physical layer	36
7.1 Physical Layer overview.....	36
7.2 PhL resource	37
7.2.1 Signal frame	37
7.2.2 Data subframe	39
7.2.3 OFDM symbol.....	40
7.2.4 Code Block.....	41
7.2.5 Resource element.....	41
7.2.6 Working mode	42

7.3	DLL–PhL interface	44
7.3.1	General	44
7.3.2	Service primitive	45
7.3.3	Primitive process	47
7.4	System management-PhL interface.....	47
7.4.1	General	47
7.4.2	Service primitive	47
7.4.3	Primitive process	49
7.5	DCE Independent Sublayer (DIS).....	50
7.5.1	General	50
7.5.2	Data sequence	50
7.6	DTE-DCE interface	51
7.6.1	General	51
7.6.2	Interface signal.....	51
7.6.3	Signal process	52
7.7	Medium Dependent Sublayer (MDS)	52
7.7.1	General	52
7.7.2	MDS specification	52
7.8	MDS-MAU interface	59
7.8.1	General	59
7.8.2	MDS-MAU interface	59
7.9	Medium Attachment Unit (MAU)	62
7.9.1	Overview	62
7.9.2	Electrical specifications	62
7.9.3	Transmission Medium Interface	62
7.9.4	Medium	63
8	Data-link layer	65
8.1	General.....	65
8.1.1	DLL protocol architecture.....	65
8.1.2	DLL working mechanism.....	67
8.2	DLL services.....	70
8.2.1	General	70
8.2.2	Data-link service model	71
8.2.3	Data-link service description	72
8.2.4	Data-link management service description.....	85
8.2.5	Clock synchronization service description.....	101
8.3	DLL protocol	107
8.3.1	Working procedure	107
8.3.2	DLPDU structure	116
8.3.3	State machine	129
8.3.4	Error handling.....	134
9	Application Layer	136
9.1	Overview.....	136
9.1.1	Protocol architecture.....	136
9.1.2	Data type of AL.....	137
9.1.3	Data mapping modeling	140
9.1.4	Communication model	143
9.2	AL Service	144
9.2.1	Overview	144

9.2.2	Application Service Elements.....	145
9.2.3	Application service	169
9.3	Specification for AL protocol	171
9.3.1	Overview	171
9.3.2	Message format.....	171
9.3.3	State machine	176
Annex A (informative)	Example of data subframe allocation	184
A.1	Example A	184
A.2	Example B	185
A.3	Example C	186
Annex B (normative)	RS code generating polynomial	187
Annex C (informative)	Example for NodeID and MAC address mapping	188
Annex D (informative)	Example for multicast group working mechanism	189
Annex E (informative)	Example of service data mapping DTU message	190
Annex F (informative)	Example of OPC UA data model and AUTOBUS data model mapping.....	191
Annex G (informative)	Example of RTA service processing	192
Annex H (informative)	AUTOBUS virtualization solution example.....	193
Bibliography		196
Figure 1 – BITMAP data type		19
Figure 2 – Corresponding relationship between AUTOBUS protocol stack and OSI reference model.....		20
Figure 3 – AUTOBUS protocol stack architecture		21
Figure 4 – AUTOBUS network topology		23
Figure 5 – AUTOBUS system management process		24
Figure 6 – Resource block information structure		31
Figure 7 – AUTOBUS Physical Layer model		37
Figure 8 – AUTOBUS SF and OFDM symbol		38
Figure 9 – SF structure		39
Figure 10 – Carrier mode A and carrier mode B of data subframe		40
Figure 11 – OFDM symbol structure of PhL.....		40
Figure 12 – OFDM timing structure		41
Figure 13 – DLPDU and CB		41
Figure 14 – Resource element schematic diagram		42
Figure 15 – Mapping between data units across the DLL-PhL interface.....		44
Figure 16 – DLL-PhL interface service primitive process		47
Figure 17 – System management-PhL interface primitive process		50
Figure 18 – DTE-DCE interface signal process		52
Figure 19 – PhL channel process		53
Figure 20 – Scrambling sequence generation		53
Figure 21 – Convolutional encoder with a code rate of 1/2		55
Figure 22 – Bit deletion process with code rates 2/3 and 3/4.....		55
Figure 23 – Generation of m sequence		57
Figure 24 – OFDM symbol structure diagram		58

Figure 25 – MDS-MAU interface service process diagram	60
Figure 26 – Signal spectrum template	61
Figure 27 – Connector of the shielded twisted pair	64
Figure 28 – Terminal resistor	64
Figure 29 – DLL in AUTBUS protocol stack architecture	65
Figure 30 – Relationship of AUTBUS DLL to other filedbus layers and to users of AUTBUS DLS	66
Figure 31 – AUTBUS DLL protocol architecture diagram	66
Figure 32 – Resource mapping between DLL and PhL	69
Figure 33 – DLL resource mapping message queue scheduling diagram	70
Figure 34 – Primitive process of CLMDTA	73
Figure 35 – Primitive process of CLMDTNA	75
Figure 36 – Primitive process of CLMDTRA	77
Figure 37 – Primitive process of CLMDTRRNA	79
Figure 38 – Primitive process of CMDTA	81
Figure 39 – Primitive process of CMDTNA	83
Figure 40 – Data-link config management service primitive process	86
Figure 41 – TN active logout discovery service primitive process	90
Figure 42 – Data-link discovery management service primitive sequence diagram	90
Figure 43 – Local data-link maintenance service primitive process diagram	93
Figure 44 – Remote data-link maintenance service primitive process diagram	93
Figure 45 – Data-link establish management service primitive process	96
Figure 46 – Data-link release management service primitive sequence diagram	98
Figure 47 – Data-link update management service primitive sequence diagram	100
Figure 48 – Delay measurement service primitive process	102
Figure 49 – Clock synchronization service primitive process	104
Figure 50 – Clock interrupt service primitive process	106
Figure 51 – Resource mapping configuration diagram	108
Figure 52 – Initial access configuration procedure diagram	109
Figure 53 – The random access configuration procedure diagram	110
Figure 54 – Node leave procedure diagram	112
Figure 55 – DLS data sending procedure diagram	113
Figure 56 – DLS data receiving procedure diagram	114
Figure 57 – Clock synchronization delay measurement procedure diagram	115
Figure 58 – Clock register structure diagram	115
Figure 59 – Clock synchronization procedure	116
Figure 60 – Universal DLPDU structure	116
Figure 61 – Basic configuration DLPDU structure	117
Figure 62 – General configuration block structure	118
Figure 63 – Address assignment DLPDU structure	119
Figure 64 – Multicast assignment DLPDU structure	120
Figure 65 – Resource allocation DLPDU structure	121
Figure 66 – Access notification DLPDU structure	122

Figure 67 – Resource application DLPDU structure.....	123
Figure 68 – Resource release DLPDU structure.....	124
Figure 69 – Status query DLPDU structure	125
Figure 70 – Status response DLPDU.....	125
Figure 71 – Announcement DLPDU structure.....	126
Figure 72 – Clock synchronization DLPDU structure	128
Figure 73 – Common DLPDU structure	129
Figure 74 – DLDE state machine	130
Figure 75 – DLME state machine	132
Figure 76 – DLCE state machine	133
Figure 77 – AL in AUTBUS protocol stack architecture.....	136
Figure 78 – Relationship of AUTBUS AL to other fieldbus layers and to users of AUTBUS application service	136
Figure 79 – Diagram of AUTBUS AL protocol architecture	137
Figure 80 – Data mapping modeling object structure.....	141
Figure 81 – Diagram of mapping data buffers and service data variables	142
Figure 82 – Diagram of P/S model	143
Figure 83 – P/S model of PUSH mode	143
Figure 84 – P/S model of PULL mode	144
Figure 85 – Diagram of C/S communication model.....	144
Figure 86 – RT data transmission model.....	147
Figure 87 – Diagram of nRT data request response model.....	149
Figure 88 – Diagram of nRT data AP interaction based on C/S communication model.....	149
Figure 89 – Diagram of nRT data AP interaction based on P/S communication model.....	150
Figure 90 – Time synchronization application interaction pocess	152
Figure 91 – Time query fow	153
Figure 92 – Resource AP interaction based on C/S communication model	155
Figure 93 – Resource ASE local service function Diagram	155
Figure 94 – NETWORKID identification field structure.....	156
Figure 95 – NETWORKID interaction process diagram.....	158
Figure 96 – Diagram of MAC mapping table service function.....	158
Figure 97 – IP mapping table service function diagram	159
Figure 98 – AUTBUS AL data message is mapped to IP payload data payload	159
Figure 99 – AUTBUS AL protocol header and IP protocol header mapping.....	160
Figure 100 – IP data message is mapped to valid data of AUTBUS AL data message.....	160
Figure 101 – Static configuration initialization process	163
Figure 102 – Join the network interaction process.....	163
Figure 103 – Passively exit the network interaction process.....	165
Figure 104 – Actively exit network interaction process	165
Figure 105 – Diagram of diagnostic object mapping	166
Figure 106 – Diagram of the logging process	167
Figure 107 – A physical bus divided into two virtual bus domain diagrams	167
Figure 108 – Diagram of virtualized AP interaction based on C/S communication model	169

Figure 109 – Virtualization ASE local implementation diagram	169
Figure 110 – AL protocol message header structure	171
Figure 111 – Message structure of multiple DTUs	173
Figure 112 – Transmission unit message format	174
Figure 113 – Time service notification message structure	174
Figure 114 – System management data message format	176
Figure 115 – Diagram of state transition of publisher state machine	177
Figure 116 – Diagram of state transition of subscriber state machine	179
Figure 117 – Diagram of state transition of client state machine	180
Figure 118 – Diagram of transition state of server state machine	180
Figure 119 – Diagram of transition state of ALCE state machine	181
Figure 120 – Diagram of transition state of ALDE state machine	182
Figure 121 – Diagram of state transition for ALME state machine	183
Figure A.1 – Resource allocation example A	184
Figure A.2 – Resource allocation example B	185
Figure A.3 – Resource allocation example C	186
Figure D.1 – Multicast working mechanism diagram	189
Figure E.1 – Data transmission unit message content example	190
Figure F.1 – OPC UA data model and mapping of AUTBUS data model	191
Figure G.1 – Example of RTA service processing	192
Figure H.1 – Diagram of AUTBUS industrial bus service deployment	193
Figure H.2 – Diagram of networking topology based on the virtualized bus	193
Figure H.3 – Virtual network topology based on logical business function RT1	194
Table 1 – Unsigned type data coding	16
Table 2 – UINT16 type data coding	17
Table 3 – TIMEV type data coding	17
Table 4 – DATE2000 type coding	17
Table 5 – TIMEOFDAY type coding	18
Table 6 – TIMEDIFFER type coding	18
Table 7 – STRING type data coding	18
Table 8 – Physical device configuration information table	25
Table 9 – System configuration related information table	26
Table 10 – PhL synchronization management information table	27
Table 11 – System diagnosis and maintenance related information table	28
Table 12 – Physical communication resource management information table	29
Table 13 – Protocol stack version management information table	31
Table 14 – Link node management information table	32
Table 15 – Link timeout management information table	33
Table 16 – Time management information table	34
Table 17 – Network management information table	35
Table 18 – AUTBUS TMs and corresponding parameters	42
Table 19 – Working mode under carrier mode A	43

Table 20 – Working mode under carrier mode B	44
Table 21 – Ph-Param service primitives and parameters	45
Table 22 – Ph-Param service primitive parameter description	45
Table 23 – Ph-Data service primitives and parameters	45
Table 24 – Ph-Data service primitive parameter description	46
Table 25 – Ph-Clock-Sync service primitives and parameters	46
Table 26 – Ph-Clock-Sync service primitive parameter description	46
Table 27 – Ph-RESET primitives and parameters	48
Table 28 – Ph-RESET service primitive parameter description	48
Table 29 – Ph-SET-VALUE primitives and parameters	48
Table 30 – Ph-SET-VALUE primitive parameter status description	48
Table 31 – Ph-GET-VALUE service primitives and parameters	48
Table 32 – Ph-EVENT service primitive and parameters	49
Table 33 – Ph-SYNC service primitives and parameters	49
Table 34 – RS code mode	54
Table 35 – Convolutional code mode	55
Table 36 – Bit interleaving parameters	56
Table 37 – OFDM configuration parameters	57
Table 38 – Modulation Coding Scheme in carrier mode A	58
Table 39 – Modulation Coding Scheme in carrier mode B	58
Table 40 – Minimum services at MDS-MAU interface	59
Table 41 – Allowable constellation diagram errors in different modulation modes	61
Table 42 – System transmission parameters	62
Table 43 – Transmission medium interface	62
Table 44 – NodeID address assignment of AUTOBUS DLL	67
Table 45 – NodeID and MAC address mapping	68
Table 46 – Members of Multicast Group Mapping Table Description	68
Table 47 – CLMDTA service primitives and parameters	74
Table 48 – CLMDTA service primitive parameter description	74
Table 49 – CLMDTNA service primitives and parameters	76
Table 50 – CLMDTNA service primitive parameter description	76
Table 51 – CLMDTRA service primitives and parameters	78
Table 52 – CLMDTRA service primitive parameter description	78
Table 53 – CLMDTRRNA service primitives and parameters	80
Table 54 – CLMDTRRNA service primitive parameter description	80
Table 55 – CMDTA service primitives and parameters	82
Table 56 – CMDTA service primitive parameter description	82
Table 57 – CMDTNA service primitives and parameters	84
Table 58 – CMDTNA service primitive parameter description	84
Table 59 – DLS status output value description	85
Table 60 – Data-link config management service	86
Table 61 – Data-link config management service primitive parameter description	86
Table 62 – CFG_PARAM_INFO structure description	87

Table 63 – TIMEOUT_CFG structure description	88
Table 64 – GROUP_IDMAP_S structure description.....	88
Table 65 – NODEID_MAC_S structure description	88
Table 66 – COMM_RES_CFG structure description	89
Table 67 – Data-link discovery service primitives and parameters.....	91
Table 68 – Data-link discovery management service primitive parameter description	91
Table 69 – NODE_MGT_INFO_S structure description	92
Table 70 – Data-link maintenance service primitives and parameters.....	94
Table 71 – Data-link maintenance service primitive parameter description	94
Table 72 – DIAG_INFO_S structure member parameter description	94
Table 73 – Data-link establish management service primitives and parameters.....	96
Table 74 – Data-link establish management service parameter description	96
Table 75 – CH_RES_INFO_S structure member description	97
Table 76 – Data-link release management service primitives and parameters	98
Table 77 – Data-link release management service primitive parameter description	99
Table 78 – Data-link update management service primitives and parameters	100
Table 79 – Data-link update management service primitive parameter description	100
Table 80 – DLMS status return value description	101
Table 81 – Delay measurement service primitives and parameters	102
Table 82 – Delay measurement service primitive parameter description.....	103
Table 83 – Clock synchronization service primitives and parameters	104
Table 84 – Clock synchronization service primitive parameter description.....	105
Table 85 – Clock interrupt service primitives and parameters.....	106
Table 86 – Clock interrupt service primitive parameter description	106
Table 87 – DLCSS status return value description	107
Table 88 – DLDE state transition description	131
Table 89 – DLME state machine state transition description	133
Table 90 – DLCE state machine state transition description	134
Table 91 – Description of time information structure TIMEINFO_S	138
Table 92 – Description of clock option information structure CLOCK_OPTION_INFO_S.....	138
Table 93 – Description of network configuration parameter structure NETWORK_CFG_PARA_S	139
Table 94 – Description of MAC mapping table NETWORKID_MAC_MAP_TABLE_S	139
Table 95 – Description of IP mapping table structure IP_MAP_TABLE_S	140
Table 96 – Service data mapping table parameter list	142
Table 97 – Communication models supported by AUTOBUS AL ASEs.....	145
Table 98 – Application service interface return value list.....	145
Table 99 – RT Data service interface parameters.....	146
Table 100 – nRT data service interface parameters	148
Table 101 – Timing service interface parameters	151
Table 102 – Resource service interface parameters.....	154
Table 103 – Addressing service interface parameters	157
Table 104 – Management service interface parameters	161

Table 105 – Virtual service interface parameters	168
Table 106 – Read service parameters.....	170
Table 107 – Write service parameters.....	170
Table 108 – Application service errcode description.....	171
Table 109 – AL service type flag encoding.....	172
Table 110 – State transition table of publisher state machine.....	177
Table 111 – State transition table of subscriber state machine.....	178
Table 112 – State transition table of client state machine.....	179
Table 113 – State transition table of server state machine	180
Table 114 – State transition table of ALCE state machine	181
Table 115 – State transition table of ALDE state machine	182
Table 116 – State transition table of ALME state machine.....	183
Table B.1 – RS code (255,247) generating polynomial.....	187
Table B.2 – RS code (255,239) generating polynomial.....	187
Table B.3 – RS code (255,223) generating polynomial.....	187
Table C.1 – Example of NodeID and MAC address mapping table saved on TN.....	188
Table H.1 – Virtual bus environment node NETWORKID list	195

IECNORM.COM : Click to view the full PDF of IEC PAS 62356:2020

INTERNATIONAL ELECTROTECHNICAL COMMISSION

INDUSTRIAL COMMUNICATION NETWORKS – BROADBAND FIELDBUS SPECIFICATION – AUTBUS

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.

A PAS is an intermediate specification made available to the public and needing a lower level of consensus than an International Standard to be approved by vote (simple majority).

IEC PAS 62356 has been processed by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

The text of this PAS is based on the following document:

This PAS was approved for publication by the P-members of the committee concerned as indicated in the following document

Draft PAS	Report on voting
65C/973/DPAS	65C/983/RVDPAS

Following publication of this PAS, which is a pre-standard publication, the technical committee or subcommittee concerned may transform it into an International Standard.

This PAS shall remain valid for an initial maximum period of 2 years starting from the publication date. The validity may be extended for a single period up to a maximum of 2 years, at the end of which it shall be published as another type of normative document, or shall be withdrawn.

INDUSTRIAL COMMUNICATION NETWORKS – BROADBAND FIELDBUS SPECIFICATION – AUTBUS

1 Scope

This document defines the broadband fieldbus specification AUTBUS. AUTBUS implements real-time, high reliability and deterministic transmission and application for both industrial fieldbus data and ISO/IEC/IEEE 8802-3 Ethernet data by shared medium bus.

This document explains the structure and content of AUTBUS, and describes the definition and specification of Physical Layer (PhL) protocol / service, Data-link Layer (DLL) protocol / service and Application Layer (AL) protocol / service of AUTBUS.

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.

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/IEEE 8802-3:2017, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Standard for Ethernet*

3 Terms and definitions, abbreviated terms and symbols

3.1 Terms and definitions

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

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

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

3.1.1 cyclic

a term used to describe recurring events in a regular manner

3.1.2 control device

refers to the physical entity with logical operation, timing, calculation and other functions to control all kinds of field device

3.1.3**field device**

a physical entity connected to a process or factory device, and at least one signal element communicates with other device over cable

3.1.4**management node**

a device that allocates and manages AUTOBUS PhL communication resources

3.1.5**terminal node**

a device that receives the AUTOBUS PhL communication resources allocated by the management node and communicates through the allocated resources

3.1.6**clock synchronization**

clock calibration of the Terminal Node device

3.1.7**clock query**

Get clock information from the Management Node or Terminal Node device

3.1.8**cyclic time**

the time of cyclic processing of a device or module

3.1.9**RT data**

data sensitive to time deterministic requirements

3.1.10**non-RT data**

Data insensitive to time deterministic requirements

3.1.11**OFDM Symbol**

a symbol contains all subcarriers in the frequency domain, the minimum data transmission unit in time domain in Orthogonal Frequency Division Multiplexing (OFDM) technology

3.1.12**upper-side band**

a set of subcarriers with a frequency of 16,896 ~ 32,256 MHz on an OFDM symbol

3.1.13**lower-side band**

a set of subcarriers with a frequency of 1,536 ~ 16,896 MHz on an OFDM symbol

3.1.14**half-side band**

the upper-side band or lower-side band of an OFDM symbol

3.1.15**carrier mode**

data transfer mode over data subframe made up of OFDM symbols

3.1.16

working mode

OFDM operation in a specific transmission mode, Reed-Solomon coding, convolutional coding, and modulation mode

3.1.17

two-wire non-bridged

the devices are connected by physical media with two wires, without being connected through a bridge device

3.1.18

cyclic frame

the physical layer signal frame is processed in a cyclic way, and the user should configure the number of continuous cyclic signal frames

3.1.19

cyclic symbol

the method for physical layer processing of OFDM symbols in signal frames, user should configure the number of symbols for continuous cyclic processing

3.1.20

code block

an octet sequence consisting of a certain number of octets is the basic data unit for data exchange between the PhL entity and the data-link layer entity

3.1.21

data transmission channel

a logical channel of the data-link layer mapped by the determined physical communication resource for transmitting data-link layer data frames

3.2 Abbreviated terms

For the purposes of this document, the following abbreviated terms apply.

AL	Application layer
ALCE	Application layer clock entity
ALDE	Application layer data entity
ALME	Application layer management entity
ALS	Application layer service
AE	Application entity
AP	Application process
APO	AP object
AR	Application relationship
ASE	Application service elements
BPSK	Binary phase shift keying
CB	Code block
C/S	Client/Server
CLMDTA	Connection-less Mode Data Transmission with Acknowledge
CLMDTNA	Connection-less Mode Data Transmission with No Acknowledge
CLMDTRA	Connection-less Mode Data Transmission with Request and Acknowledge
CLMDTRRNA	Connection-less Mode Data Transmission with Request and Response but No Acknowledge
CMDTA	Connection Mode Data Transmission with Acknowledge

CMDTNA	Connection Mode Data Transmission with No Acknowledge
DCE	Data communication equipment
DIS	DCE independent sublayer
DL-	Data-link layer (as a prefix)
DLCSS	Data-link clock synchronization service
DLCE	Data-link clock entity
DLDE	Data-link data entity
DLE	Data-link entity
DLL	Data-linkL layer
DLM	Data-link management
DLME	Data-link managemnent entity
DLMS	Data-link management service
DLMS-user	Data-link managemet service user
DLS	Data-link service
DLP	Data-link protocol
DLS-user	Data-link service user
DLSDU	Data-link service data unit
DLPDU	Data-link protocol data unit
DTC	Data transmission channel
DTE	Data terminal equipment
DTS	Data transmission service
SynS	Synchronize signal
MAC	Medium access control
MCS	Modulation coding scheme
MN	Management node
NodeID	Node identification
nRT	non-Real-Time
OFDM	Orthogonal Frequency Division Multiplexing
PhL	Physical layer
PhL-	Physical layer (as a prefix)
PhLDE	Physical layer Data Entity
PhLME	Physical layer management entity
PhPCI	Physical layer protocol control information
PhPDU	Physical layer protocol data unit
PhS	Physical layer service
PhSAP	Physical layer service access point
PhSDU	Physical layer service data unit
QAM	Quadrature amplitude modulation
RF	Radio frequency
RMS	Root mean square
RS	Reed-Solomon
RT	Real-Time
RTA	Real-Time acyclic

RTC	Real-Time ayclic
SAP	Service access point
SynS	Synchronization signal
TM	Transmit mode
TN	Terminal node

3.3 Symbols

T_S	Sampling interval
N_{CBL}	Bit interleaving block length
N_{COL}	Column number of bits interleaving
N_{BPS}	Bit interleaving symbol bit number
N_{SD}	Number of subcarriers of an OFDM symbol
N_{SYM}	Number of OFDM symbols of a signal frame

4 Data type

4.1 Overview

AUTBUS data coding specifies the machine independent syntax for the data conveyed by each layer services. AUTBUS supports the definition and transmission of basic and complex data types.

The basic type cannot be divided into smaller elemental types. Complex data types are composed of several basic types and other structured types, and the complexity and depth of nesting are not constrained by this document.

4.2 Basic data type

4.2.1 Unsigned type coding (UINT)

The value of AUTBUS unsigned type data is an unsigned integer, Table 1 shows the coding.

Table 1 – Unsigned type data coding

Type name	Value range	length
UINT8	0 ~ (2 ⁸ -1)	one octet
UINT16	0 ~ (2 ¹⁶ -1)	two octets
UINT32	0 ~ (2 ³² -1)	four octets
UINT64	0 ~ (2 ⁶⁴ -1)	eight octets
UINT128	0 ~ (2 ¹²⁸ -1)	sixteen octets

Table 2 shows the coding of each bit of the octet for UINT16. When data is transferred, the MSB of the highest valid octet of this type of data is transmitted at first.

Table 2 – UINT16 type data coding

octet	bit							
	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
2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

4.2.2 Time type coding (TIMEV)

The time data type is an unsigned integer of 64 bits, which represents the time incremented with 1 ns. Table 3 shows the coding.

Table 3 – 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

4.2.3 Date type coding (DATE2000)

The DATE2000 data type length is seven octets, indicating date and time. Table 4 shows the meaning of each octet.

Table 4 – DATE2000 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 a 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, 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 0 ~ 50 correspond to 2000 – 2050, the values 51 ~ 99 correspond to 1951 – 1999.

4.2.4 Time of day type coding (TIMEOFDAY)

TIMEOFDAY data type length is six octets and indicates the time of day. Table 5 shows the meaning of each octet.

Table 5 – 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	

4.2.5 Time difference type coding (TIMEDIFFER)

TIMEDIFFER data type length is six octets and indicates time difference. Table 6 shows the meaning of each octet.

Table 6 – 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 number of days of deviation
6	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	

4.2.6 String type coding (STRING)

STRING type is a set of octet in order. The length is not fixed, it is 1~ n octet. Table 7 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 7 – STRING 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

4.2.7 Bitmap type coding (BITMAP)

BITMAP type length is 2~256 octets, includes Width and Mask_Info information. Figure 1 shows the bitmap type coding.

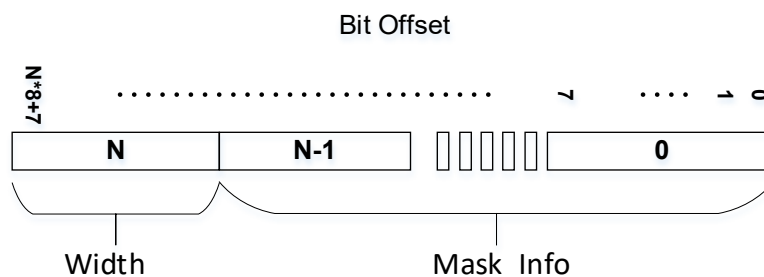


Figure 1 – BITMAP data type

Description:

Width: One octet, its value indicates the number of the octets of Mask_Info.

Mask_Info: The information represented by bit, the octet length is Width value and the bit length is Width×8.

4.3 Complex type

4.3.1 Struct type coding (STRUCT)

STRUCT type is an ordered set of different basic data types or structured data. The data of these basic data types or structured data types is called a member of the struct. The struct type data may be accessed as a whole, or a member of the struct data may be accessed only by specifying the list of members.

4.3.2 Array type coding (ARRAY)

ARRAY type is an ordered set of elements of the same type. The data type of array elements is not limited in this specification, but each element belong to the same type. Once an array is defined, the number of elements in this array cannot be changed.

5 AUTBUS overview

5.1 Overview

AUTBUS is an industrial fieldbus with two-wire medium and 100 Mbps bandwidth, which is used for process control and distributed control applications and is compatible with ISO/IEC/IEEE 8802-3, IPv4 and IPv6. The AUTBUS provides fixed-bandwidth data service and variable-bandwidth data service for periodic real-time process data, burst real-time control, alarm data and non-real-time data through pre-configuration or dynamic application of physical communication resource. It supports network slicing technology and it implements the transmission and application of the existing fieldbus based on the AUTBUS network. AUTBUS adopts Orthogonal Frequency Division Multiplexing (OFDM) technology, and has feature. It is a deterministic fieldbus communication network based on high-precision clock synchronization. It uses a time-triggered mechanism and the optimal time for data delivery is 8 μs.

5.2 AUTBUS protocol stack architecture

The AUTBUS protocol follows the ISO/IEC 7498-1 OSI reference model. The AUTBUS defines the Physical Layer (PhL), Data-link Layer (DLL) and Application Layer (AL). The AUTBUS AL covers the basic functions of layer 3 to layer 7 of OSI, it also meets the requirements of compatible application services based on other fieldbuses and Ethernet, and AUTBUS AL provides related common features to simplify user operation. Figure 2 shows the corresponding relationship between AUTBUS protocol stack and OSI reference mode.

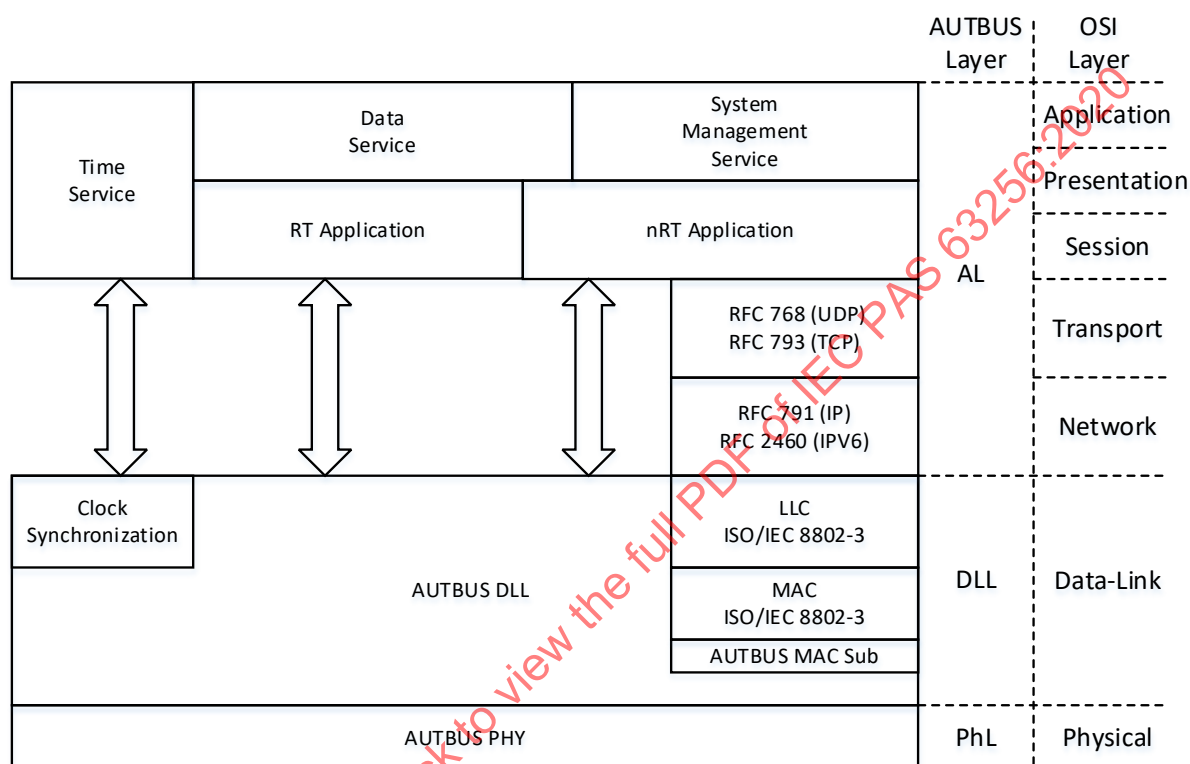


Figure 2 – Corresponding relationship between AUTBUS protocol stack and OSI reference model

AUTBUS PhL is based on OFDM technology, adopts two-wire medium transmission, supports full duplex mode. AUTBUS PhL implements physical signal encoding and decoding through RS code, convolutional code and interleaving technology. The communication resources allocated in time domain and frequency domain are used to provide highly reliable synchronous transmission service. AUTBUS PhL provides system management service interface to meet the flexible configuration requirements of system management users.

AUTBUS DLL uses the service interface provided by PhL to implement connection and connection-less data transmission services between AUTBUS devices. AUTBUS DLL provides clock synchronization, data transmission and system management services to users. Based on AUTBUS DLL system management service, it provides the configuration, discovery and maintenance of AUTBUS device, as well as the allocation and management of physical communication resources. AUTBUS DLL provides an interface compatible with ISO/IEC/IEEE 8802-3.

The AUTBUS AL uses the service interface provided by the DLL, defines a communication model meeting the requirement of industrial automation application, provides real-time cyclic process data, real-time acyclic alarm data, non-real-time data transmission service, and AL time synchronization and network configuration, diagnostics and logging features. The AUTBUS AL

also provides network slicing to realize the integration of the AUTOBUS network and other heterogeneous fieldbus networks.

The PhL, DLL and AL in the AUTOBUS protocol stack have corresponding data entity and management entity, each layer provides a corresponding service access point to the upper layer user, and each layer in the AUTOBUS protocol stack provides a service access point facing the system management. The DLL and the AL in the AUTOBUS protocol stack both have clock entity, which respectively provide a clock entity service access point for the upper layer user. The gray parts in Figure 3 denote the AUTOBUS protocol stack content.

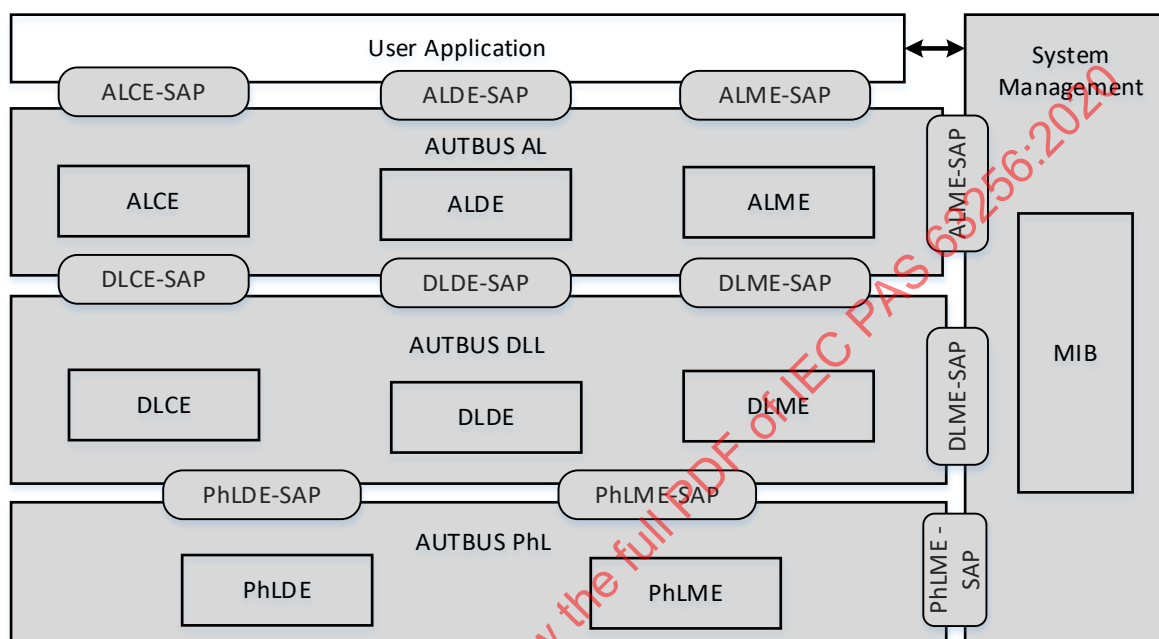


Figure 3 – AUTOBUS protocol stack architecture

Description:

- PhLDE: The PhL Data Entity, processing interface data and converting them into physical signals that should be transmitted by the medium.
- PhLME: PhL Management Entity, performing PhL system management including synchronization.
- PhLDE-SAP: The PhLDE Service Access Point, the PhLDE implements the interaction processing with the data unit of the DLL through this service access point, including the interactive processing of the DLL clock synchronization data.
- PhLME-SAP: The PhLME Service Access Point, the PhLME provides system management services to the system management or DLL through this service access point.
- DLDE: DLL Data Entity, processing DLL data, encapsulate and parsing the DLPDU.
- DLME: DLL Management Entity, performing DLL system management.
- DLCE: DLL Clock Entity, performing DLL clock synchronization processing.
- DLDE-SAP: DLDE Service Access Point, the DLDE provides data transmission service to the AL through this service access point.
- DLME-SAP: DLME Service Access Point, the DLME provides system management service to system management or AL through this service access point.
- DLCE-SAP: DLCE Service Access Point, the DLCE provides clock synchronization service to AL through this service access point.
- ALDE: AL Data Entity, process the AL data, encapsulate and parsing the AL message.

- ALME: AL Management Entity, performing AL system management
- ALCE: AL Clock Entity, performing AL time synchronization service processing
- ALDE-SAP: The ALDE Service Access Point, the ALDE provides the data transmission service to the user layer through this service access point.
- ALME-SAP: The ALME Service Access Point, the ALME provides system management services to the system management or user layer through this service access point.
- ALCE-SAP: The ALCE Service Access Point, the ALCE provides the time synchronization service to the user layer through this service access point.

5.3 Device type

5.3.1 Overview

AUTBUS devices distinguish between control device and field device according to the control role of device in industrial automation applications. AUTBUS devices distinguish management node and terminal node according to the management role of the device in the process of AUTBUS physical communication resource allocation.

5.3.2 Control Device

It refers to the physical entity that has the functions of logic operation, timing, calculation and so on, and controls all kinds of field device. The common device is programmable logic controller.

5.3.3 Field Device

It refers to the physical entity connected to the process or factory device, and at least one signal element communicates with other device over cable. The field devices distinguish input device and output device.

Input device: All kinds of transmitters and sensors, such as temperature transmitters, pressure sensors, etc.

Output device: All kinds of actuators and driving unit, such as switches, valves.

5.3.4 Management Node

The AUTBUS Management Node (MN) is responsible for the unified allocation, recycling and scheduling of physical communication resources in the network. The AUTBUS MN may be the control device or the field device, or the device that is only used for physical communication resource allocation and scheduling and not involved in the industrial automation application. There is one and only one active MN in an AUTBUS network.

5.3.5 Terminal Node

The Terminal Node (TN) should only receive physical communication resources allocated by MN and communicate with other devices through the allocated resources. The TN may be the control device or the field device.

5.4 Network topology

AUTBUS supports bus and ring topologies. Figure 4 shows the network topology. An AUTBUS network supports 254 nodes.

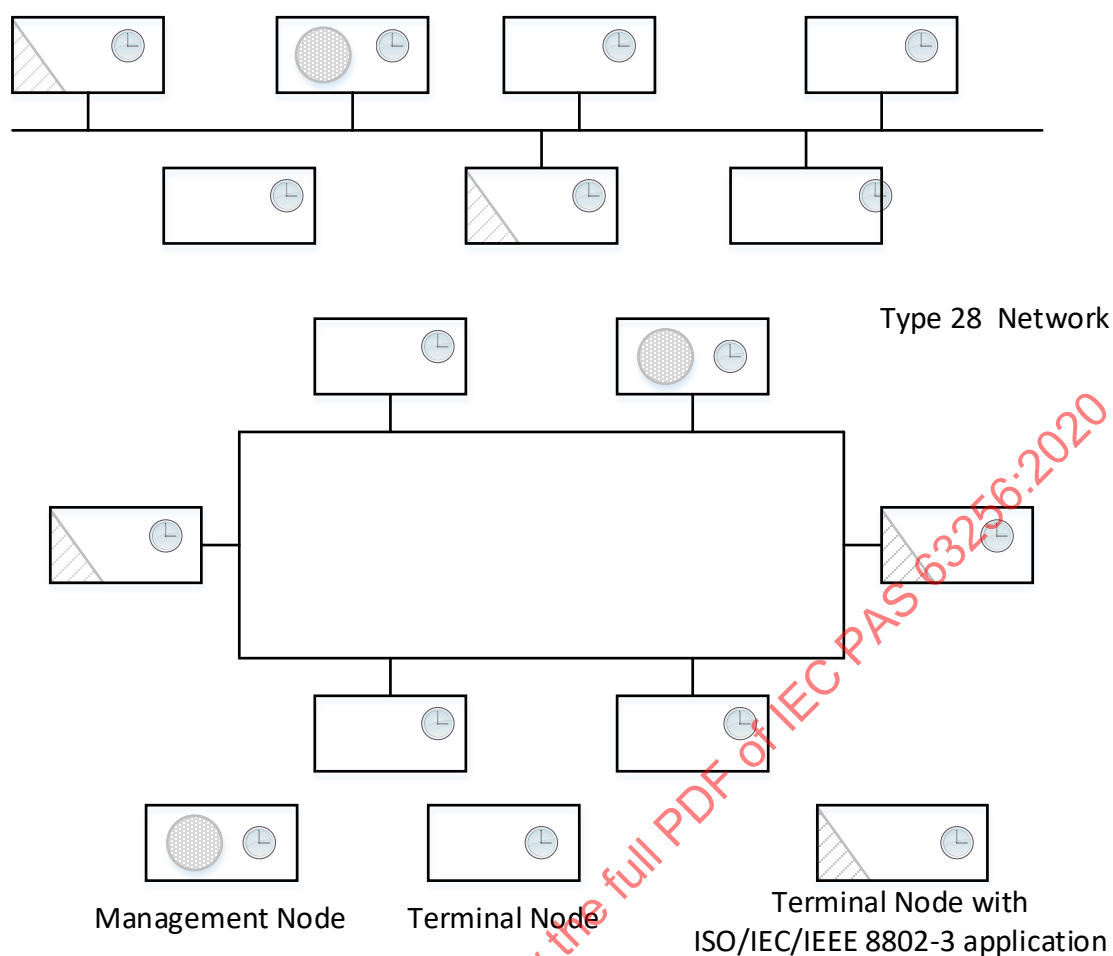


Figure 4 – AUTOBUS network topology

6 System management

6.1 Overview

AUTOBUS defines the management service access point in the PhL, the DLL and the AL (refer to Figure 3). AUTOBUS defines the corresponding management information table for each layer (see 6.3.1 to 6.3.3). The AUTOBUS network adopts a centralized and distributed management architecture. The MN is used for unified configuration, discovery and maintenance of the network, and allows the TN to be found and accessed to the network, and response to the configuration and maintenance request of the MN. The deployment of the network should be completed through these interactions. AUTOBUS network PhL communication resources are centrally allocated and scheduled by the MN, the TN can identify and process the communication resources of the whole network and be responsible for the management of its related communication resources. The TN should also interact with the MN to implement the application, release and reset of new communication resources.

6.2 System management process

The AUTOBUS implement management of management entities at the corresponding layer based on the management service interface provided by the AL, DLL and the PhL. The remote peer entity is managed through the corresponding management message and the management protocol frame. Figure 5 shows the AUTOBUS system management process.

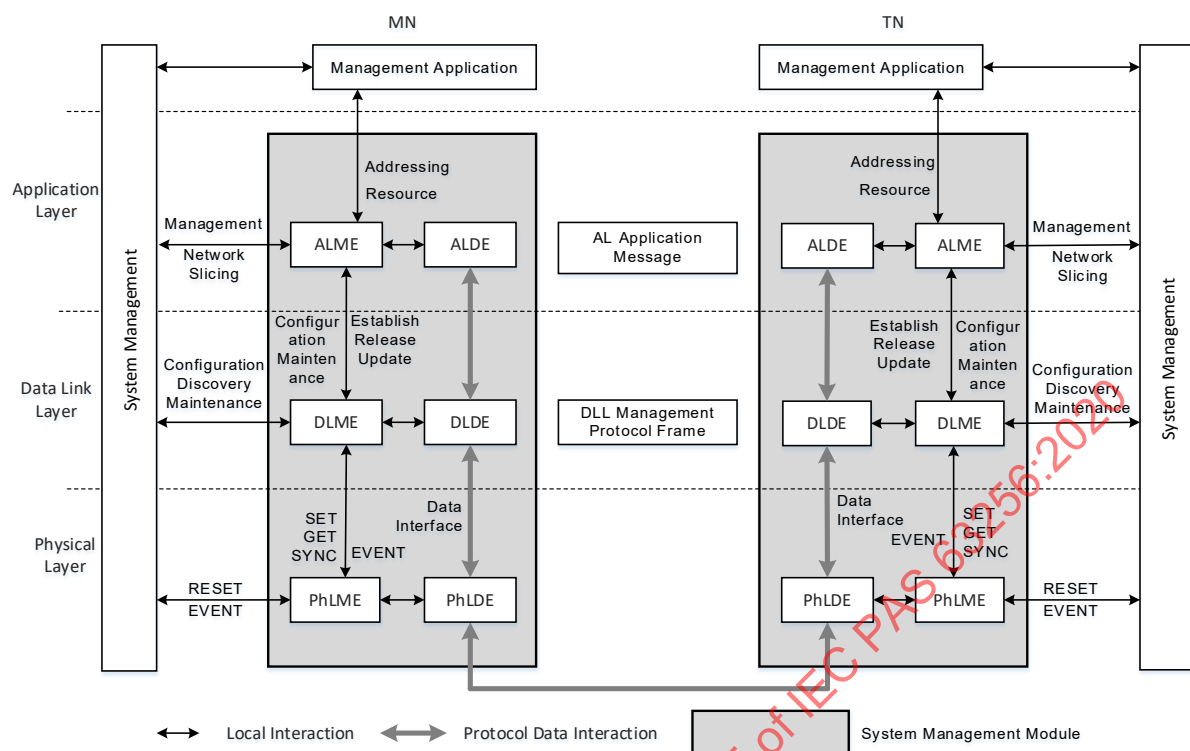


Figure 5 – AUTOBUS system management process

The AUTOBUS PhLME provides basic operations for reset, set, get and synchronization of the local PhL entity. The local PhL entity should convert and encode the corresponding DLL protocol data based on the management service request of the local DLL, and send the protocol data to the remote physical entity that connected to the medium. The remote physical entity should receive the protocol data and decode them for further processing. The AUTOBUS DLME provides network configuration, discovery and maintenance of the link layer. The AUTOBUS DLME implements the service function of establishing, releasing and maintaining the DTC resources based on the system management services provided by the PhL, to meet the configuration management of communication resources by AL users. The system management information table related to the PhL, DLL and AL interact with the corresponding layer entity of the local device to implement the corresponding system management function. The corresponding layer entity should request or response the remote peer entity through the message and DLPDU.

6.3 System management information table

6.3.1 PhL related management information table

6.3.1.1 General

The PhL management information includes physical device configuration information, system configuration information, PhL synchronization management information, system diagnosis and maintenance information and physical communication resource management information. The system management user or data-link management user should access the corresponding information table through the service interface provided by the PhL.

6.3.1.2 Physical device configuration information table

Table 8 – Physical device configuration information table

Parameter index	Parameter name	RW	Data Type	Description
0x0000	node_id	R/W	UINT16	Device node NodeID number High 8bit is reserved, low 8bit is valid Number valid range: 0~255 See addressing in 8.1.2.2. 0: AUTOBUS management node 1~253: AUTOBUS terminal node or multicast group NodeID. Multicast group NodeID decreases from 253 in turn., max support 16 groups 254: Reserve to nodes that are not assigned NodeID 255: Broadcast address number
0x0001	mcn_bmp	RW	UINT16	The multicast group to which the node belongs bit 0 ~ bit 15: Corresponding broadcast group 238~253 The corresponding bit set to 0 indicates that the node device does not belong to the multicast group, and set to 1 indicates that the node device belongs to the multicast group.
0x0002	device_role	R/W	UINT8	Device type node role represented by bit bit 0 indicates whether it is management node bit 1 indicates whether it is terminal node The corresponding bit set to 1 indicates it is corresponding role, and set to 0 indicates it is not corresponding role The values of bit 0 and bit 1 on the working node should be different bit 2 ~ bit 7: reserved
0x0003	device_state	RO	UINT8	The state of the node at different phases represented by bit bit 0: Initialization state bit 1: configuration state bit 2: running state bit 3: synchronization state The corresponding bit set to 0 indicates state is normal, set to 1 indicates state is abnormal. bit 5 ~ bit 7: reserved
0x0004 ~ 0x008F	Reserved			Protocol reserved
0x0090 ~ 0x00FF	Reserved			Reserved for equipment manufacturer

6.3.1.3 System configuration related information table

Table 9 defines system configuration related information.

Table 9 – System configuration related information table

Parameter Index	Parameter name	RW	Data Type	Description
0x0100	trans_mode	R/W	UINT8	Transmit mode Value range: 0 ~ 255, see Table 18 for valid value 0: Corresponding transmit mode 0 1: Corresponding transmit mode 1 2: Corresponding transmit mode 2 3: Corresponding transmit mode 3 The default value is 0 Other values are reserved
0x0101	work_mode_up	R/W	UINT8	Upper-side band working mode Value range: 0 ~ 255, valid value range: 0 ~ 21, see 7.2.6 working mode. 0 ~ 9: corresponding working mode A0 ~ A9; 10 ~ 21: corresponding working mode B0 ~ B11; Default value is 0 Other values are reserved
0x0102	work_mode_down	R/W	UINT8	Lower-side band working mode value range and value description are same with work_mode_up
0x0103	cb_up	RO	UINT8	Upper-side band coding mode Value range: 0 ~ 255 High 5bit value corresponds to RS coding Mode, see Table 34, valid value rang 0 ~ 11. Other values are reserved. Low 3bit value corresponds to convolutional code coding mode, see table 23, valid value range 0 ~ 2. Other values are reserved.
0x0104	cb_down	RO	UINT8	Lower-side band coding mode Value range and value description are same with cb_up.
0x0105	inter_mode	RO	UINT8	Interleaving mode Value range 0 ~ 255 High 4 bits indicate the upper side-band interleaving mode , low 4 bits indicate the lower-side band interleaving mode, the valid value range of every 4 bits is 0 ~ 4, and other values are reserved, see Table 35 for valid value.
0x0106	power_ratio	R/W	UINT8	Side-band power ratio Value range 0 ~ 255 0: Indicates upper and lower-side band power average distribution, other values are reserved.
0x0107	band_flag	R/W	UINT8	Side-band occupation flag represented by bit bit 0: lower-side band occupation flag. bit 1: upper-side band occupation flag. The corresponding bit set to 0 indicates that it is not occupied, set to 1 indicates it is occupied bit 2 ~ bit 7: reserved

Parameter Index	Parameter name	RW	Data Type	Description
0x0108	net_state	RO	UINT8	AUTBUS network state Value range: 0 ~ 255 0: Initialization state 1: Normal working state Other values are reserved
0x0109	reset_en	WO	UINT8	Reset flag bit represented by bit bit 0: PhL reset flag The corresponding bit set to 0 indicates disable, set to 1 indicates enable. bit 1 ~ bit 7: reserved
0x010A	lb_mode	R/W	UINT8	Loopback mode setting indicated by bit. bit 0: Loopback at the DCE and DTE interfaces bit 1: Loopback at MAU near the cable side. The corresponding bit set to 0 indicates disable, set to 1 indicates enable. bit 2 ~ bit 7: reserved
0x010B ~ 0x018F	reserved			Protocol reserved
0x0190 ~ 0x01FF	reserved			Reserved for equipment manufacturer

6.3.1.4 PhL synchronization management information table

Table 10 defines PhL synchronization management information.

Table 10 – PhL synchronization management information table

Parameter Index	Parameter name	RW	Data Type	Description
0x0200	sync_delay	RW	UINT8	Synchronization delay information Value range: 0 ~ 255 0: Indicates that the synchronization is normal or it is synchronizing Other values N: should wait for N signal frames to start synchronous processing
0x0201	cycle_id	RO	UINT16	Cycle ID value Value range: 0 ~ 65 536 Valid value range is 0 ~ 16 383 Other values are reserved
0x0202	symbol_id	RO	UINT8	OFDM symbol ID Value range: 0 ~ 255 Valid value range is 0 ~ 63 Other values are reserved
0x0203	frame_id	RO	UINT64	Signal frame ID Value range: 0 ~ (2 ⁴⁴ -1) Other values are reserved

Parameter Index	Parameter name	RW	Data Type	Description
0x0204	time_mode	R/W	UINT8	The time information mode currently supported by the node. Value range: 0 ~ 255 0: TAI mode 1: UTC mode Other values are reserved
0x0205	time_high	RO	UINT64	Time information of the current above second level of the node. Value range: 0 ~ (2 ⁴⁸ -1) 0: Information is not available Other values are reserved
0x0206	time_low	RO	UINT64	Time information of the current below second level of the node. Value range: 0 ~ (2 ⁴⁸ -1) Other values are reserved
0x0207 ~ 0x028F	Reserved			Protocol reserved
0x0290 ~ 0x02FF	Reserved			Reserved for equipment manufacturer
0x0300 ~ 0x03FD	delay_cn	RO	UINT32	Transmission delay (calculated by the management node) between the management node and the terminal node and the unit is ns. 254 Table items correspond to the delay of node id 0 to node id 253. Value range: 0 ~ (2 ³² -1)
0x03FE ~ 0x03FF	reserved			Protocol reserved

6.3.1.5 System diagnosis and maintenance related information table

Table 11 defines system diagnosis and maintenance related information.

Table 11 – System diagnosis and maintenance related information table

Parameter Index	Parameter name	RW	Data Type	Description
0x0400	fm_sync_err	RO	UINT8	Synchronization error state Value range: 0 ~ 255 0: No error 1: Frame synchronization error Other values are reserved
0x0401	symbol_err	RO	UINT64	Current OFDM symbol error of node (Frame ID and Symbol ID should be recorded when OFDM symbol error occurs). Value range: 0 ~ (2 ⁶⁴ -1) 0: no OFDM symbol error Other values: Frame ID and Symbol ID

Parameter Index	Parameter name	RW	Data Type	Description
0x0402	band_err	RO	UINT64	The current side-band error of the node (Frame ID, Symbol ID and upper or lower side-band should be recorded when the error occurs). Value range: 0 ~ (2 ⁶⁴ -1) 0: No side-band error Others: Frame ID and Symbol ID
0x0403	data_err	RO	UINT64	The current node has a data transmission error and records the Frame ID, Symbol ID and side-band information. Value range: 0 ~ (2 ⁶⁴ -1) 0: No transmitting error Others: Frame ID and Symbol ID and side-band information.
0x0404 ~ 0x048F	reserved			Protocol reserved
0x0490 ~ 0x04FF	reserved			Reserve for equipment manufacturer

6.3.1.6 Physical communication resource management information table

Table 12 defines Physical communication resource management information.

Table 12 – Physical communication resource management information table

Parameter Index	Parameter name	RW	Data Type	Description
0x0500	sym_down_num	R/W	UINT8	Symbol number of downlink subframe Value range: 2 ~ 62 Other values are invalid
0x0501	cycle_frame_num	R/W	UINT8	Cycle frame length Value range: 0 ~ 16 Other values are reserved
0x0502 ~ 0x05FF	reserved			Protocol reserved
0x0600	res_block_id	R/W	UINT16	The resource block ^[1] number used by the node. Value range: 0 ~ 65 535
0x0601	sequence_id	R/W	UINT8	Resource sub-block ^[1] Sequence Number. Value range: 0 ~ 255
0x0602	tx_node_id	R/W	UINT8	The device node NodeID for transmitting data based on symbolic Resources. Value range: 0 ~ 255
0x0603	res_rt_attr	R/W	UINT16	Real-time attribute of the resource Value range: 0 ~ 512 0: Static resources, always occupied 1 ~ 512: Number of times of corresponding resource blocks are occupied by circulation Other values are reserved.

Parameter Index	Parameter name	RW	Data Type	Description
0x0604	sym_reserved	R/W	UINT8	Reserved field
0x0605	bd_flag	R/W	UINT8	Side-band occupation flag indicated by bit. bit 0: The upper-side band occupation flag bit 1: The lower-side band occupation flag The corresponding bit set to 0 indicates that the half-side band is not occupied, and set to 1 indicates that the half-side band is occupied. bit 2 ~ bit 7: reserved
0x0606	bd_mode	R/W	UINT8	Corresponding side-band coding mode represented by bit. The high 5 bit value corresponds to RS coding mode, see Table 34, the valid value range is 0 ~ 11. Other values are reserved. The low 3 bit value corresponds to the convolutional code coding mode, see Table 35, the valid value range is 0 ~ 2. Other values are reserved.
0x0607	sym_pos	R/W	UINT16	Symbol resource starting position Value range: 0 ~ 511 Other values are reserved.
0x0608	sym_len	R/W	UINT16	Symbol resource length Value range: 1 ~ 512 Value 0 is invalid Other values are reserved.
0x0609 ^[2]	sym_cycle_num	R/W	UINT16	Number of times symbol resources are recycled Value range: 1 ~ 512 Value 0 is invalid. Other values are reserved.
0x060A ~ 0x060F	reserved			Protocol reserved
0x0610 ~ 0x09FF	[2]	[2]	[2]	[2]
0x0A00 ~ 0x0FFF	reserved			Protocol reserved
Explanation: [1] See 7.2.1 signal frame. Node device processes the data based on the continuous half-side band or side-band symbol in the specified signal frame, the continuous symbol is a resource subblock, the device implements deterministic data communication through one or multiple resource subblocks. Multiple resource sub-blocks form a resource block. [2] The index from 0x0600 to 0x609 corresponds to a resource block information. Figure 6 shows the resource block information structure, and the index 0x060A ~ 0x060F are reserved index for the corresponding resource block. 64 resource blocks information is allowed on each physical device, that is, the corresponding valid index range of resource information management is 0x0600 to 0x09FF.				

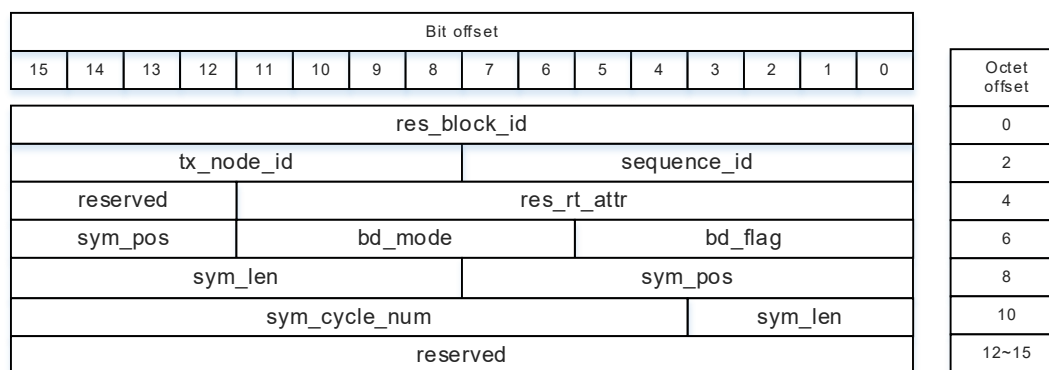


Figure 6 – Resource block information structure

6.3.2 DLL related management information table

6.3.2.1 General

The DLL management information usually exists in the storage area as software variable and should be accessed by the system management user or the AL user. The important information should be placed in the non-volatile storage area or hardware table item. The DLL related management information includes protocol stack version management information, link node management information and link timeout management information, the system management user or the AL management user should access the corresponding information table through the service interface provided by the DLL.

6.3.2.2 Protocol stack version management information table

Table 13 – Protocol stack version management information table

Parameter index	Parameter name	RW	Data Type	Description
0x1000	ps_version	RO	UINT32	The protocol version number indicated by dotted decimal notation: XX.XXXX.XX bit 31 ~ bit 24: major version number. Value range: 0 ~ 255. bit 23 ~ bit 8: secondary version number. Value range: 0 ~ 65 535. bit 7 ~ bit 0: protocol version number. Value range: 0 ~ 255. The protocol version number should be notified to all nodes to ensure that the protocol version is consistent, and the other two version numbers are used as the software version properties of the management node protocol stack for the reference of the user program verification.
0x1001	reserved			Reserved for equipment manufacturer.

6.3.2.3 Link node management information table

Table 14 – Link node management information table

Parameter index	Parameter name	RW	Data Type	Description
0x1002	node_role	R/W	UINT8	The role attributes of a link node represented according to bit. bit 0: whether it is management node bit 1: whether it is terminal node The corresponding bit set to 1 indicates it is the corresponding role, and set to 0 indicates that it is not the corresponding role bit 2 ~ bit 7: reserved
0x1003	acc_status	R/W	UINT8	Access status Value range: 0 ~ 3 0: initialization 1: runing 2: pause 3: offline Other values are reserved.
0x1004	start_nodeID	R/W	UINT16	The start node NodeID for configuration, that is, the start NodeID value corresponding to the MAC address mapping table. The high 8 bits are reserved and the low 8 bits are valid. Value range: 0 ~ 255 See Table 8
0x1005	uc_num	R/W	UINT16	Number of valid nodes accessing the network. Valid value range: 0 ~ 255 Other values are reserved.
0x1006	mc_num	R/W	UINT8	The number of valid multicast group configured by the network Value range: 0 ~ 16 Other values are reserved.
0x1007 ~ 0x107F	reserved			Protocol reserved
0x1080 ~ 0x10FF	reserved			Reserved for equipment manufacturer

6.3.2.4 Link timeout management information table

Table 15 – Link timeout management information table

Parameter index	Parameter name	RW	Data Type	Description
0x1100	T1	R/W	UINT16	Configure timeout, unit is μ s Value range: 0 ~ 65 535 Value 0 indicates that timeout check is not performed.
0x1101	T2	R/W	UINT16	Configure cycle, unit is μ s. This value should be an integer multiple of the physical signal frame period time. Value range: 0 ~ 65 535 Value 0 indicates that configuration cycle check is not performed.
0x1102	T3	R/W	UINT16	Heartbeat detection cycle, unit is μ s, this value should be an integer multiple of the physical signal frame period time Value range: 0 ~ 65 535 Value 0 indicates no heartbeat detection.
0x1103	Max_Discovery_timeout	R/W	UINT32	Discovery timeout, unit is μ s Value range: 0 ~ $(2^{32}-1)$ Value 0 indicates that the discovery timeout check is not performed.
0x1104	Max_Heartbeat_timeout	R/W	UINT8	Number of heartbeat detection timeout frame period Value range: 0 ~ 255 Value 0 indicates no heartbeat detection.
0x1105	Max_Retry_Req_Cnt	R/W	UINT8	Maximum number of retry requests Value range: 0 ~ 255 Value 0 indicates no retry request
0x1106	Max_Retry_Res_Cnt	R/W	UINT8	Maximum number of retry responses Value range: 0 ~ 255 Value 0 indicates no retry response
0x1107	Max_Retry_Cnt	R/W	UINT8	Maximum number of retry Value range: 0 ~ 255 Value 0 indicates no retry
0x1108	Max_ReqIn_Cnt	R/W	UINT8	Maximum number of random access requests. Value range: 0 ~ 255 Value 0 indicates that the number of random access requests is not limited
0x1109	Max_Establish_Req_Retry_Cnt	R/W	UINT8	Maximum number of establish requests Value range: 0 ~ 255 Value 0 indicates that the maximum number of establish request is not limited
0x110A	Max_Update_Req_Retry_Cnt	R/W	UINT8	Maximum number of update requests Value range: 0 ~ 255 Value 0 indicates that the maximum number of update request is not limited

Parameter index	Parameter name	RW	Data Type	Description
0x110B ~ 0x18FF	reserved			Protocol reserved
0x1900 ~ 0x1FFF	reserved			Reserved for equipment manufacturer

6.3.3 AL related management information table

6.3.3.1 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 table through the AL management service interface.

6.3.3.2 Time management information table

Table 16 – Time management information table

Parameter index	Parameter name	RW	Data Type	Description
0x2000	time_fmt	R/W	UINT8	Different time formats represented by bit bit 0: indicates that time format is based on hour, minute and second. bit 1: indicates that time format is based on milliseconds and microseconds. bit 2: indicates that the time format is based on second pulse. The corresponding bit set to 0 is invalid, set to 1 is valid bit 3 ~ bit 7: reserved
0x2001	time_ns	R/W	TIMEV	Time count based on nanosecond
0x2002	date2000	R/W	DATE2000	Convert to date value of year, month, day, minute and second, 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 Value range: 1 ~ 8 Value 0 is invalid. Other values are reserved
0x2006	tid_val	R/W	UINT8	Clock ID Value range: 0 ~ 7 Other values are reserved
0x2007	tid_attr	R/W	UINT8	Control-field attribute of clock Value range: 0 ~ 1 0: control clock 1: field clock Other values are reserved

Parameter index	Parameter name	RW	Data Type	Description
0x2008	tcd_lst	R/W	ARRAY	ARRAY of type UINT8 TCD command for the corresponding clock Value range see 9.3.2.3
0x2009 ~ 0x208F	reserved			Protocol reserved
0x2090 ~ 0x20FF	reserved			Reserve for equipment manufacturer

6.3.3.3 Network management information table

Table 17 – Network management information table

Parameter index	Parameter name	RW	Data Type	Description
0x2100	network_id	R/W	UINT32	Indicate NETWORKID of node or module See 9.2.2.5.1.1.
0x2101	ip_switch	R/W	UINT8	Whether supporting the IPV4 or the IPV6 represented by bit bit 0: Whether supporting IPV4 bit 1: Whether supporting IPV6 The corresponding bit set to 0 indicates that not support, set to 1 indicates that support. bit 2 ~ bit 7: reserved
0x2102	multicast_id0 ~ 15	R/W	UINT32	NETWORKID with group type indicates the multicast group to which the device or module belongs to. Each device or module belongs to a maximum of 16 multicast groups. See Table 93
0x2103	mac_addr	RW	UINT64	Unique MAC address of the device or module Value range: 0 ~ ($2^{48}-1$) Other values are reserved.
0x2104	nodeid	RW	UINT8	Node ID address of the device Value range: 0 ~ 255 See Table 44
0x2105	dctime	RW	UINT16	Data calculation cycle time, unit is μ s Value range: 0 ~ 65 535
0x2106	bctime	RW	UINT16	Bus service cycle time, unit is μ s Value range: 0 ~ 65 535
0x2107	wctime	RW	UINT16	Bus service data writing cycle time, unit is μ s Value range: 0 ~ 65 535
0x2108	rctime	RW	UINT16	Bus service data reading cycle time, unit is μ s Value range: 0 ~ 65 535
0x2109	cctime	RW	UINT16	Bus service competition cycle time, unit is μ s Value range: 0 ~ 65 535

Parameter index	Parameter name	RW	Data Type	Description
0x210A	totcnt	RW	UINT16	Number of timeout allowed by device or module Value range: 0 ~ 65 535
0x210B	rt_data_addr	RO	UINT32	Real-time data buffer start address Value range: 0 ~ (2 ³² -1)
0x210C	rt_data_size	RW	UINT32	Real-time data buffer size Value range: 0 ~ (2 ³² -1)
0x210D	nrt_data_addr	RW	UINT32	Non-real-time data buffer start address Value range: 0 ~ (2 ³² -1)
0x210E	nrt_data_size	RW	UINT32	Non-real-time data buffer size Value range: 0 ~ (2 ³² -1)
0x210F ~ 0x28FF	reserved			Protocol reserved
0x2900 ~ 0x2FFF	reserved			Reserved for equipment manufacturer

7 Physical layer

7.1 Physical Layer overview

Figure 7 shows AUTOBUS Physical Layer model

IECNORM.COM : Click to view the full PDF of IEC PAS 62356:2020

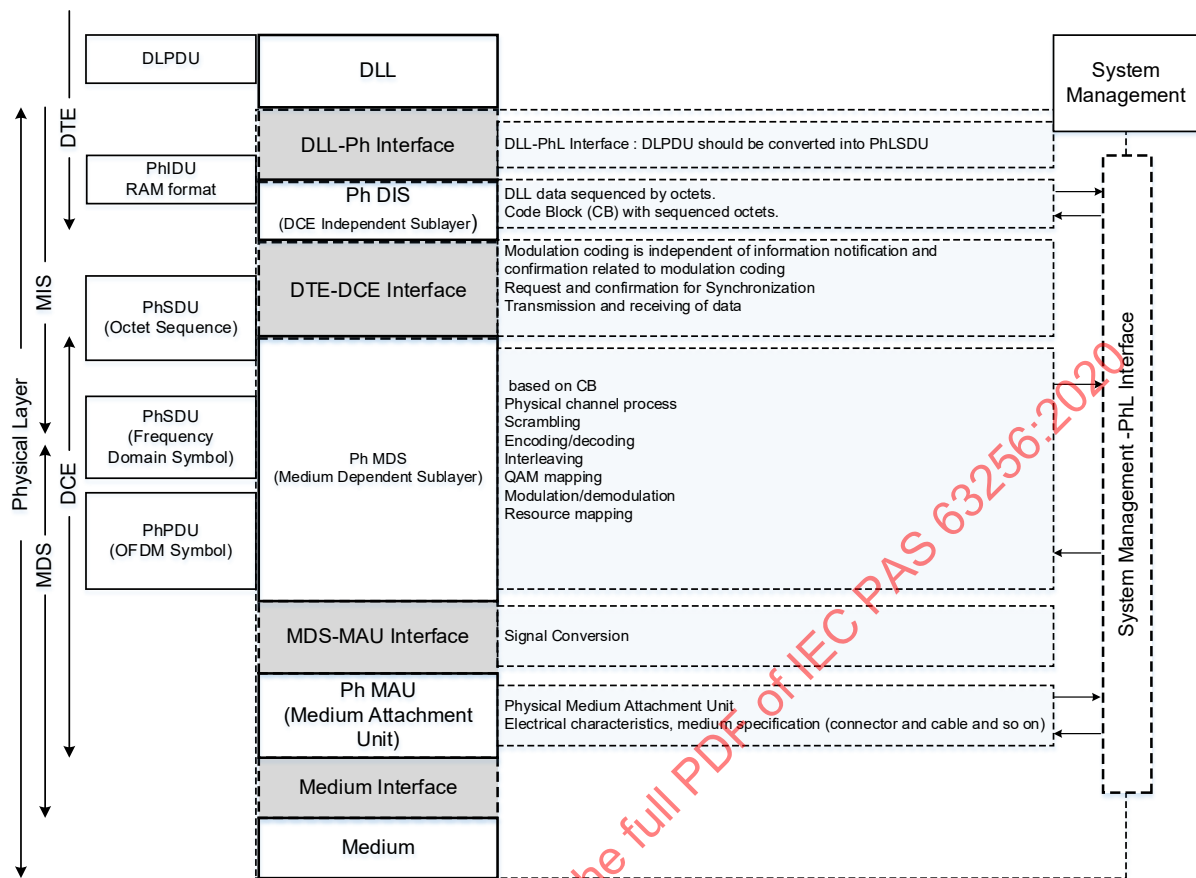


Figure 7 – AUTBUS Physical Layer model

The AUTBUS PhL receives the data unit from the DLL, converts them into service data unit that can be processed by PhL, adds the required check control information, and then generates OFDM symbols by scrambling, encoding, and mapping, then the physical signal is transmitted to the corresponding physical medium after D/A conversion. The physical signal is transmitted to one or more devices on the AUTBUS network through the medium. The physical signal is received, then A/D conversion, demodulation and decoding, and the control information is removed to recovery to the data unit which can be processed by the DLL entity.

The AUTBUS PhL entity is partitioned into a Data Terminal Equipment (DTE) and a Data Communication Equipment (DCE). DTE implements the conversion between the DLL Protocol Data Unit (DLPDU) and the PhL Service Data Unit (PhLSDU). DCE does the OFDM processing to the PhLSDU, generates the OFDM symbols and converts them into the physical signals which shall be processed by the medium.

7.2 PhL resource

7.2.1 Signal frame

In AUTBUS network, the DLPDU sent by DLL are converted into Code Block (CB) that should be processed by the PhL through DLL-PhL interface. The CB should be encoded by Reed-Solomon (RS) coding and convolution coding, QAM mapping in PhL, then corresponds to the physical resource defined by OFDM symbol and subcarrier, and modulated by IFFT. Then the PhL signal frame (SF) is formed and transmitted to the cable. The receiver recovers the digital signal and stream through the reverse process.

The sampling interval of AUTBUS in time domain is T_s :

$$T_s = 1 / (24000 \times 4096)_s \quad (1)$$

That is 10,17 ns. The sampling interval is fixed in different TMs.

The basic unit of PhL signal in AUTOBUS is SF, and 64 OFDM symbols form one SF. Each OFDM symbol is divided into upper-side band and lower-side band in the frequency domain subcarrier. The high frequency subcarrier (16,896 ~ 32,256 MHz) is named upper-side band and the low frequency subcarrier (1,536 ~ 16,896 MHz) is named lower-side band. The upper-side band and lower-side band can be allocated to different MN or TNs when AUTOBUS allocates its channel resources to users.

Figure 8 shows the structure of the SF and OFDM symbol.

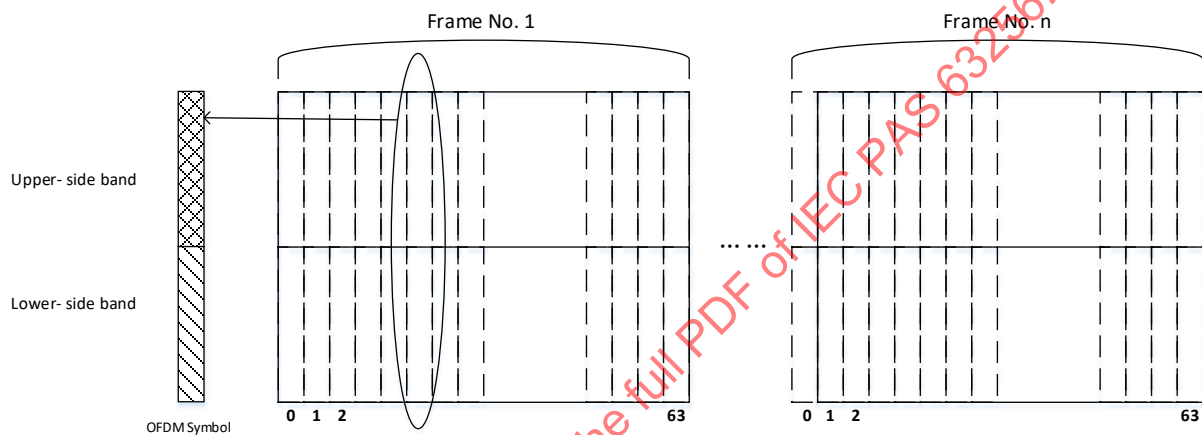


Figure 8 – AUTOBUS SF and OFDM symbol

Figure 9 shows the block diagram of the SF. The AUTOBUS signal frame should be decomposed by the following elements: Synchronization Signal (SynS), downlink subframe and data subframe, in which:

- **Synchronization Signal (SynS)** is used to identify the start of a SF. SynS has the function of SF synchronization and SF header positioning. The SynS is fixed to the lower-side band of the first and the second OFDM symbol of each SF (see Figure 9).
- **Downlink subframe** is used to send data and information from MN to TNs. It should include channel information, bandwidth allocation result, system maintenance status and other content. Downlink subframe contains the lower-side band of greater than or equal to 2 OFDM Symbols (see Figure 9).
- **Data subframe** is the allocable PhL communication resources. The minimum allocation unit is half-side band of one OFDM symbol (see 7.2.2 Carrier mode A) or half-side band of two OFDM symbols (see 7.2.2 Carrier mode B). In Figure 9, all blank parts are data subframe. The MN and TNs should request or config the corresponding data subframe according to their requirement for data transmission.

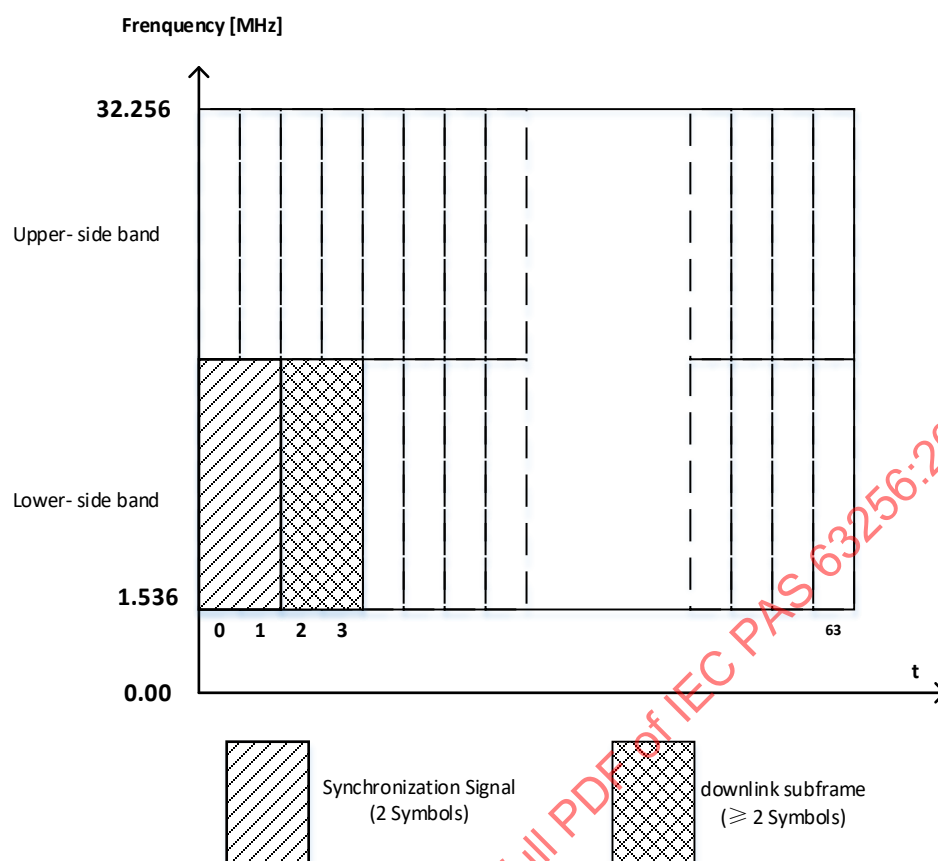


Figure 9 – SF structure

7.2.2 Data subframe

After AUTOBUS allocates the data subframe to the MN or each TN, in order to enable the receiver to decode the corresponding information quickly and accurately, pilot signals need to be inserted according to a specific form in the data subframe, this pilot is called symbol pilot, AUTOBUS supports discrete pilot and continuous pilot.

According to the difference of inserting symbol pilot mode and the data carrier structure of data subframe, the data subframe supports two kinds of carrier modes: carrier mode A and carrier mode B. They are defined as follows:

- **Carrier mode A** Using continuous pilot, a single OFDM half-side band should transmit both the data signal and the symbol pilot signal, in which the ratio of the symbol pilot signal to the number of subcarriers occupied by the data signal is 1:8.
- **Carrier mode B** Using discrete pilot, the first OFDM half-side band of the applied data subframe should transmit the symbol pilot signal, and the subsequent OFDM symbols should transmit the data signal.

Figure 10 shows the difference between carrier mode A and carrier mode B.

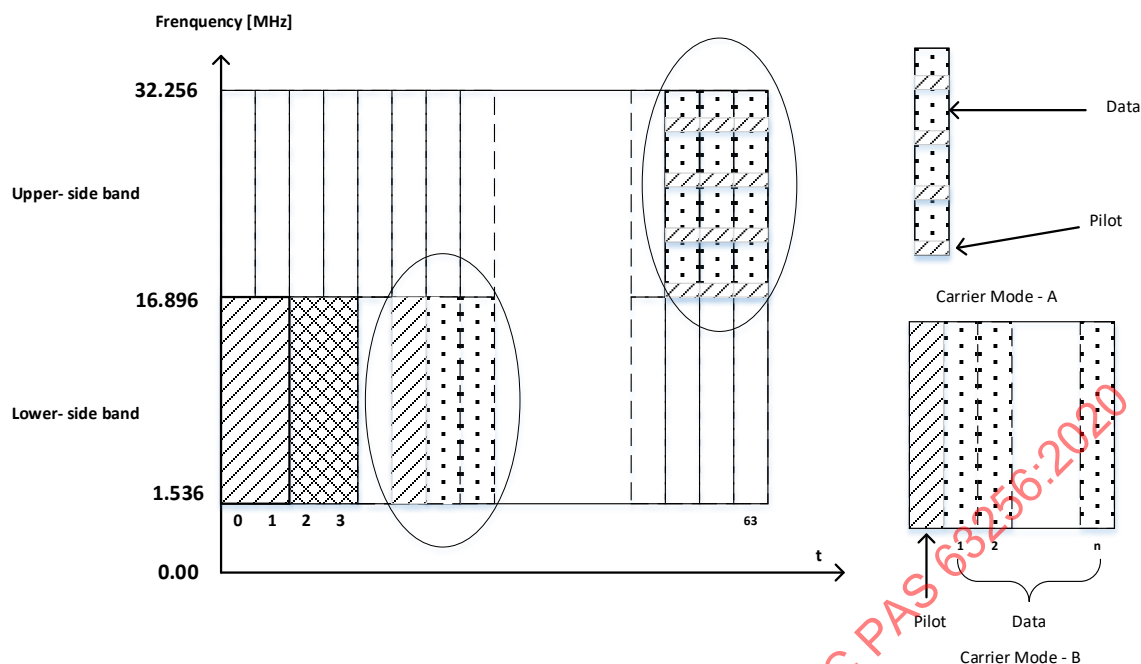


Figure 10 – Carrier mode A and carrier mode B of data subframe

Carrier mode A is generally used for periodic cyclic data transmission. AUTOBUS supports up to 8 users intervals if the carrier mode A is used. Carrier mode A also supports the application and use of equal allocation of interval resources for multi-users.

Carrier mode B is used both for cyclic fixed rate data transmission and variable rate data transmission.

NOTE Fixed rate data transmission means that the transmitted data is transmitted on the bus in a fixed cycle, and the data size transmitted per cycle is fixed. Variable rate data transmission means that the transmitted data is not processed on the bus in a fixed cycle, and the size of the data being processed is not fixed.

See Annex A for the distinction and choice of carrier mode A or carrier mode B.

7.2.3 OFDM symbol

Each OFDM symbol in the AUTOBUS PhL has the maintenance information as shown in Figure 11 to maintain the availability of OFDM symbols.

DV-H (2 bits)	DV-L (2 bits)	RVD (2 bits)	Symbol_Length (10-bits)	OFDM timing (64-bits)	Symb_Contents (14-812 octets)
------------------	------------------	-----------------	----------------------------	--------------------------	----------------------------------

Figure 11 – OFDM symbol structure of PhL

- **DV-H** is the availability of the upper-side band and carrier mode of the OFDM symbol, 0b00 is unable, 0b01 is carrier mode A, 0b10 is carrier mode B, 0b11 is reserved.
- **DV-L** is the availability of the lower-side band and carrier mode of the OFDM symbol, 0b00 is unable, 0b01 is carrier mode A, 0b10 is carrier mode B, 0b11 is reserved.
- **RVD** is reserved bits.
- **Symbol_Length** is the number of valid data octets loaded by this OFDM symbol.
- **OFDM_timing** is the time information contained in this OFDM symbol.
- **Symb_Contents** is the data loaded by this OFDM symbol and consists of 1 or 2 CBs (see 7.2.4).

Figure 12 shows OFDM timing structure. The structure has 64 bits and is the basis for AUTOBUS clock synchronization between nodes.



Figure 12 – OFDM timing structure

- **Frame Index** is the SF number of the current OFDM symbol, which counts from 0 when the MN is working normally. The valid value range is $0 \sim (2^{44} - 1)$. When the count equal to $(2^{44} - 1)$, it starts from 0 again.
- **Symbol Index** indicates the position of the current OFDM symbol in a SF. The valid value range is $0 \sim 63$ and start counting from received the SynS of a SF.
- **Cycle Index** is the current sampling subcarrier number. The valid value range is $0 \sim 4\,095$ and start counting from the start of the received OFDM symbol.

7.2.4 Code Block

The basic unit of data interaction between AUTOBUS PhL and DLL is Code Block (CB). The CB is composed of a certain number of octet sequences, which have the following characteristics:

- The number of octets of one CB is fixed corresponding to the different working modes of PhL (see 7.2.6). 1 ~ 2 CBs can be processed on each OFDM symbol half side-band.
- A DLPDU may be placed in one or more CBs after being serialized by octets. A CB may be store octet sequence information for more than one DLPDU. Figure 13 shows the relationship between DLPDU and CB.
- The PhL entity automatically adds CRC check codes with the length of four octets for each CB before starting to process the CB.
- If the CB contains invalid octet sequence, the value of octets are set to 0 by default.
- The MSB in the code block stores the octet sequence information corresponding to the MSB data of DLPDU, and the LSB stores the octet sequence information corresponding to the LSB data of DLPDU.

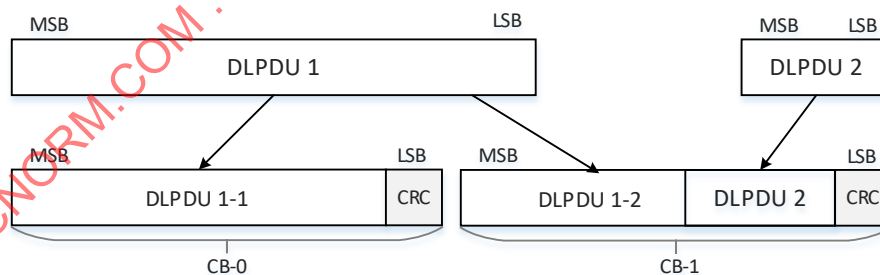


Figure 13 – DLPDU and CB

7.2.5 Resource element

The transmission signal of each SF is defined as the resource grid with N_{SD} subcarriers and N_{SYM} OFDM symbols, N_{SD} is determined by the transmit mode, the configuration value is shown in Table 18, and N_{SYM} is 64. Figure 14 shows resource grid structure, each element in the resource grid is called a resource element, and each resource element is uniquely defined by (k, l) , where $k = 0, \dots, N_{SD} - 1$ and $l = 0, \dots, N_{SYM} - 1$. k represents the frequency domain and l represents the time domain. Resource element (k, l) corresponds to complex value

$a_{k,l}$. The corresponding resource $a_{k,l}$ is set to 0 if it is not used for the transmission of physical signals.

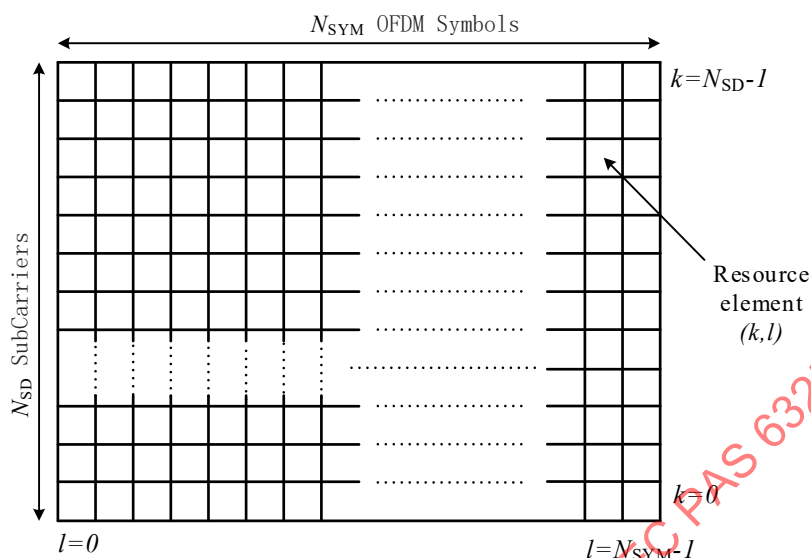


Figure 14 – Resource element schematic diagram

7.2.6 Working mode

The AUTOBUS PhL adopts OFDM technology and supports both time division multiplexing (TDM) and frequency division multiplexing (FDM). The AUTOBUS PhL provides four different Transmit Modes (TM 0 / 1 / 2 / 3) in frequency domain. AUTOBUS users can select the TM according to the actual signal-to-noise ratio (SNR) of the application environment. An OFDM symbol time length should be configured as 64 μ s, 32 μ s, 16 μ s or 8 μ s corresponding to different TMs, The SF period varies from 500 μ s to 4,0 ms. Table 18 shows the corresponding parameters in different TMs of the AUTOBUS PhL.

Table 18 – AUTOBUS TMs and corresponding parameters

Parameters	Transmit Mode			
	0	1	2	3
Frequency Range [MHz]	1,536 – 32,256	1,536 – 32,256	1,536 – 32,256	1,536 – 32,256
Lower-Side Band Frequency Range [MHz]	1,536 – 16,896	1,536 – 16,896	1,536 – 16,896	1,536 – 16,896
Upper-Side Band Frequency Range [MHz]	16,896 – 32,256	16,896 – 32,256	16,896 – 32,256	16,896 – 32,256
Frequency Bandwidth [Mhz]	30,72	30,72	30,72	30,72
Sub-carrier Spacing, B [KHz]	24	48	96	192
Number of Sub-carriers N_{SD}	1 280	640	320	160
FFT Size [K=1 024]	4K	2K	1K	0,5K
Sub-carrier modulation scheme	16QAM, 64QAM, 256QAM	16QAM, 64QAM, 256QAM	16QAM, 64QAM, 256QAM	16QAM, 64QAM, 256QAM

Parameters	Transmit Mode			
	0	1	2	3
Useful Symbol length [us]	41,666 7	20,833 3	10,416 7	5,208 3
Cyclic Prefix length [us]	20,833 3	10,416 7	5,208 3	2,604 2
Symbol length [us]	62,5	31,25	15,625	7,812 5
Sampling Interval [ns]	10,17	10,17	10,17	10,17
Frame length [ms]	4,0	2,0	1,0	0,5
OFDM Symbols per frame N_{SYM}	64	64	64	64
Available Channel per Symbol	1 or 2	1 or 2	1 or 2	1 or 2

AUTBUS PhL has multiple working modes to carry DLL data. Different working modes adapt to different channel environments and data transmission requirements. In each working mode, the allocation of physical resources and the efficiency of data carry are varied based on the selection of TM, frame format and modulation coding strategy (MCS).

After the configuration is complete, the MN should broadcast the working mode to each TN of the whole network. If the network configuration is successful, the AUTBUS network should work according to this mode and implement the communication between nodes until the MN reconfigures the working mode next time.

The CB is the basic unit of data processing in the PhL. The valid octet number carried by an OFDM symbol is an integer multiple of the number of valid octets of a CB. AUTBUS supports a total of 22 working modes, of which carrier mode A supports 10 working modes, carrier mode B supports 12 working modes. The parameter configuration and the equivalent available bandwidth in each mode are shown in Table 19 and Table 20.

Table 19 – Working mode under carrier mode A

Working mode	Transmit mode	FFT Points	Valid data in CB (octets)	Valid data of a OFDM symbol (octets)	Valid N_{SD} of OFDM symbol	Equivalent available bandwidth (Mbps)
A0	0	1 024	173	346	1 280	44,288
A1	0	1 024	103	412	1 280	52,736
A2	0	1 024	173	692	1 280	88,576
A3	1	512	84	168	640	43,008
A4	1	512	103	206	640	52,736
A5	1	512	173	346	640	88,576
A6	2	256	57	114	320	58,368
A7	2	256	84	168	320	86,016
A8	3	128	22	44	160	45,056
A9	3	128	39	78	160	79,872

Table 20 – Working mode under carrier mode B

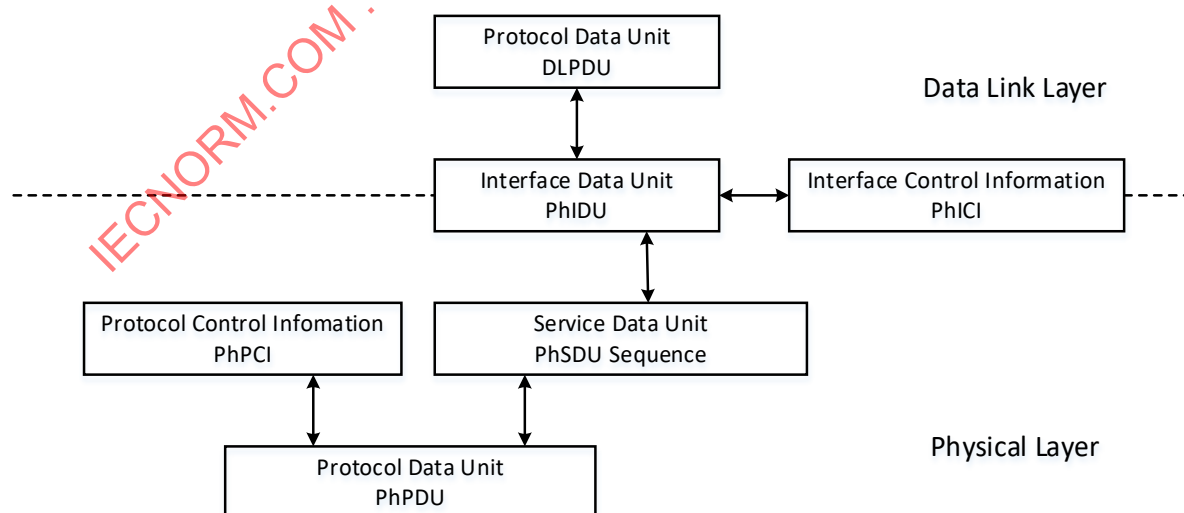
Working mode	Transmit mode	FFT Points	Valid data in CB (octets)	Valid data of a OFDM symbol (octets)	Valid N_{SD} of OFDM symbol	Equivalent available bandwidth (Mbps)
B0	0	1 024	123	246	1 280	31,488
B1	0	1 024	203	406	1 280	51,968
B2	0	1 024	123	492	1 280	62,976
B3	0	1 024	203	812	1 280	103,936
B4	1	512	59	118	640	30,208
B5	1	512	99	198	640	50,688
B6	1	512	123	246	640	62,976
B7	1	512	203	406	640	103,936
B8	2	256	59	118	320	60,416
B9	2	256	99	198	320	101,376
B10	3	128	27	54	160	55,296
B11	3	128	47	94	160	96,256

7.3 DLL–PhL interface

7.3.1 General

Subclause 7.3 defines the required Physical Service (PhS) primitive and constraints on their use.

PhIDU shall be transferred between the DLL and the PhL in accordance with the requirements of ISO/IEC 7498-1 as shown in Figure 15.

**Figure 15 – Mapping between data units across the DLL-PhL interface**

These services provide for the interchange of information data units between a DLL entity and its associated PhL entity. They also provide methods for data interaction between data entities,

allows cooperative work DLL entities to coordinate transmission and data exchange on shared medium, and ensures the deterministic and synchronous of data exchange.

7.3.2 Service primitive

7.3.2.1 Ph-Param (para, value)

The PhL informs the current working mode, coding and modulation related parameters to DLL. The DLL shall implement the encapsulation and processing of the PhLSDU according to the parameters. Table 21 shows the primitives and parameters of the Ph-Param service. The primitive parameters are described in Table 22.

Table 21 – Ph-Param service primitives and parameters

Parameter name	Indication
para	M
value	M

Table 22 – Ph-Param service primitive parameter description

para		value
para index	para description	
0	Transmit_mode	0 ~ 3, see Table 18, current system transmitting mode
1	Work_mode	0 ~ 21, see Table 19 and Table 20, current working mode parameter 0 ~ 9 corresponding to working mode A0 ~ A9, 10 ~ 21 corresponding to working mode B0 ~ B11
2	Rs_code	0 ~ 11, see Table 34, RS code mode
3	Con_code	0 ~ 2, see Table 35, Convolutional code mode
4	Qam_mode	0 ~ 2, respectively corresponding to three mapping mode of 16QAM, 64QAM and 64QAM

7.3.2.2 Ph-Data (length, data, status)

AUTBUS PhL provides the ability of the DLL to transmit and receive data based on the service. Table 23 shows the service primitives and parameters. Primitive parameter length is the octet length, data is DLPDU and status is the processing result of the corresponding service primitive. The primitive parameters are described in Table 24.

Table 23 – Ph-Data service primitives and parameters

Parameter name	Request	Indication	Confirm
length	M	M	M
data	M	M	—
status	—	M	M

Table 24 – Ph-Data service primitive parameter description

Parameter name	Description
length	Specify the octet length of data
data	Transmitting or receiving DLPDU, data is invalid when the flag is the transmitting confirmation
status	This parameter is a 16-bit length value Status information indicated by bit bit 0 indicates transmitting or receiving status bit 1 indicates data status bit 2 indicates CRC check status The corresponding bit set to 0 indicates that the status is correct, and set to 1 indicates status error bit 3 ~ bit 15 reserved

The DLL Data Entity (DLDE) requests the PhL entity to transmit the data based on the service request primitive. When the PhL receives the DLL Ph-Data request primitive, the PhLDE shall convert DLPDU to PhSDU according to the PhICI information and add the required PhPCI information, and the PhPDU in the form of physical signal shall be transmitted to the physical medium by encoding and modulation. The transmitting result is notified to the DLL through the Ph-Data.Confirm primitive.

The PhL Data Entity (PhLDE) informs the DLDE to receive data based on the service indication primitive. When the PhL receives the PhPDU in the form of physical signal through the medium, after demodulation and decoding, the PhLDE should resolve the PhSDU transmitted by the peer PhLDE and convert them into DLPDU according to the PhICI information, and transmit to DLL through the Ph-Data.Indication primitive.

7.3.2.3 Ph-Clock-Sync (command, data, ofdmtiming)

The AUTOBUS PhL provides clock synchronization service to the DLL. The DLL should implement the function of clock synchronization and delay measurement based on this service. Table 25 shows the service primitives and parameters. The primitive parameters are described in Table 26.

Table 25 – Ph-Clock-Sync service primitives and parameters

Parameter name	Request	Indication
command	M	M
data	M	M
ofdmtiming	-	M

Table 26 – Ph-Clock-Sync service primitive parameter description

Parameter name	Description
command	DLL requests PhL clock synchronization command 0 Clock synchronization 1 Delay measurement
data	Clock synchronous DLPDU, fixed length
ofdmtiming	PhL OFDM timing information Only the indication primitive is valid

When the DLL requests the PhL to transmit clock synchronization information (command=0 or 1) through the Ph-Clock-Sync.Request primitive, the PhLDE should fill the current OFDM timing information into the clock information field (see 8.3.2.12) of the clock synchronization DLPDU, convert the clock synchronization DLPDU to PhSDU according to the PhICI information and add the required PhPCI information. The PhLDE should encode and modulate the PhPDU to form physical signals, and then transmit them to the physical medium. When the PhL receives a PhPDU with clock information in the form of a physical signal through the medium, after demodulation and decoding, the PhLDE should resolve the PhSDU transmitted by the peer entity and convert it into a clock synchronous DLPDU according to the PhICI information. PhLDE should transmit the PhSDU to DLL through Ph-Clock-Sync.Indication primitive.

When the PhLDE receives the Ph-Clock-Sync.Request (Command=1) primitive, the PhLDE should notify the PhL Management Entity (PhLME) to perform delay measurement processing in addition to transmitting clock synchronization DLPDUs (see 7.4.2.5).

7.3.3 Primitive process

Figure 16 shows DLL-PhL interface service primitive process.

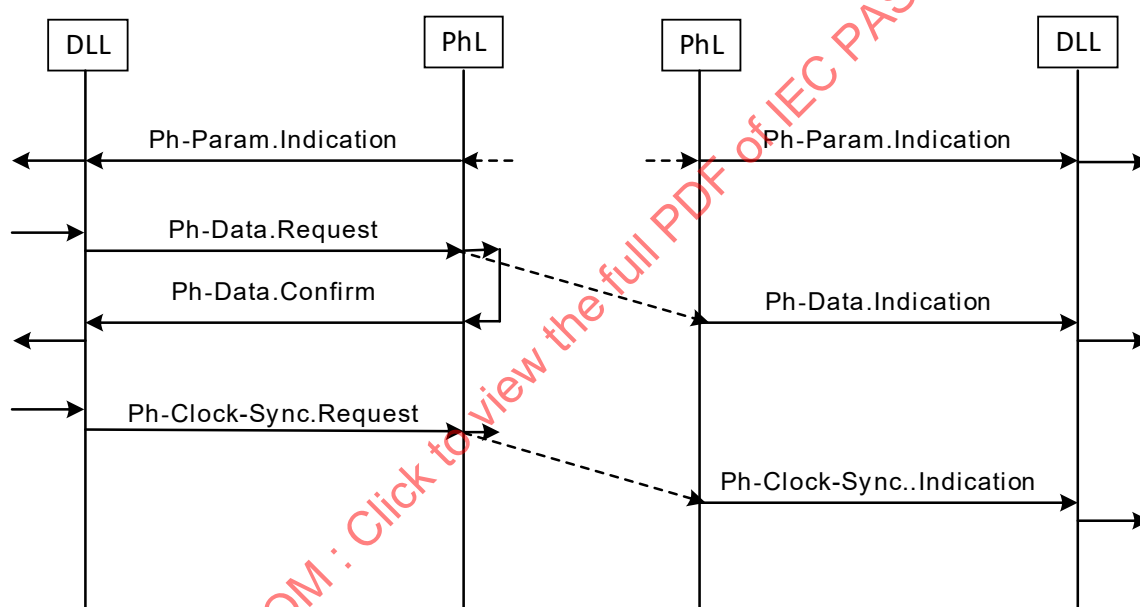


Figure 16 – DLL-PhL interface service primitive process

7.4 System management-PhL interface

7.4.1 General

This interface provides services to the PhL, which are required for initialization and selection of options.

The system management user implements the processing of the related content of the PhLME based on this interface. The service interface parameter parameter_index should be consistent with the actual content of the PhLME. See 6.3.1 for related information about PhL entity and system management information. Subclause 7.4 does not require this interface to be exposed. Each device in the network can use a subset of the described primitives in this document.

7.4.2 Service primitive

7.4.2.1 Ph-RESET (flag)

The PhL system management user informs the PhL to reset based on the service primitive. Table 27 shows the service primitives and parameters.

Table 27 – Ph-RESET primitives and parameters

	Request	Confirm
flag	M	—

The PhL confirms whether it needs to be reset through the parameter flag, as described in Table 28. For reset operations with flag = 1, system management users should receive confirmation information.

Table 28 – Ph-RESET service primitive parameter description

Parameter name	Parameter value
flag	Reset flag: 0: Do not reset the PhL 1: Reset all functions of the PhL and the PhL entity parameters should be initialized to the default values.

7.4.2.2 Ph-SET-VALUE (parameter_index, value, status)

The PhL system management user uses the service to set the related parameters of PhLME. Table 29 shows the service primitives and parameters. See 6.3.1 for the specific description of the parameters parameter_index and value. The parameter status identifies the confirmation status of the requested service. Table 30 shows the service primitive parameter status.

Table 29 – Ph-SET-VALUE primitives and parameters

	Request	Confirm
parameter_index	M	—
value	M	—
status	—	M

Table 30 – Ph-SET-VALUE primitive parameter status description

Parameter name	Parameter value
status	Service returns status value 0: success 1: failure

7.4.2.3 Ph-GET-VALUE (parameter_index, value, status)

The PhL system management user uses the service to get the related parameters of the PhLME. Table 31 shows the service primitives and parameters. See 6.3.1 for the parameter parameter_index description. The parameter status as shown in Table 30 indicates the confirmation status of the corresponding request service.

Table 31 – Ph-GET-VALUE service primitives and parameters

	Request	Confirm
parameter_index	M	—
value	—	M
status	—	M

7.4.2.4 Ph-EVENT (parameter_index, value)

The PhLME should notify the system management user based on the service when it detects an event. The service indication primitive parameter should clarify the relevant information at the time of the event. Table 32 shows the service primitive and parameters.

Table 32 – Ph-EVENT service primitive and parameters

Parameter name	Indication
parameter_idx	M
value	M

7.4.2.5 Ph-SYNC (command, parameter_idx, value, status)

The PhL system management user uses the service to process the synchronization related parameters of the PhLME (see 6.3.1.4). Table 33 shows the Ph-SYNC service primitives and parameters.

The parameter command indicates the specific instruction of the primitive processing. When the parameter command is 0, it indicates that the PhL system management user should read the corresponding value of the parameter_idx. When the parameter command is 1, it indicates that the corresponding content of the parameter_idx in the PhLME should be set with value. The parameter status indicates the execution result of the corresponding primitive, the value of 0 indicates the primitive process is success, and the value of 1 indicates that the primitive process is failure. Table 33 shows the service primitives and parameters. See 6.3.1.4 for the parameter parameter_index description.

Table 33 – Ph-SYNC service primitives and parameters

Parameter name	Request	Indication	Comfirm
command	M	M	—
parameter_idx	M	M	M
value	—	M	M
status	—	—	M

7.4.3 Primitive process

Figure 17 shows system management -PhL interface service primitive process.

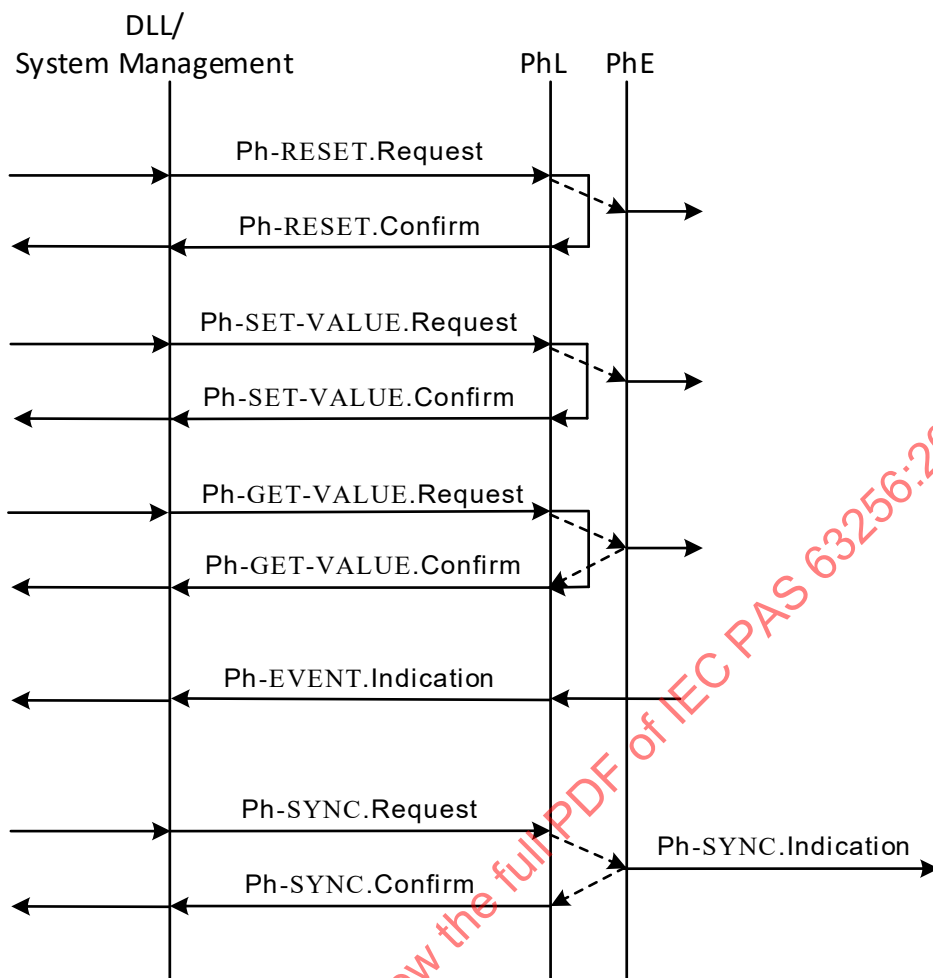


Figure 17 – System management-PhL interface primitive process

7.5 DCE Independent Sublayer (DIS)

7.5.1 General

The AUTBUS PhL entity is partitioned into a DTE component and a DCE component. The DTE component interface and the DLL entity form the DCE Independent Sublayer (DIS). It exchanges Interface Data Units (IDU) across the DLL-PhL defined in 7.3, and provides the basic conversions between the PhIDU of the DLL-PhL interface and the basic data unit required for physical transmission and receiving.

This sublayer is independent of all the PhL variations, including encoding, modulation, mapping, electrical characteristics, medium etc. All these variations are grouped under the designation Data Communication Equipment (DCE).

7.5.2 Data sequence

The DIS of the AUTBUS PhL should sequence the PhID transmission as a PhSDU sequence, the reverse is similar, DIS needs to convert the received PhSDU sequence to PhID, and transmit it to DLL.

The PhSDU sequence of the AUTBUS PhL is composed of CBs, PhID should be converted into PhSDU for transmission in octets. The most significant octets should be transmitted before or at the same time as less significant octets. On reception, the most significant octets of PhSDU should be received before or at the same time as less significant octets. In the absence of

errors, the PhIDU indicated to the receiving DLL entity shall be unchanged from the PhIDU whose transmission was requested by the originating DLL entity.

7.6 DTE-DCE interface

7.6.1 General

The AUTOBUS PhL entity is partitioned into a Data Terminal Equipment (DTE) component containing the DIS, and a Data Communication Equipment (DCE) component containing the MDS and lower sublayers. The DTE-DCE interface connects these two physical components, and is itself within the MIS. (See Figure 7)

It is not mandatory for the DTE-DCE interface, or any other interface, to be exposed.

When the data is serialized in the DIS, the DCE should be informed to transmit data through this interface. After the DCE completes the data signal receiving and processing, the DTE shall be notified to perform data serialization processing. The AUTOBUS PhL DTE-DCE interface realizes the services through a set of defined interface signal interactions.

7.6.2 Interface signal

7.6.2.1 Synchronization Request (SyR)

The SyR signal is a continuous timing signal provided by the DCE to the DTE. The basic unit of the timing signal is an OFDM symbol period. The DCE informs the DTE to implement SF synchronization and to continue processing the physical SF based on the SyR signal. AUTOBUS specifies that the synchronization request signal is fixed to 2 symbol period, and every 64 OFDM symbol period process a SF synchronization.

7.6.2.2 Synchronization Confirm (SyC)

DTE transmit SyC signal to DCE to implement synchronization confirm, at the same time, DTE should transmit synchronization information to DLL entity and upper user for the next step processing.

7.6.2.3 Transmit Request (TxR)

DCE starts to prepare for data transmission according to the SyC signal transmitted by DTE. DCE should be ready to receive TxR signal of DTE after the relevant parameters of transmitting are confirmed. If there is an error in transmitted data, the DTE should notify through this signal, the DTE of AUTOBUS PhL do not confirm whether this signal has been processed by DCE.

7.6.2.4 Transmit Data (TxD)

DTE transmit TxR signal to notify DCE to transmit data. The DCE performs scrambling, encoding, interleaving, modulating, and mapping processing on the CB to generate a physical signal based on the determined CB and the determined parameter information, and transmits out the physical signal over medium.

7.6.2.5 Receive Request (RxR)

DTE starts to prepare for data receiving according to the RxR signal transmitted by DCE. DTE should be ready to receive RxR signal of DCE after the relevant parameters of receiving are confirmed. If there is an error in received data, the DCE should notify DTE through this signal, the DCE of AUTOBUS PhL do not confirm whether this signal has been processed by DTE.

7.6.2.6 Receive data (RxD)

DCE transmit RxD signal to notify DTE to receive data. The DTE receives the signal, and obtains the valid data from the determined CB, implements the conversion from the serialized data to the interface data unit, and informs the DLL entity for further processing.

7.6.3 Signal process

The trigger time of DTE-DCE interface signal in AUTOBUS PhL is confirmed according to the configuration of bus cycle time related parameters. The basic unit of the interface signal trigger period is an OFDM symbol period. Figure 18 shows the process of DTE-DCE interface signal on the device.

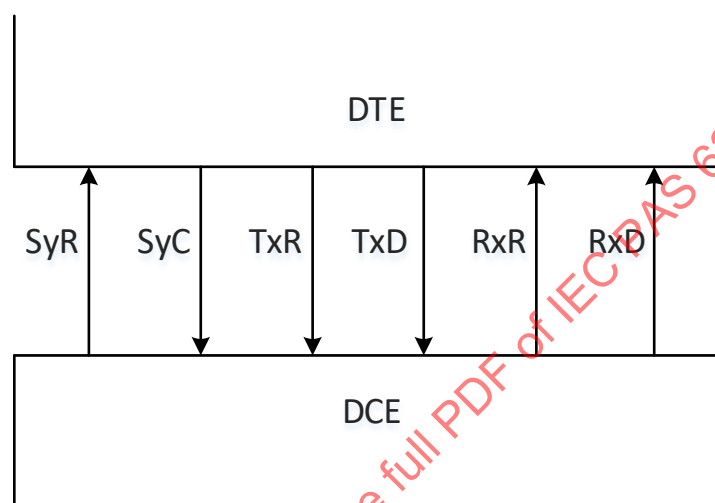


Figure 18 – DTE-DCE interface signal process

7.7 Medium Dependent Sublayer (MDS)

7.7.1 General

The AUTOBUS PhL MDS is a part of the DCE (see Figure 7). It exchanges PhSDU data through the DTE-DCE interface specified in 7.6 and it transmits the mapped symbols to the MAU through the MDS-MAU interface. The peer MDS should demodulate and decode the signal received through the MDS-MAU interface. The PhSDU should be recovered and be transmitted to DIS based on the DTE-DCE interface for further processing.

The MDS of AUTOBUS PhL specifies the physical channel process of DCE, including scrambling, encoding, decoding, interleaving, mapping, modulation and demodulation.

7.7.2 MDS specification

7.7.2.1 Physical channel process

Figure 19 shows the AUTOBUS PhL channel process of transmission and receiving. In the transmission process, data streams are interleaved after scrambling, RS encoding and convolution encoding respectively, then modulated by 16QAM, 64QAM or 256QAM (see 7.7.2.5) to form an OFDM signal. According to different TMs, the data is combined into SFs and then transmitted out on the cable line. The receiving system is the inverse process of the transmission system which is used to recover the data flow correctly from the encoded physical signal.

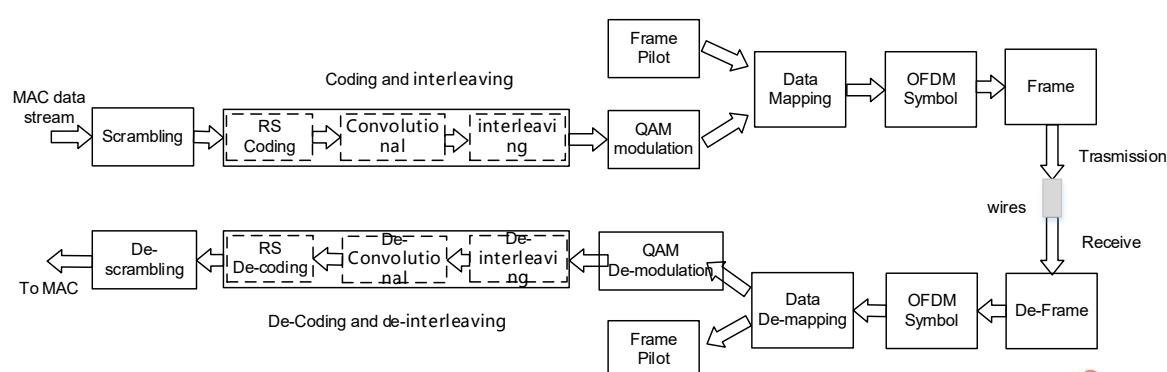


Figure 19 – PhL channel process

7.7.2.2 System parameter configuration

The PhL signal is transmitted by OFDM modulation. The core process of generating OFDM signal is IFFT (Inverse Fast Fourier Transform), and the OFDM signal processing corresponding to FFT (Fast Fourier transform). Under the given the system bandwidth and sampling interval condition, different FFT/IFFT sampling points should get different sub-carrier interval and OFDM symbol length. The AUTOBUS system supports 0, 1, 2 and 3, in total four different TMs, corresponding to 4 096, 2 048, 1 024 and 512 sampling points respectively. For specific parameter configuration, see Table 18.

7.7.2.3 Scrambling

In order to avoid the long sequences consisting of "0" or "1" only, the CB to be transmitted should be scrambled before encoding. The scrambling sequence is a m sequence with length 11, the generating polynomial is

$$x^{11} + x^2 + 1 \tag{2}$$

The initialization phase is 0b1111111111. The scrambling sequence generator is reinitialized as each CB begins processing. Figure 20 shows the scrambling sequence generation process, the “enable” indicates the control switch, “data” is the input octet sequence to be scrambled, “output” is the scrambled octet sequence.

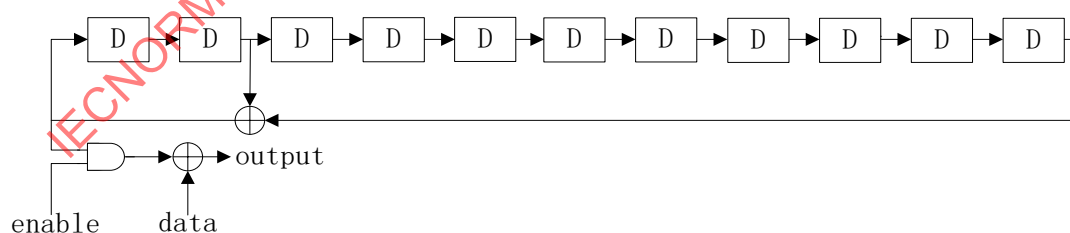


Figure 20 – Scrambling sequence generation

7.7.2.4 Encoding and interleaving

PhL encoding adopts concatenated code that the inner code is the convolutional codes and the outer code is the RS codes. The transmission block checked by CRC first enters the outer code encoder for RS encoding, and then enters the inner code encoder for convolutional code encoding. The encoded bits enter the bit Interleaver for interleaving processing.

7.7.2.4.1 RS coding

RS code uses the shorter code of original RS(255,247), RS(255,239), RS(255,223) system code. Original RS code length is 255 octets. The check length is 8, 16 or 32 octets, which is the cyclic code based on finite field GF(256), the generating polynomial of the domain is

$$p(x) = x^8 + x^4 + x^3 + x^2 + 1 \quad (3)$$

When the RS code generating polynomial is

$$g(x) = \sum_{i=0}^{L-1} g_i x^i \quad (4)$$

L is Check bit length, the input information sequence polynomial is

$$m(x) = \sum_{i=0}^{K-1} m_i x^i \quad (5)$$

K is information bit length, the system code output polynomial is

$$C(x) = \sum_{i=0}^{N-1} c_i x^i = x^{N-K} m(x) + r(x) \quad (6)$$

Where

$$r(x) = \frac{x^{N-K} m(x)}{g(x)} \quad (7)$$

Here the RS(255,247), RS(255,239), RS(255,223) generate polynomial $g^0(x)$, $g^1(x)$, $g^2(x)$, see Annex B.

The RS code mode used is shown in Table 34

Table 34 – RS code mode

RS code mode	RS code
0	(239,207)
1	(159,127)
2	(119,103)
3	(79,63)
4	(59,51)
5	(39,31)
6	(109,177)
7	(139,107)
8	(104,88)
9	(69,61)
10	(51,43)
11	(34,26)

7.7.2.4.2 Convolutional encoding

The transmission data encoded by RS map each octet to 8-bit stream according to the principle of high bit priority transmission, and send it into convolutional code encoder, the bit rates respectively are 1/2, 2/3 and 3/4, as shown in Table 35.

Table 35 – Convolutional code mode

Convolutional code	Convolutional code rate
0	1/2
1	2/3
2	3/4

The convolutional encoder with constraint length of 7 and a code rate of 1/2, generation polynomial is $G_1 = 171_8$ and $G_0 = 133_8$. Figure 21 shows a convolutional encoder with a code rate of 1/2.

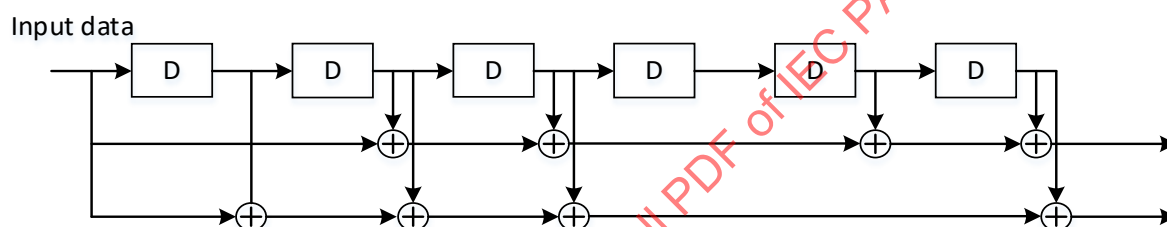


Figure 21 – Convolutional encoder with a code rate of 1/2

The initial value of the shift registers of the encoder set to the corresponding value of the last 6 bits in the input stream, so that the initial and final states of the shift register are the same. According to Figure 21, the coded bit stream output in A and B order. By deleting the encoding bits, the convolutional code with the code rates 2/3 and 3/4 shall be got. The process of deleting bit is shown in Figure 22.

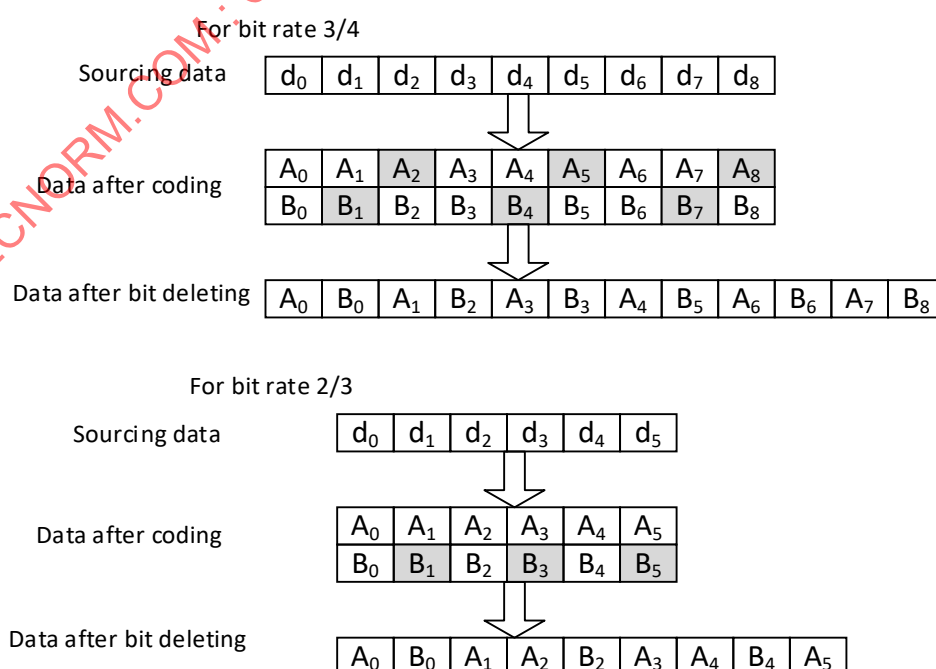


Figure 22 – Bit deletion process with code rates 2/3 and 3/4

7.7.2.4.3 Bit interleaving

The coded bit stream should be interleaved. Grouping a bit stream with N_{CBIB} as an interleaving block. All coded data bits are interleaved in the resources block of an OFDM symbol. Interleaving is partitioned into two steps: the first permutation ensures that adjacent coding bits are mapped to non-adjacent subcarriers, and the second permutation ensures that adjacent coding bits are alternately mapped to high and low significant bits of the constellation. The specific interleaving rules are:

The first permutation:

$$i = (N_{CBIB} / N_{COL}) (k \bmod N_{COL}) + \text{floor}(k / N_{COL}) \quad k = 0, 1, \dots, N_{CBIB} \quad (8)$$

Where N_{CBIB} is Interleaved block length, k is the coded bit sequence number before the first permutation, i is the coded bit sequence number after the first permutation.

The second permutation:

$$j = s * \text{floor}(i/s) + (i + N_{CBIB} - \text{floor}(i * N_{COL} / N_{CBIB})) \bmod s \quad (9)$$

$$i = 0, 1, \dots, N_{CBIB} \quad s = \max(N_{BPS}/2, 1)$$

Here j is the coded bit sequence number after the second permutation. N_{BPS} is the number of bits corresponding to a modulation symbol. Table 36 shows the bit interleaving parameters. Carrier mode A should use interleaving modes 3, 4. Carrier mode B should use interleaving modes 0, 1, 2.

Table 36 – Bit interleaving parameters

Interleaving mode	Modulation mode	Symbol bit number (N_{BPS})	Column number (N_{COL})	Interleaving block length (N_{CBIB})
0	16QAM	4	16	320
1	64QAM	6	16	480
2	256QAM	8	16	640
3	64QAM	6	14	420
4	256QAM	8	14	560

7.7.2.5 QAM mapping

The modulation mapper uses the binary number 0, 1 as input to generate complex modulation output symbol $\tilde{x}$. The modulation mapping scheme is 16QAM, 64QAM and 256QAM. The binary number obtains the constellation point Z according to the modulation mode, then normalizes to obtain the correct complex modulation symbol $\tilde{x}$.

7.7.2.6 Resource mapping

The reference signal is generated by m sequence with a generating polynomial of $x^{11} + x^8 + x^5 + x^2 + 1$. Figure 23 shows the generation circuit of m sequence.

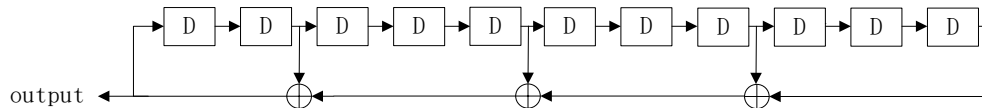


Figure 23 – Generation of m sequence

The m sequence in Figure 23 is modulated by BPSK and gets the reference signal generation sequence $r(m)$. For the pilot physical resource mapping, the reference signal $r(m)$ should be mapped to OFDM symbol resource. The SynS pilot is mapped to the 0th, 1st OFDM symbol of the SF and for the pilot symbol 0, only the even subcarriers. The scattered pilot should be mapped to the upper-side band or lower-side band of one OFDM symbol, and the contiguous pilot should be mapped to a partial subcarrier in the specified OFDM symbol half-side band. For the physical resource mapping of the data, the modulated complex symbol block should be mapped to the corresponding OFDM symbol half-side band resource, and the corresponding subcarrier resource equal interval distribution is implemented in the frequency domain.

7.7.2.7 OFDM signal generation

The time continuous signal of the l OFDM symbol of a SF is defined by the following formula.

$$s_l(t) = \left[\sum_{k=N_S}^{N_S+N_{SD}-1} a_{k^{(-)},l} \cdot e^{j2\pi k\Delta f(t-N_{CP,l}T_s)} \right] \quad (10)$$

where

$$k^{(-)} = k - N_S \quad (11)$$

Table 37 shows OFDM configuration parameters.

Table 37 – OFDM configuration parameters

Transmit mode	Δf Subcarrier interval, kHz	N FFT number	N_{SD} Valid subcarrier number	N_{CP} Cyclic prefix length	N_S Starting subcarrier number
0	24	4 096	1 280	2 048	64
1	48	2 048	640	1 024	32
2	96	1 024	320	512	16
3	192	512	160	256	8

The transmitted function is obtained from the take real function of the above formula

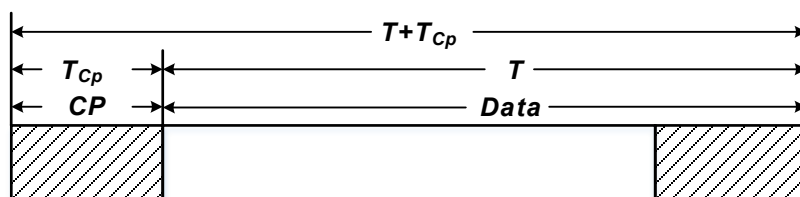
$$S_l(t) = \text{Re}[s_l(t)] \quad (12)$$

7.7.2.8 Framing

For each symbol in the SF, there is a structure shown in Figure 24, and 64 symbols form a SF.

Expression is

$$S(t) = \sum_{l=0}^{63} S_l(t - T_l) \quad (13)$$



CP : Cyclic Prefix **Data** : Data **T_{CP}**: Cyclic Prefix Time **T** : Data Time

Figure 24 – OFDM symbol structure diagram

7.7.2.9 Modulation Coding Scheme (MCS)

AUTBUS PhL supports a variety of encoding modulation modes, and the MCS under carrier mode A and carrier mode B is shown in Table 38 and Table 39.

Table 38 – Modulation Coding Scheme in carrier mode A

Working mode	Transmit mode	RS coding	Convolutional encoding	modulation mode
A0	0	(209,177)	1/2	64QAM
A1	0	(139,107)	2/3	64QAM
A2	0	(209,177)	3/4	256QAM
A3	1	(104,88)	1/2	64QAM
A4	1	(139,107)	2/3	64QAM
A5	1	(209,177)	3/4	256QAM
A6	2	(69,61)	2/3	64QAM
A7	2	(104,88)	3/4	256QAM
A8	3	(34,26)	2/3	64QAM
A9	3	(51,43)	3/4	256QAM

Table 39 – Modulation Coding Scheme in carrier mode B

Working mode	Transmit mode	RS coding	Convolutional encoding	Modulation mode
B0	0	(159,127)	1/2	16QAM
B1	0	(239,207)	1/2	64QAM
B2	0	(159,127)	2/3	64QAM
B3	0	(239,207)	3/4	256QAM
B4	1	(79,63)	1/2	16QAM
B5	1	(119,103)	1/2	64QAM
B6	1	(159,127)	2/3	64QAM
B7	1	(239,207)	3/4	256QAM

Working mode	Transmit mode	RS coding	Convolutional encoding	Modulation mode
B10	2	(79,63)	2/3	64QAM
B11	2	(119,103)	3/4	256QAM
B14	3	(39,31)	2/3	64QAM
B15	3	(59,51)	3/4	256QAM

7.8 MDS-MAU interface

7.8.1 General

The Medium Attachment Unit (MAU) is an optionally separate part of a communication element that connects to the medium directly or via passive components (see Figure 7). The MDS-MAU interface of AUTOBUS PhL provides the function of signal conversion and docking. The interface should convert the digital signal output by MDS into analog signal to be sent by MAU, or convert the analog signal received from MAU into digital signal that can be processed by MDS.

7.8.2 MDS-MAU interface

7.8.2.1 Services

AUTOBUS PhL MDS-MAU interface shall support at least the set of required services given in Table 40 and specified in 7.8.2.3.

Table 40 – Minimum services at MDS-MAU interface

Service	Abbreviation	Direction
Required:		
Transmit signal	TxS	To MAU
Receive signal	RxS	From MAU
Transmit enable	TxE	To MAU
Optional:		
Loopback enable	LbE	To MAU

7.8.2.2 Service process

Figure 25 shows AUTOBUS PhL MDS-MAU interface service process.

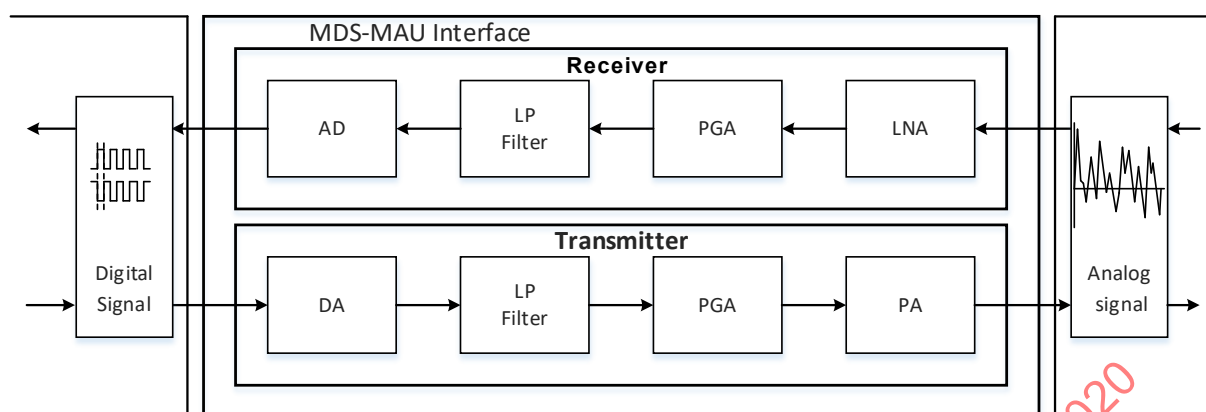


Figure 25 – MDS-MAU interface service process diagram

AUTBUS PhL MDS implements the encoding and modulation based on OFDM, outputs the digital signal to the MDS-MAU interface, and continues to transmit the signal. The interface includes the transmitter and the receiver. In the transmitter the digital signal should be converted to an analog signal by the D/A module, and after filtering and gain amplification, power amplification, the analog signal should be transmitted to the medium and be sent to the network. In the receiver the analog signal should be low noise amplified, power amplified and filtered, and then be recovered to the corresponding digital signal by A/D module, and the digital signal should be transmitted to MDS for demodulation and decoding and other signal processing.

7.8.2.3 Service specifications

7.8.2.3.1 Transmit Signal (TxS)

The Transmit Signal service (TxS) shall transfer the encoded PhPDU signal sequence across the MDS-MAU interface to the MAU. If the Transmit Enable (TxE) is set to logic 0 (low level), the output analog signal sequence shall be transmitted on to the medium.

7.8.2.3.2 Receive Signal (RxS)

The Receive Signal (RxS) shall transfer the encoded PhPDU signal sequence across the MDS-MAU interface to MDS. The output signal sequence should be a decodable PhPDU. Once the AUTBUS PhL runs, the RxS service is always valid by default and shall not be confirmed to enable. The RxS shall echo the signal transmitted via TxS by simultaneously receiving the transmissions from the medium.

7.8.2.3.3 Transmit Enable (TxE)

The Transmit Enable service (TxE) shall provide the MDS with the facility to enable the MAU to transmit. The TxE shall be set to logic 0 (low level) at the commencement of preparing the sequence of PhPDU signals to be sent and then set to logic 1 (high level) after the MDS-MAU interface processed the sequence and transmitted it on to the medium. Once the AUTBUS physical device is powered on, the TxE is set to logic 0 (low level) by default, which always allows the user to send.

7.8.2.3.4 Loopback enable (LbE)

When the MDS-MAU interface provides Loopback Enable (LbE) service, the signal sent by the MAU to the medium should be loopback to the MAU as the received signal to continue to process. LbE causes all signals sent by the MAU to be local loopback, which has no effect on the signals received by the MAU from the medium. The service status of LbE shall not change when the MAU is transmitting or receiving.

NOTE This confirmation service is of local significance only and provides a device with the facility to test the integrity and functionality of the PhL circuit, excluding the medium.

7.8.2.4 Transmit specification

7.8.2.4.1 Transmit level

The AUTBUS PhL specifies the maximum allowable output power which is 100 mW.

7.8.2.4.2 Transmit frequency spectrum specification

The transmission frequency range implemented in the AUTBUS PhL MDS-MAU interface is 1,536 MHz ~ 32,256 MHz. In order to reduce the out-of-band power of RF signal, the filter should be used to filter the RF signal. Figure 26 shows the signal template for a possible filter implementation under signal bandwidth.

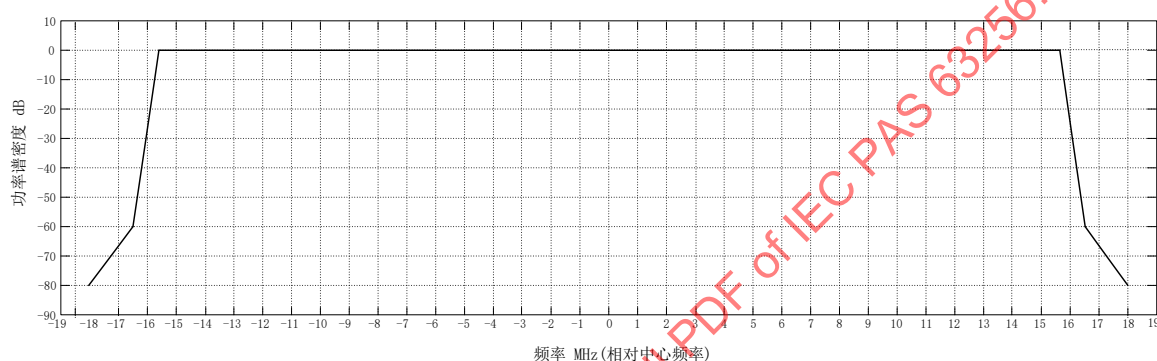


Figure 26 – Signal spectrum template

7.8.2.4.3 Transmit subcarrier frequency tolerance

The clock frequency range of AUTBUS PhL is required to be 100 MHz (± 150 ppm). During transmitting, the subcarrier frequency is synchronized with the clock frequency.

7.8.2.4.4 Modulation accuracy

7.8.2.4.4.1 Transmitter constellation diagram

The constellation diagram after averaging subcarrier, OFDM frame and grouping for RMS should comply with Table 41.

Table 41 – Allowable constellation diagram errors in different modulation modes

Modulation mode	Relative constellation error (dB)
16QAM	45
64QAM	45
256QAM	45

7.8.2.4.5 System transmission parameters

The maximum transmit power of the system is 20 dBmW, the receiving sensitivity is -82 dBmW.

Maximum transmission distance: 500 m.

Table 42 shows the parameters specifications in different transmission rates.

Table 42 – System transmission parameters

Parameter	Value			Unit
Net load bit rate	14	57	103	Mbps
SNR	13,4	18	23	dB

7.9 Medium Attachment Unit (MAU)

7.9.1 Overview

The AUTBUS PhL MAU provides access to the communication network, and the devices connected to the network communicate over the medium. AUTBUS network communication medium is usually composed of a pair of twisted-pair wires or a coaxial cable that supports bus and ring topologies. The device is connected to the network through the medium. A network contains a backbone cable that requires terminal resistor at both ends of the backbone to eliminate the signal reflection.

7.9.2 Electrical specifications

Receiver signal voltage: ≤ 20 Vpp.

Transmitter signal voltage: ≤ 20 Vpp.

Clock jitter range: ≤ 10 ns.

Time synchronization accuracy: ≤ 100 ns.

Electrical isolation characteristics: transformer coupling.

7.9.3 Transmission Medium Interface

The input and output interfaces contain the same 2 signal lines to connect to the network, this pair of signal lines is differential mode. Table 43 shows the interface signal definition.

Table 43 – Transmission medium interface

Short name	Name
IO	Tx/Rx +
/IO	Tx/Rx -

The minimum requirements for the AUTBUS physical transmission medium interface are defined as follows:

Number of pins: 2

Power supply characteristics: 12 VDC ~ 36 VDC

Connector: 50 Ω ~ 100 Ω

Couplers: capacitance coupling, transformer coupling

Isolator: photoelectric isolation or safety barrier is required on individual occasions

Working temperature: -40 °C ~ 150 °C

Carrier bandwidth: 1,536 MHz ~ 32,256 MHz

Transmit power: 10 ~ 20 dBmW

Receive power: -82 ~ 0 dBmW

Sensitivity: -82 dBmW (min)

7.9.4 Medium

7.9.4.1 Cable

The transceiver's preferred cable is a fully shielded single pair of twisted-pair cables. As long as all devices share shielding or grounding at one end of the cable (typically at the end of the control room) and all shielding are connected to this ground, the cable is combined in a given network. Unshielded cables should only be used if known ambient noise and crosstalk are sufficiently low to ensure that the BER is in the acceptable limit.

The signal coding mode on the cable is OFDM (constellation mapping from 16QAM to 256QAM), and the polarity of the signal is a differential balanced twisted pair with no polarity.

The preferred minimum wire size: 0,511 mm diameter (24 AWG).

7.9.4.1.1 Twisted pair

Twisted-pair medium that comply with the AUTBUS PhL medium specification for connecting or testing fieldbus devices should meet the following minimum requirements:

Impedance: 100 Ω ~ 120 Ω

Capacitance: < 50 pF/m

Velocity factor: > 60 %

Frequency band: > 50 MHz

Attenuation: 30 dB/100 m@100 MHz

Temperature: -40 °C ~ 150 °C

Withstand voltage: 500 VMS

Insulation voltage: 36 VDC

Maximum current: $\geq$ 500 mA

7.9.4.1.2 Coaxial cable

Coaxial cable medium that comply with the AUTBUS PhL medium specification for connecting and testing fieldbus devices should meet the following minimum requirements:

Impedance: 50 Ω

Capacitance: < 100 pF/m

Velocity factor: > 70 %

Frequency band: > 50 MHz

Attenuation: 30 dB/100 m@100 MHz

Temperature: -40 °C ~ 150 °C

Withstand voltage: 500 VMS

Insulation voltage: 36 VDC

Maximum current: ≥ 500 mA

7.9.4.2 Connector

The cable connects multiple devices to form the AUTOBUS network, and any device connects with the cable medium through the connector to implement the communication with other devices, which can form a network of up to 254 nodes. Figure 27 shows the connector of the shielded twisted pair.

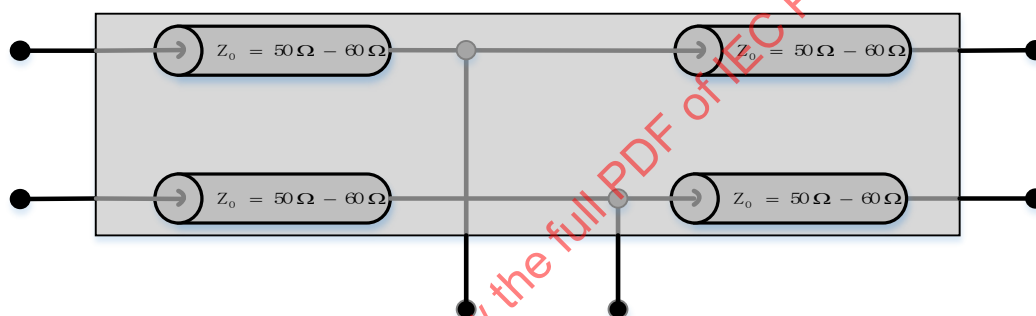


Figure 27 – Connector of the shielded twisted pair

7.9.4.3 Terminal resistor

The terminal resistor should be located at both ends of the backbond cable and connect one signal wire to another. The terminal resistor should not be connected to the cable shield. The impedance characteristics of the terminal resistor should be consistent with the impedance characteristic parameters of the connected cable. Figure 28 shows the terminal resistor connected to the twisted pair receiver.

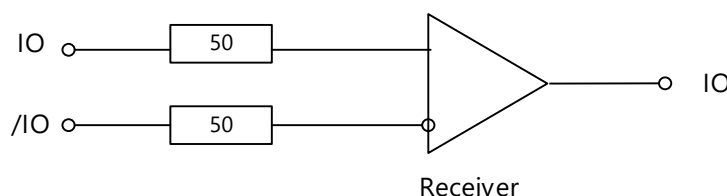


Figure 28 – Terminal resistor

8 Data-link layer

8.1 General

8.1.1 DLL protocol architecture

Figure 29 shows the architecture of the AUTOBUS DLL.

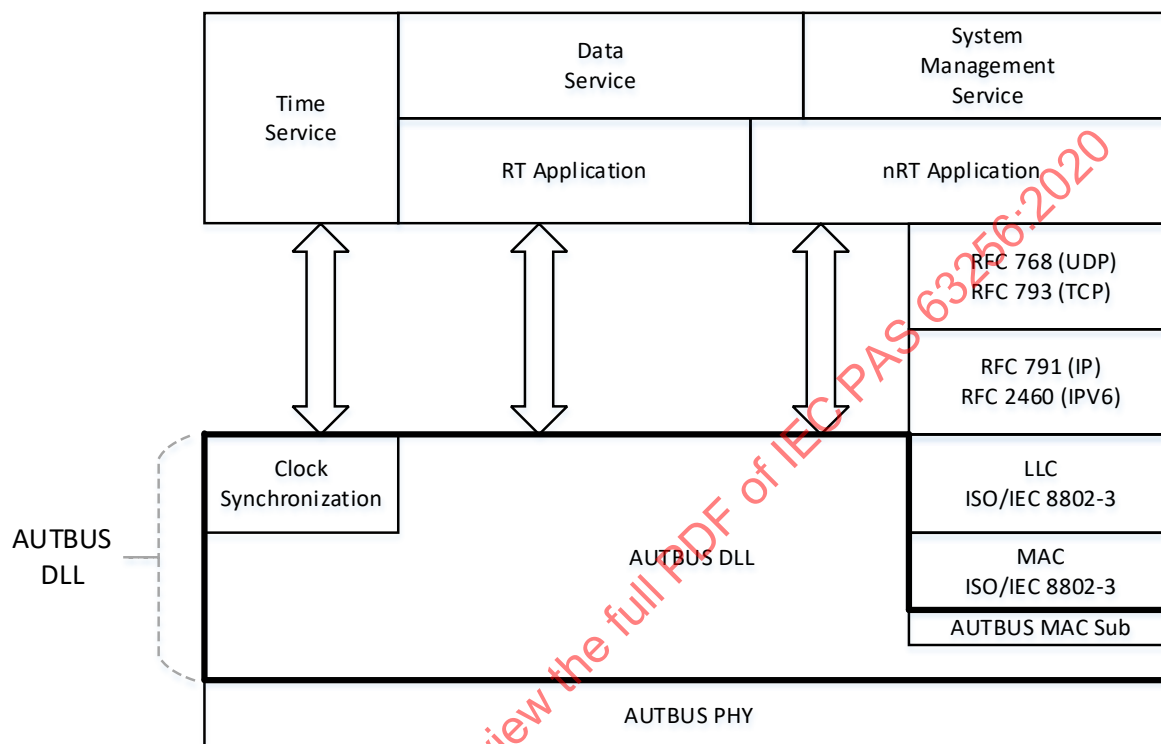


Figure 29 – DLL in AUTOBUS protocol stack architecture

AUTOBUS Data-link Service (DLS) is provided by the AUTOBUS Data-link Protocol (DLP) making use of the services available from the AUTOBUS PhL. The DLS characteristics and function defined in Clause 8 should be exploited by the higher-level protocol. AUTOBUS DLL provides Data-link Management Service (DLMS) for system management. The relationship between the standards for AUTOBUS DLS, AUTOBUS DLP, AUTOBUS AL Protocol and Systems Management is illustrated in Figure 30.

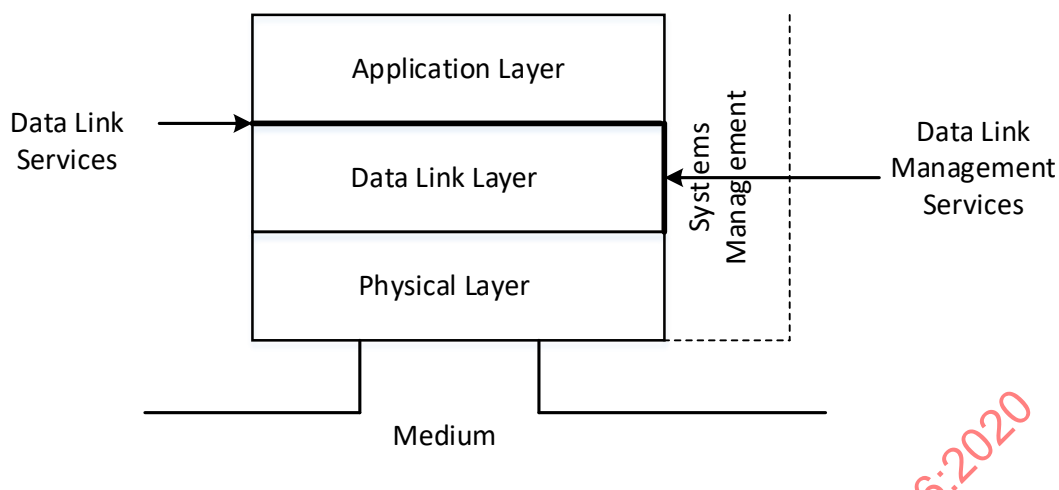


Figure 30 – Relationship of AUTBUS DLL to other filedbus layers and to users of AUTBUS DLS

The services provided by AUTBUS DLL to upper users include DLS, DLMS and Data-link Clock Synchronization Service (DLCSS). AUTBUS Data-link layer Entity (DLE) includes Data-link Clock Entity (DLCE), Data-link Data Entity (DLDE) and Data-link Management Entity (DLME). Based on the communication resource allocation scheme for PhL provided by the AUTBUS DLMS, AUTBUS DLS provides data transmission service with different RT levels through connection and connection-less service. Figure 31 shows the AUTBUS DLL protocol architecture.

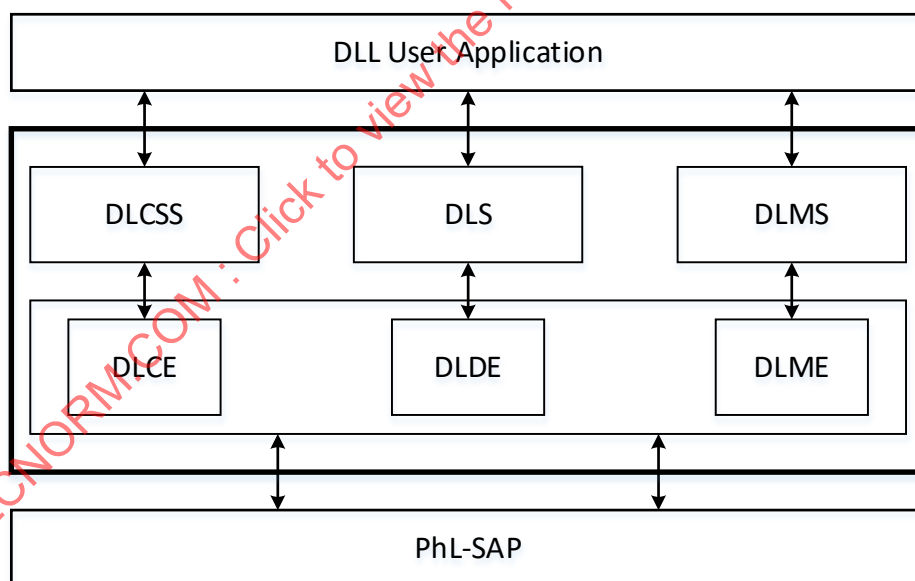


Figure 31 – AUTBUS DLL protocol architecture diagram

Data-link Service (DLS): AUTBUS DLS exploits Data Transmission Channel (DTC) established based on PhL OFDM symbol resources to implement the encapsulation and transmission for application data or protocol data. After the state machine of AUTBUS DLS is started, DLS provides the connection and the connection-less data transmission service based on DLDE for different RT requirements through different DTCs. These channels should be pre-configured or applied and allocated on demand.

Data-link Management Service (DLMS): AUTOBUS DLMS provides system management service, including node joining, node leaving and configuration of network parameters, establishment, update and release of DTCs, etc. AUTOBUS DLL configures DLME through DLMS, implements the encapsulation and parsing of DLPDU including ISO/IEC/IEEE 8802-3 Ethernet DLPDU in DLDE, and the corresponding service function subsets form the AUTOBUS mac sublayer. (AUTOBUS mac sub), as shown in Figure 29.

Data-link Clock Synchronization Service (DLCSS): DLCSS provides the clock synchronization service based on the PhL Service Access Point (PhL-SAP), defining delay measurement, clock synchronization and clock interrupt services.

8.1.2 DLL working mechanism

8.1.2.1 Node

AUTOBUS network contains at least one MN and 0 ~ 253 TNs. Only one active MN is allowed on the AUTOBUS network. The MN should configure all available nodes on the network and manage node access to the network. The TN works normally after it is authorised by MN. The TN sends the data after the MN confirms the available DTC resources.

Each TN in AUTOBUS network support both static configuration and automatic discovery.

8.1.2.2 Addressing

Nodes in AUTOBUS network are addressed on DLL by device NodeID. There are three different types of NodeID: unicast, broadcast and multicast.

Each AUTOBUS node (MN or TN) has a unique node identifier (NodeID) in an AUTOBUS network. NodeID 0 should only be assigned to the MN. NodeID 1 ~ 253 can be used as the address of TNs or multicast group.

The NodeID address assignment of AUTOBUS DLL is shown in Table 44. The unicast address of AUTOBUS network should be globally unique.

Table 44 – NodeID address assignment of AUTOBUS DLL

NodeID	NodeID name	Description
0	DLL_NODEID_MN	AUTOBUS MN
1~253	——	Normal AUTOBUS TN or Multicast group NodeID
254	——	reserved
255	DLL_NODEID_BROADCAST	AUTOBUS broadcast

The NodeID should be assigned by the MN when accessing the network. The source address and the destination address should be identified by NodeID when the DLL DLPDU is transmitted on the network. The NodeID has one-to-one mapping relationship with the device's global unique MAC address in AUTOBUS network that supports ISO/IEC/IEEE 8802-3 Ethernet. When the AUTOBUS network carries the service of ISO/IEC/IEEE 8802-3, the AUTOBUS MN maintains the mapping table between NodeID and the MAC address of ISO/IEC/IEEE 8802-3 in real time, and broadcasts it to all TNs. The NodeID and MAC address mapping table is described in Table 45.

Table 45 – NodeID and MAC address mapping

Item	Description
NodeID	The NodeID address of the device, with a valid range of 0 ~ 253
STATE	Item state, 0 is valid and 1 is invalid
MAC-ADDR	The MAC address of the corresponding node

Annex C shows an example of the mapping of NodeID and MAC address stored on a TN in the AUTOBUS network.

8.1.2.3 Multicast

AUTOBUS protocol specification uses multicast address to implement the data transmission between single node and multiple nodes. The AUTOBUS DLL protocol specifies that the range of multicast type NodeID is 238 ~ 253, that is up to 16 multicast groups are supported. The assignment of multicast type NodeID start from 253 in descending order. The multicast group mapping table is stored on each node. The members of the multicast group mapping table are described in Table 46.

Table 46 – Members of Multicast Group Mapping Table Description

Item	Description
Multicast index	Multicast index starting from 1
Multicast ID	Multicast type NodeID specified in Table 44
Member mask	Multicast member represented by a mask mode, where the mask bit corresponds to the NodeID of the member node

The establishment of multicast group mapping table supports static configuration and dynamic configuration. The static configured multicast mapping table information is sent by the MN to all TNs. The MN determines whether the corresponding TN joins the corresponding multicast group as a member of the multicast group successfully through the response information of the member TN. The dynamic configured multicast group mapping table is applied by the TN according to the requirements of DTC, and the MN should assign the dynamic multicast address after receiving the request. The assignment information is sent to the requesting TN to create or update the corresponding multicast group mapping table.

Annex D shows an example for the AUTOBUS multicast group working mechanism.

8.1.2.4 Resource mapping and scheduling

DTC is a logical channel for transmitting DLPDUs on DLL and it is mapped with the determined PhL communication resource. According to the bus configuration, the AUTOBUS node receives the data from determined DTC. If the data received meets the requirements of the node configuration, the data shall be stored in the packet receiving buffer queue of DLL based on the scheduled result of receiving packet queue, and will be processed by AL user. The data to be sent in the packet transmitting buffer queue of DLL should be taken out based on the scheduled result of the sending message queue, and be scheduled to the requested DTC. The PhL sends out the data on the corresponding symbol resource with setting mode. Figure 32 shows the resource mapping between DLL and PhL.

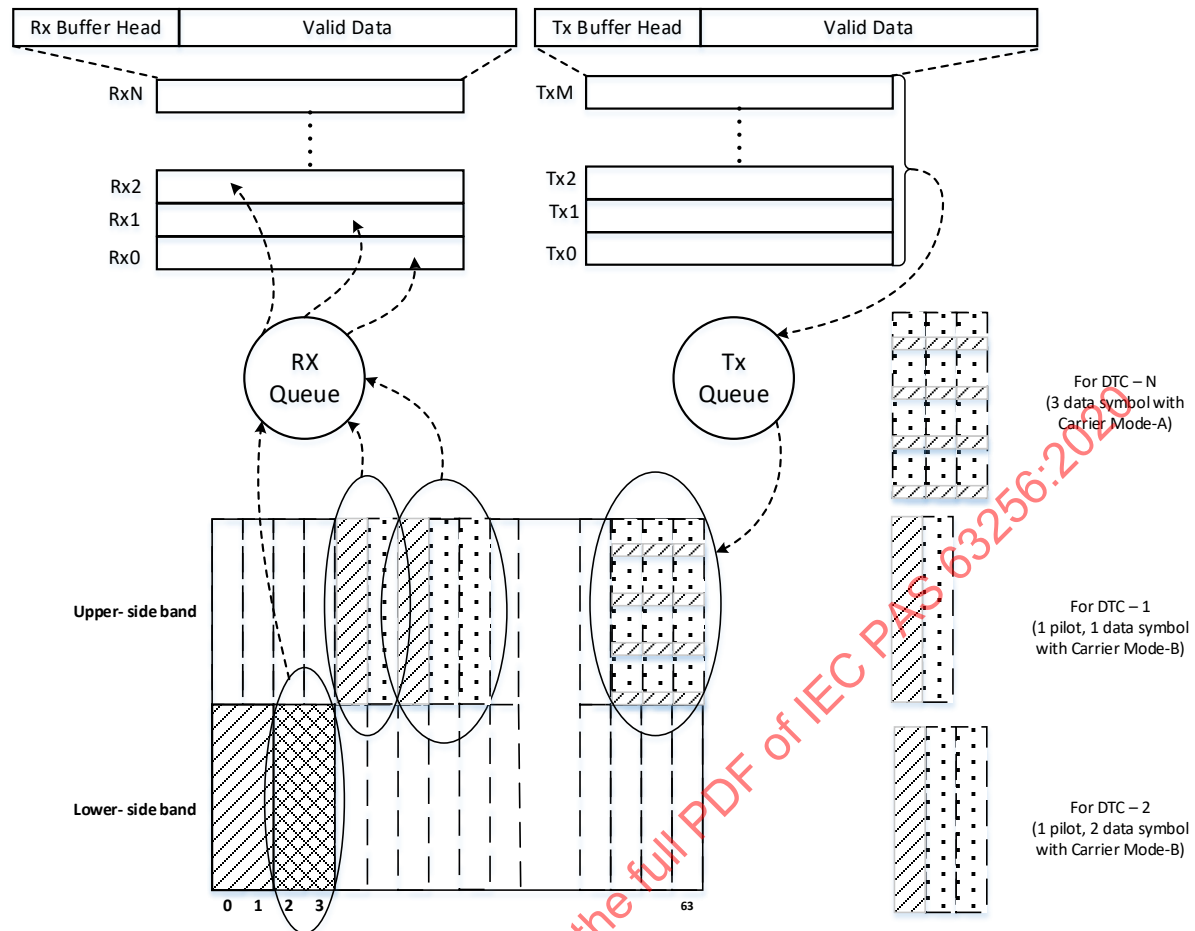


Figure 32 – Resource mapping between DLL and PhL

Description:

- Rx Queue: Receiving message queue
- Tx Queue: Transmitting message queue
- Rx0, Rx1, Rx2....RxN: Packet receiving buffer queue
- Tx0, Tx1, Tx2...TxM: Packet transmitting buffer queue
- Rx Buffer Head: Head information of packet receiving buffer queue that contains data type, queue number, priority, etc.
- Tx Buffer Head: Head information of packet transmitting buffer queue that contains data type, queue number, priority, etc.
- Valid Data: Valid data that has been received or will be sent

When the AUTOBUS DLL receives the packets, it puts the parsed data into packet receiving buffer queue according to the data type by scheduling the receiving message queue; and when sending packets, AUTOBUS DLL puts the data to be sent into the corresponding packet transmitting buffer queue according to the different data types. The valid data is sent out based on the symbol resource of PhL by scheduling the transmitting message queue. In AUTOBUS DLL the packet buffer resources are mapped to the corresponding physical resources by scheduling receiving/transmitting message queue, and the determined data buffer resources are processed on the determined physical resources. This scheduling relationship for receiving message queues and transmitting message queues is shown in Figure 33.

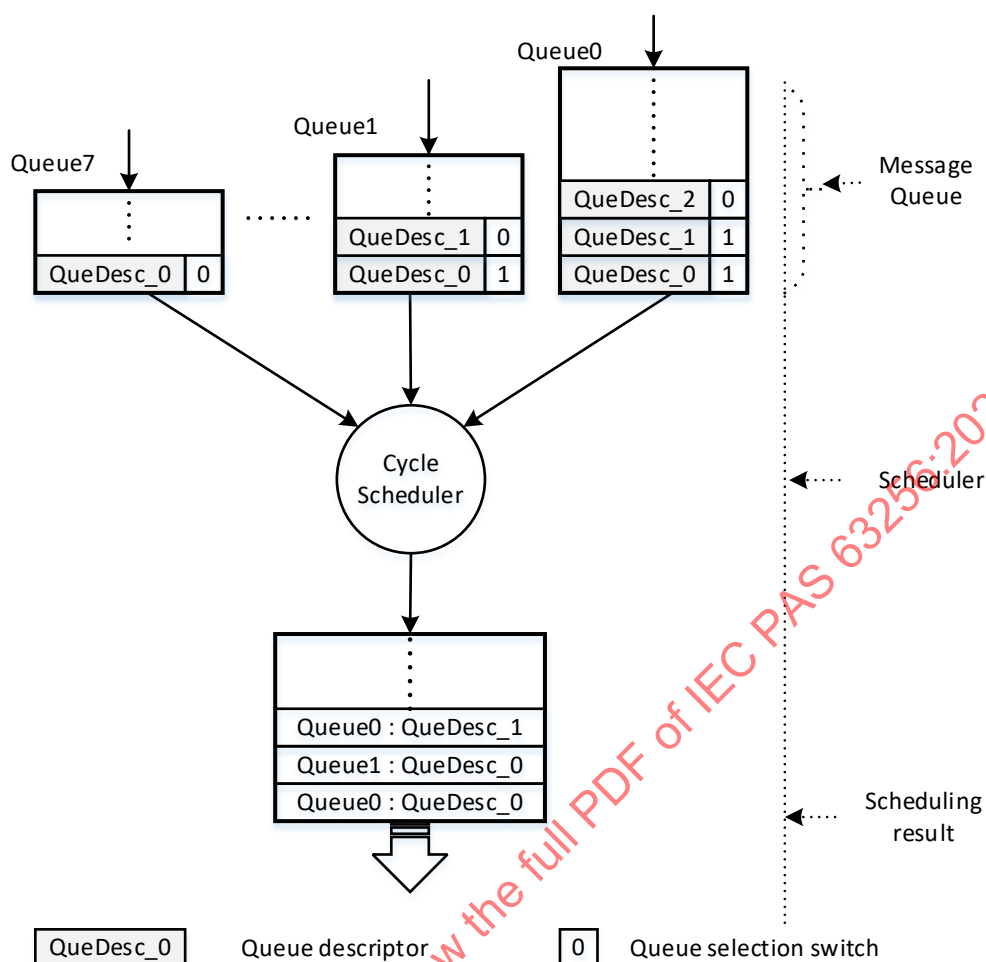


Figure 33 – DLL resource mapping message queue scheduling diagram

Description:

- Queue descriptor: include the data address, the length and the corresponding queue priority of the buffer.
- Queue selection switch: 1 indicates that the corresponding queue is selected, 0 indicates that the corresponding queue is not selected, and the corresponding switch value reset to 0 after being scheduled.
- Scheduler: the cyclic scheduler based on priority, cyclic scheduling of message queues and output the selected queue descriptor information.
- Scheduling result: Queue descriptor information that is output to PhL or DLL.
- Queue number and queue depth: the system supports 8 message queues; each message queue depth can be configured to be different.

8.2 DLL services

8.2.1 General

Subclause 8.2 describes basic packet communication services and models in an automation environment. AUTBUS DLL provides time-critical and non-time-critical communication services. Time-critical refer to the requirement to complete specified functions between devices in a defined time window in an industrial field environment. Failure to complete specified functions within the time window risk failure or harm in industrial production.

Subclause 8.2 defines in an abstract way the externally visible service provided by the AUTBUS DLL in terms of:

- a) the description of service function
- b) primitive sequence diagram with each primitive actions and events
- c) the form of externally service interface and related parameters

The purpose of Subclause 8.2 is to define the services provided to:

- 1) The Fieldbus Application Layer at the boundary between the Application and Data-link Layers of the Fieldbus Reference Model.
- 2) System Management at the boundary between the Data-link Layer and System Management of the Fieldbus Reference Model.

The AUTBUS DLS provides point-to-point or point-to-multipoint data transmission services for DLL users; and it supports both connection and connection-less service through the rational allocation and utilization of physical resources. DLMS provides the service between the DLME and the AL user or system management user; and the service is required for the implementation of DLS. The AUTBUS DLCSS provides accurate clock information for stable operation of time-sensitive services.

8.2.2 Data-link service model

8.2.2.1 Connection-less Mode Data Transmission with Acknowledge (CLMDTA)

Based on the service model, the local DLS-user encapsulates the data request through the DLSDU format and sends the request to a remote DLS-user, at the remote node, if the corresponding DLS-user received the DLPDU correctly, the local DLS-user received a confirmation DLPDU to confirm the success of the data transmission service; if an error occurs during the data transmission, the local DLS-user cannot receive the confirmation DLPDU, it will retransmit the DLPDU, and the maximum number of retransmissions has been set when the system is configured.

8.2.2.2 Connection-less Mode Data Transmission with No Acknowledge (CLMDTNA)

Based on the service model, the local DLS-user encapsulates the data request through the DLSDU format and sends the request to one (unicast) or all (broadcast) remote DLS-users, Whether or not the remote DLS-user received the DLPDU, the local DLS-user will receive a confirmation DLPDU from the local DLE to confirm the success of the data transmission service. On each correctly addressed remote node, if the DLPDU is correctly received, the DLPDU is delivered to the corresponding DLS-user for processing.

8.2.2.3 Connection-less Mode Data Transmission with Request and Acknowledge (CLMDTRA)

Based on the service model, the local DLS-user received a data request from a local or remote node, identifies the variable of the corresponding DLDE, and then according to the setting DLSDU format, sends the response DLPDU to a local or remote node. After receiving the response DLPDU, the data request node confirms that the data request is successful, and sends a response DLPDU to the local DLS-user, the local DLS-user receives the response DLPDU then confirms the success of the service. If there is any error in the response DLPDU during data transmission, it will cause the data request node to transmit the request DLPDU repeatedly, or cause the local DLS-user to retransmit the response DLPDU, the maximum number of retransmissions is set when the system is configured.

8.2.2.4 Connection-less Mode Data Transmission with Request and Response but No Acknowledge (CLMDTRNA)

Based on the service model, the local DLS-user receives data requests from one or more nodes including the local node, and the related variable on local data-link node is identified according to the received requests, the response DLPDU is sent to all data request nodes according to the setting DLSDU structure. The DLSDU and content with this connection-less request and response but no acknowledge for data transmission service, as well as the DTC are fixed, and the local DLS-user always considers the response DLPDU can be received by all data request nodes successfully, and will not resend it again.

8.2.2.5 Connection Mode Data Transmission with Acknowledge (CMDTA)

Based on this service model, local DLS-users implement point-to-point or point-to-multipoint data transmission services. Before data transmission services start, there is a successful DTC between local DLS-user and remote DLS-user. Local DLS-user sends the data requests to remote nodes, if remote DLS-user received the request DLPDU correctly, identifies the corresponding data and transmits the response DLPDU to the local DLS-user. If there is any error in the request or response DLPDU during data transmission, or any other exception causes the local DLS-user not to receive the response DLPDU correctly, the local DLS-users determine the service to be a failure, the local DLS-users send requests repeatedly and the maximum number of repeatedly sent requests is configured during the system configuration. DLS-users exploit the DTC that have been successfully established for data transmission; and the service model satisfies the data transmission reliability requirements and ensures the correct transmission of data on the fieldbus. The service model is usually used in the critical data transmission in the fieldbus.

8.2.2.6 Connection Mode Data Transmission with No Acknowledge (CMDTNA)

Based on the service model, the local DLS-user implements the point-to-point or point-to-multipoint data transmission service, before the data transmission service starts, there is a successful DTC between local DLS-user and remote DLS-users. The local DLS-user sends the data requests to the remote DLS-user and does not require the response of the remote DLS-user, then the data transmission service is determined to be successful. The remote node received the data request correctly and delivers the data to the DLS-user to continue processing according to the addressing information. The request DLPDU and response DLPDU transferred based on the service model have a fixed bandwidth and DLPDU structure.

8.2.3 Data-link service description

8.2.3.1 CLMDTA

8.2.3.1.1 Service function description

The local DLS-user encapsulates a request data based on the DLSDU for the remote DLS-user, this DLSDU is passed to the local DLDE as a data request primitive parameter. The local DLDE accepts the request, forms an appropriate DLPDU containing the DLSDU, and sends the DLPDU to the remote DLE based on the PhL-SAP.

Upon receiving the data DLPDU correctly, the remote DLDE immediately starts transmitting the response acknowledgement DLPDU to the initiating DLE, and sends the confirmation to the local DLDE.

The local DLDE should receive the acknowledgement DLPDU correctly. If the acknowledgement DLPDU is not received within a given slot time (T1, which is set at system configuration) or an incorrect DLPDU is received, the local DLDE again transmit the request data to the remote DLDE. If no correct acknowledgement DLPDU is received after a number of retransmissions equal to Max-Retry-Cnt (parameter which can be configured at system configuration), the Local DLDE reports the failed status in the confirmation primitive which it issues to the local DLS-user.

When the correct acknowledgement DLPDU is received, the local DLDE passes the completion status to the local DLS-user by means of the confirmation primitive, reporting either successful completion of the request service or the incorrect status detected.

8.2.3.1.2 Primitive process

The service primitive process is shown in Figure 34.

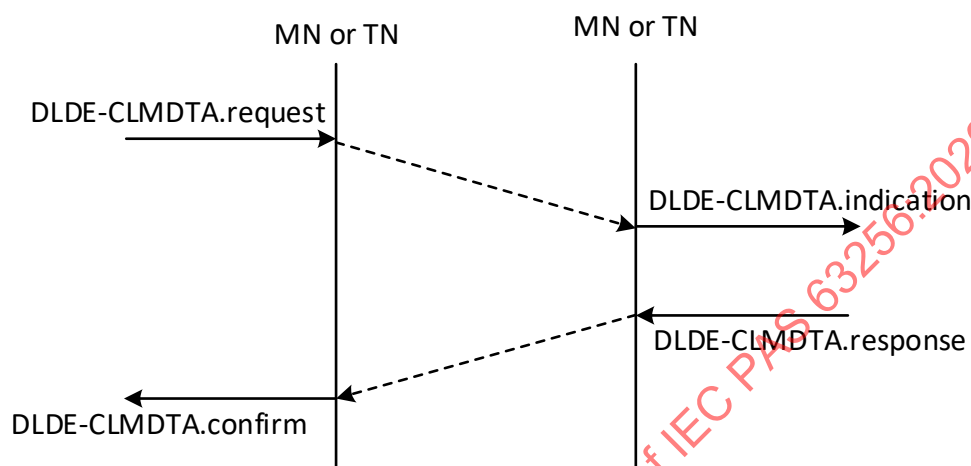


Figure 34 – Primitive process of CLMDTA

Description:

- All nodes are peer entities either as a data provider or a data receiver on a configured AUTOBUS network.
- For CLMDTA services, it is not required to establish a connection between the data receiver and the data provider before data transmission each time, so the acknowledgement DLPDU required by the data request primitive should be received within a configured and definite time slot. If no acknowledgement DLPDU is received or an error DLPDU is received, this data transmission service is failed.

8.2.3.1.3 Service interface

Based on this service interface, the local DLS-user implements the data transmission request and confirmation processing, and the remote DLS-user implements the data transmission indication and response processing.

```

DLDE-CLMDTA (
    Dst_NodeID,
    Src_NodeID,
    DataAddr,
    Ack_ID,
    Priority,
    Length,
    DT_Status
)
  
```

Table 47 indicates the primitives and parameters of the CLMDTA service. The parameters are described in Table 48.

Table 47 – CLMDTA service primitives and parameters

Parameter name	Request	Indication	Response	Confirm
Dst_NodeID	M	M	—	M
Src_NodeID	M	—	M	M
DataAddr	M	M		
Ack_ID	M	—	M	M
Priority	M	M	—	M
Length	M	M	—	—
DT_Status	—	—	M	M

Table 48 – CLMDTA service primitive parameter description

Parameter name	Data type	Input/output
Dst_NodeID	UINT16	IN
Src_NodeID	UINT16	IN
DataAddr	UINT32	IN OUT
Ack_ID	UINT32	IN OUT
Priority	UINT8	IN OUT
Length	UINT16	IN OUT
DT_Status	UINT32	OUT

Parameter description:

- Dst_NodeID: Range 0 to 65 535, the destination node address of the data transmission, see Table 44.
- Src_NodeID: Range 0 to 65 535, the source node address of the data transmission, see Table 44.
- DataAddr: Range is 0 to $(2^{32}-1)$, it is data address information. When proceeding request or response primitive, the address is the buffer address where the data needs to be sent; when proceeding indication or confirmation primitive, the address is the buffer address information of the received data.
- Ack_ID: Range 0 to $(2^{32}-1)$, the data transmission confirmation ID and the value is different each time.
- Priority: Range 0 to 255. It is the priority information to identify the RT level, the smaller value has higher priority.
- Length: Range 0 to 65 535, data length information.
- DT_Status: Range 0 to $(2^{32}-1)$, data transmission service status output value, see Table 59.

8.2.3.2 CLMDTNA

8.2.3.2.1 Service function description

The local DLS-user encapsulates the data request with a DLSDU. The DLSDU is passed to the local DLDE via the CLMDTNA request primitive. The local DLE accepts the request and attempts to send it to the remote DLDE or to all remote DLDEs. The sending DLDE should return a local confirmation of the data transmission service to the local DLS-user.

The service does not require the receiving DLDE to respond or confirm. The DLDE of the data provider and the DLDE of the data receiver should not need to confirm the connection between them before the data transmission, the data provider sends the data in a fixed format. The data receiver receives the data in a fixed format. The data provider should not confirm whether or not the data receiver received the provided data correctly. The receiving DLDE should address according to the address in the DLPDU and pass the data to the corresponding DLS-user.

8.2.3.2.2 Primitive process

The service primitive process is shown in Figure 35.

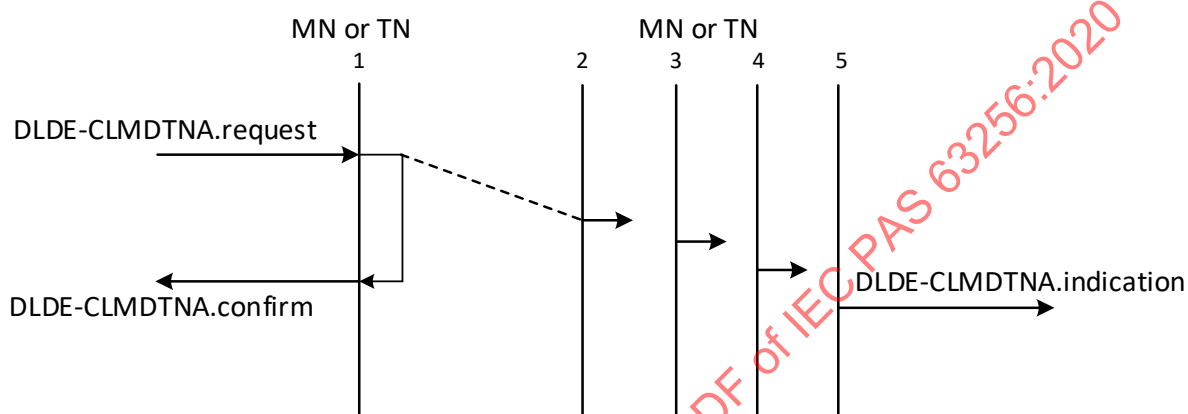


Figure 35 – Primitive process of CLMDTNA

Description:

- All nodes are peer entities either as a data provider or a data receiver on a configured AUTOBUS network.
- The DLDE of data provider receives a data transmission request from the DLS-user and confirms immediately that the service processing is completed. The DLDE of data receiver should address according to the address in the DLPDU and pass the data to the DLS-user for further indication process.

8.2.3.2.3 Service interface

Based on this service interface, the local DLS-user implements the data transmission request and confirmation processing, and the remote DLS-user implements the data transmission indication and response processing.

```

DLDE-CLMDTNA (
    Dst_NodeID,
    Src_NodeID,
    DataAddr,
    Priority,
    Length,
    DT_Status
)
  
```

Table 49 indicates the primitives and parameters of the CLMDTNA service. The parameters are described in Table 50.

Table 49 – CLMDTNA service primitives and parameters

Parameter name	Request	Indication	Confirm
Dst_NodeID	M	M	M
Src_NodeID	M	M	—
DataAddr	M	M	—
Priority	M	M	M
Length	M	M	—
DT_Status	—	—	M

Table 50 – CLMDTNA service primitive parameter description

Parameter name	Data type	Input/output
Dst_NodeID	UINT16	IN
Src_NodeID	UINT16	IN
DataAddr	UINT32	IN OUT
Priority	UINT8	IN OUT
Length	UINT16	IN OUT
DT_Status	UINT32	OUT

Parameter description:

- Dst_NodeID: Range 0 to 65 535, the destination node address of the data transmission, see Table 44.
- Src_NodeID: Range 0 to 65 535, the source node address of the data transmission, see Table 44.
- DataAddr: Range is 0 to ($2^{32}-1$), it is data address information. When proceeding request primitive, the address is the buffer address where the data needs to be sent; when proceeding indication or confirmation primitive, the address is the buffer address information of the received data.
- Priority: Range 0 to 255. It is the priority information to identify the RT level, the smaller value has higher priority.
- Length: Range 0 to 65 535, data length information.
- DataDesc: String format data description information.
- DT_Status: Range 0 to ($2^{32}-1$), data transmission service status output value, see Table 59.

8.2.3.3 CLMDTRA

8.2.3.3.1 Service function description

The local DLS-user identifies the variable of local DLDE and encapsulates the response DLSDU when it received a data transmission request from a remote DLS-user. The response DLPDU carries the request ID of the remote DLS-user, the local response ID and the response data. The DLDE of the remote DLS-user should address according to the address in the DLPDU and pass the DLPDU to the corresponding DLS-user for further processing.

After receiving the response DLPDU sent by the local DLDE, the remote DLDE shall confirm that the request ID in the DLPDU is consistent, then the request ID resource is recycled, and the remote DLS-user confirms that the request service is complete. At the same time, the remote DLDE encapsulates a new response data and sends to the local DLDE according to the response ID in the response DLPDU. The local DLDE receives the new response DLPDU with the local response ID and completes the service.

There is no need to create a connection between the data provider and the data receiver of the service. The data provider transmits data in the setting time slot (T1), and the data receiver receives the data in the setting time slot. Both can process empty data that the corresponding request ID should be an invalid value. The service provides a retransmission mechanism. If the remote data transmission request DLDE do not receive a response data with the request ID or receive an incorrect data, the remote DLS-user retransmits the data transmission request. If the local data response DLDE do not receive the response DLPDU sent by the remote DLDE, the local DLS-user retransmits the respond DLPDU with the request ID and the response ID. The maximum number of times for repeated request (Max_Retry_ReqCnt) and the maximum number of times for repeated response (Max_Retry_ResCnt) are set when the system is configured.

8.2.3.3.2 Primitive process

The service primitive process is shown in Figure 36.

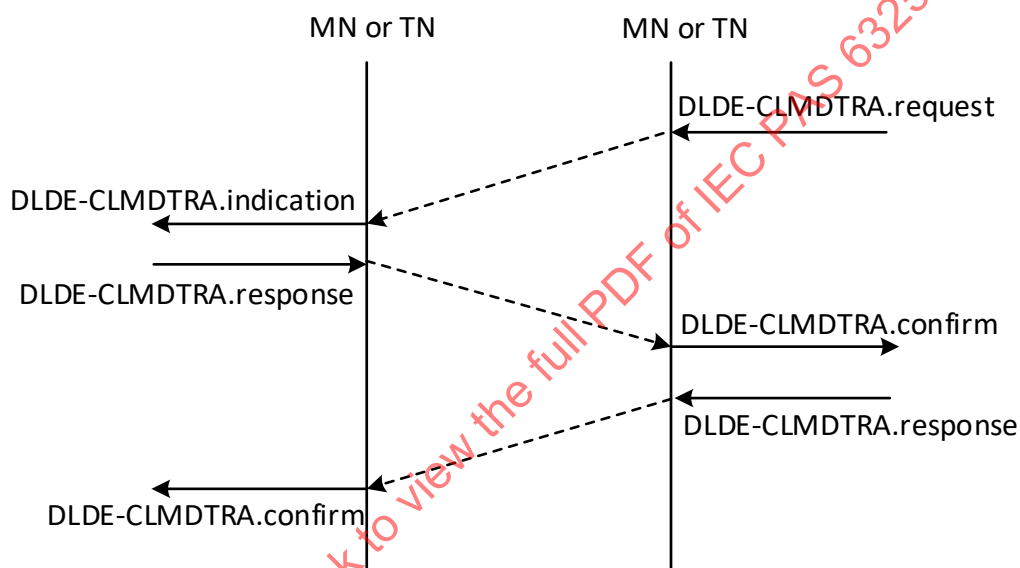


Figure 36 – Primitive process of CLMDTRA

Description:

- If there is no requested DLSDU and DLPDU, the DLDE processes the data with empty, then the data provider and the data receiver process the data in the same manner at the time slot.
- The data provider DLDE cannot rollback the identified data even if the data provider does not receive the response DLPDU. The data received by the data receiver is still valid and the data receiver maybe repeatedly receives the correct and valid data.

8.2.3.3.3 Service interface

Based on this service interface, the local DLS-user implements the data transmission indication, response and confirmation processing, and the remote DLS-user implements the data transmission request, confirmation and response processing.

DLDE-CLMDTRA (
 Dst_NodeID,
 Src_NodeID,
 BuffAddr,
 Req_ID,
 Res_ID,
 Priority,
 Length,
 DT_Status
)

Table 51 indicates the primitives and parameters of the CLMDTRA service. The parameters are described in Table 52.

Table 51 – CLMDTRA service primitives and parameters

Parameter name	Request	Indication	Response	Confirm
Dst_NodeID	M	—	M	M
Src_NodeID	M	M	M	—
DataAddr	M	M	M	—
Req_ID	M	M	M	M
Res_ID	—	—	M	M
Priority	M	M	M	M
Length	M	M	M	—
DT_Status	—	—	M	M

Table 52 – CLMDTRA service primitive parameter description

Parameter name	Data type	Input/output
Dst_NodeID	UINT16	IN
Src_NodeID	UINT16	IN
DataAddr	UINT32	IN OUT
Req_ID	UINT32	IN OUT
Res_ID	UINT32	IN OUT
Priority	UINT8	IN OUT
Length	UINT16	IN OUT
DT_Status	UINT32	OUT

Parameter description:

- Dst_NodeID: Range 0 to 65 535, the destination node address of the data transmission, see Table 44.
- Src_NodeID: Range 0 to 65 535, the source node address of the data transmission, see Table 44.
- DataAddr: Range is 0 to $(2^{32}-1)$, it is data address information. When proceeding request or response primitive, the address is the buffer address where the data needs to be sent; when proceeding indication or confirmation primitive, the address is the buffer address information of the received data.
- Req_ID: Range 0 to $(2^{32}-1)$, request ID; $(2^{32}-1)$ is invalid ID.
- Res_ID: Range 0 to $(2^{32}-1)$, response ID; $(2^{32}-1)$ is invalid ID.

- Priority: Range 0 to 255. It is the priority information to identify the RT level, the smaller value has higher priority.
- Length: Range 0 to 65 535, data length information.
- DT_Status: Range 0 to ($2^{32}-1$), data transmission service status output value, see Table 59.

8.2.3.4 CLMDTRRNA

8.2.3.4.1 Service function description

The local DLS-user shall receive data transmission request from one or more remote DLS-users, or receive a local DLS-user's own data transmission request. The local DLS-user identifies the local data after received the request DLSDU. If the identified data meet indication condition, the data shall be encapsulated as a response DLPDU and transmitted to all request DLDE that maybe contain the local DLDEs to complete the request service processing.

The indication condition that the local DLS-user should meet is determined when the system is configured, and the condition does not depend on the number of data transmission requests received by the local DLS-user. The local DLS-user takes the identified response data information as a parameter and sends it to the local DLDE. The DLDE selects the correct destination address information according to the identified parameter. The DLDE forms a DLPDU according to the fixed format and sends it to the corresponding DLDE through the PhL-SAP. If the response DLPDU is received by the remote DLDE or local DLDE receive, it should be delivered to the corresponding DLS-user for further processing according to the destination address information.

After the local DLDE encapsulated and sent the response DLPDU; that is the service has been processed successfully, and there is no need to confirm whether the remote user has receive the data or receive the data correctly. The response DLPDU is also not sent repeatedly.

8.2.3.4.2 Primitive process

The service primitive process is shown in Figure 37.

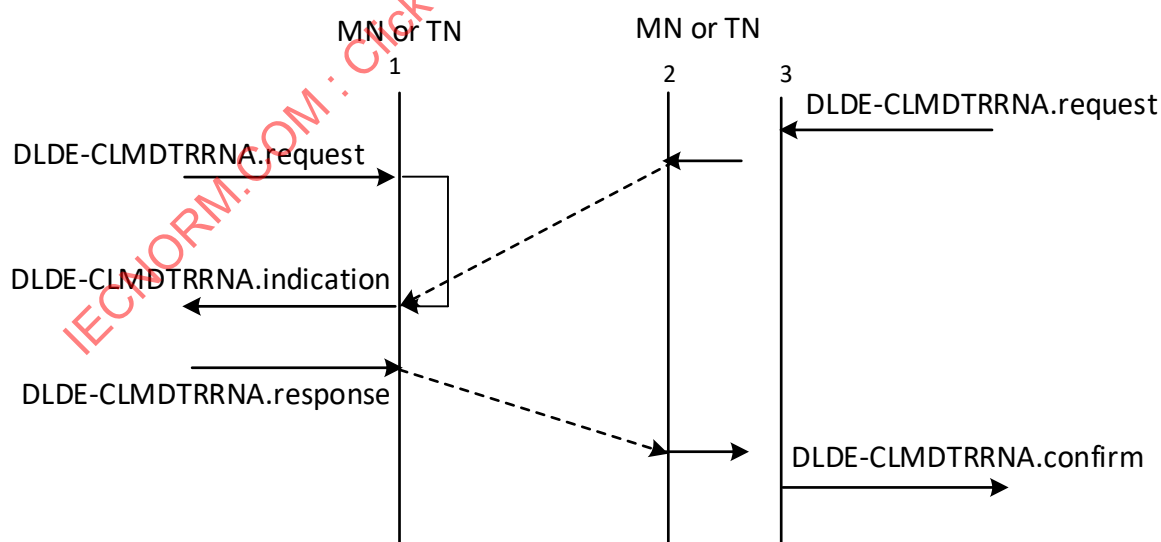


Figure 37 – Primitive process of CLMDTRRNA

Description:

- The DLPDU of the data transmission request maybe empty, in this status, the local DLS-user express the default data content, the corresponding DLDE encapsulates the DLPDU with the default data content and sends it to the corresponding remote node.

- b) If the data receiver does not send a data transmission request, or the data request is empty, the data receiver should discard the response data when it received it.

8.2.3.4.3 Service interface

Based on this service interface, the local DLS-user implements the data transmission request, indication and response processing, the remote DLS-user implements the data transmission request and the confirmation processing.

DLDE-CLMDTRRNA (

Dst_NodeID,
Src_NodeID,
DataAddr,
Priority,
Length,
DT_Status

)

Table 53 indicates the primitives and parameters of the CLMDTRRNA service. The parameters are described in Table 54.

Table 53 – CLMDTRRNA service primitives and parameters

Parameter name	Request	Indication	Response	Confirm
Dst_NodeID	M	—	M	M
Src_NodeID	M	M	M	—
DataAddr	M	M	M	—
Priority	M	M	M	M
Length	M	M	M	—
DT_Status	—	—	M	M

Table 54 – CLMDTRRNA service primitive parameter description

Parameter name	Data type	Input/output
Dst_NodeID	UINT16	IN
Src_NodeID	UINT16	IN
DataAddr	UINT32	IN OUT
Priority	UINT8	IN OUT
Length	UINT16	IN OUT
DT_Status	UINT32	OUT

Parameter description:

- Dst_NodeID: Range 0 to 65 535, the destination node address of the data transmission, see Table 44.
- Src_NodeID: Range 0 to 65 535, the source node address of the data transmission, see Table 44.
- DataAddr: Range is 0 to $(2^{32}-1)$, it is data address information. When proceeding request or response primitive, the address is the buffer address where the data needs to be sent; when proceeding indication or confirmation primitive, the address is the buffer address information of the received data.
- Priority: Range 0 to 255. It is the priority information to identify the RT level, the smaller value has higher priority.

- Length: Range 0 to 65 535, data length information.
- DT_Status: Range 0 to ($2^{32}-1$), data transmission service status output value, see Table 59.

8.2.3.5 CMDTA

8.2.3.5.1 Service function description

Before data transmission between the local DLS-user and the designated remote DLS-user, the connection channel is established through the specified parameters. The local DLS-user encapsulates the data transmission request as a DLSDU and delivers it to the local DLDE. The local DLDE encapsulates the request as a DLPDU, and calls the PhL access service interface to send it to the remote designated DLDE.

Based on confirmed connection channel, the DLPDU with data transmission request is expected to be correctly transmitted to a remote DLDE, which address is based on the address in the DLPDU, the corresponding DLS-user is notified to process and identify the corresponding data, then the new DLPDU is encapsulated and passed to the local DLDE, response to the local data transmission request processing. The local DLS-user received the response data and identifies the completion status of the data transmission service based on the information in the response data.

If there is any error in the data transmission process, the local DLDE does not receive the response DLPDU correctly, or the response DLPDU received contains error status information. The local DLS-user retransmits the request DLSDU, and the maximum number of retransmissions (Max_Retry_Cnt) is set when the system is configured.

8.2.3.5.2 Primitive process

The service primitive process is shown in Figure 38.

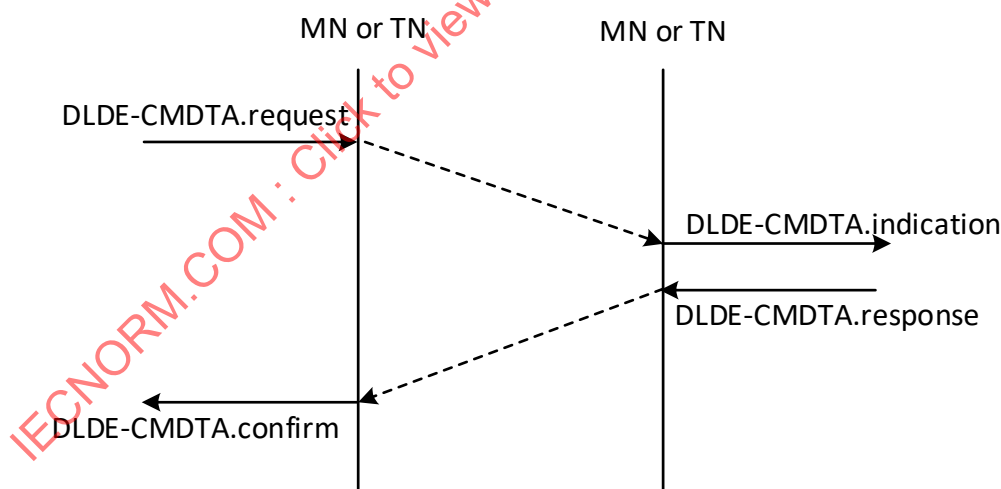


Figure 38 – Primitive process of CMDTA

Description:

- For data transmission based on connection mode, when the corresponding DLDE process the DLS-user primitive operation, each service ends with the confirmation packet received by the local DLS-user.
- The data transmission based on connection mode requires to establish confirmed connection channel before each data transmission request.

8.2.3.5.3 Service interface

The local DLS-user based on this service interface writes the data information of the specified length to the destination buffer. The written data shall be sent out on the specified channel based on the scheduling of state machine. The remote DLS-user reads valid data from a given buffer through the service interface and responds to the data transmission request from the local side based on the service interface.

```
DLDE-CMDTA (
    Dst_NodeID,
    Src_NodeID,
    DataAddr,
    Channel_ID,
    Priority,
    Length,
    DT_Status
)
```

Table 55 indicates the primitives and parameters of the CMDTA service. The parameters are described in Table 56.

Table 55 – CMDTA service primitives and parameters

Parameter name	Request	Indication	Response	Confirm
Dst_NodeID	M	—	M	M
Src_NodeID	M	M	—	—
DataAddr	M	M	M	—
Channel_ID	M	M	M	M
Priority	M	—	M	—
Length	M	M	M	—
DT_Status	—	—	M	M

Table 56 – CMDTA service primitive parameter description

Parameter name	Data type	Input/output
Dst_NodeID	UINT16	IN
Src_NodeID	UINT16	IN
DataAddr	UINT32	IN OUT
Channel_ID	UINT32	IN OUT
Priority	UINT8	IN OUT
Length	UINT16	IN OUT
DT_Status	UINT32	OUT

Parameter description:

- Dst_NodeID: Range 0 to 65 535, the destination node address of the data transmission, see Table 44.
- Src_NodeID: Range 0 to 65 535, the source node address of the data transmission, see Table 44.
- DataAddr: Range 0 to $(2^{32}-1)$, the buffer address information of the data
- Channel_ID: Range 0 to $(2^{32}-1)$, connection channel ID

- Priority: Range 0 to 255. It is the priority information to identify the RT level, the smaller value has higher priority.
- Length: Range 0 to 65 535, data length information.
- DT_Status: Range 0 to ($2^{32}-1$), data transmission service status output value, see Table 59.

8.2.3.6 CMTNA

8.2.3.6.1 Service function description

Before data transmission between the local DLS-user and the designated remote DLS-user, the connection channel is established through the specified parameters. The local DLS-user encapsulates the data transmission request as a DLSDU and passes it to the local DLDE with the specified connection channel ID as a parameter. The local DLDE simultaneously confirm the completion status of the service to the local DLS-user. The local DLDE shall encapsulate the data as a DLPDU and transmit it to the remote DLDE by means of PhL service interface.

After receiving the DLPDU based on the confirmed connection channel, the remote DLDE identifies the data and addresses the destination based on the address information in the DLPDU, then passes it to the corresponding DLS-user for further processing.

The local DLS-user implements the data transmission request as the data provider, the remote DLS-user implements the data receiving and the data identification in the corresponding DLDE as the data receiver. The DLS provides reliable data transmission service through established and confirmed connection channels.

8.2.3.6.2 Primitive process

The service primitive sequence is shown in Figure 39.

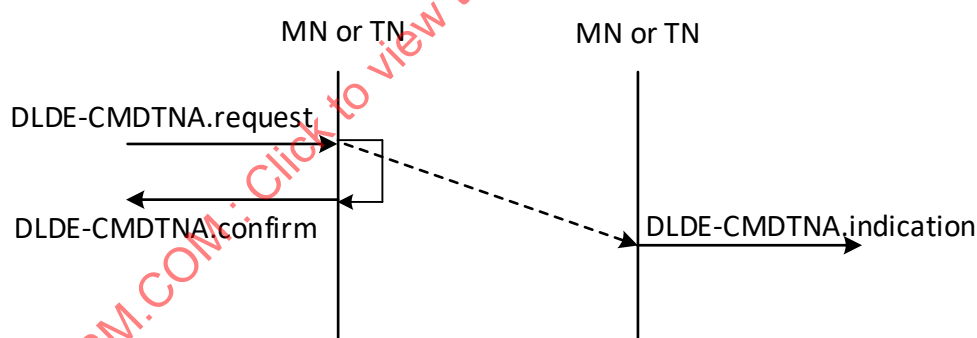


Figure 39 – Primitive process of CMTNA

8.2.3.6.3 Service interface

The local DLS-user implements the data transmission request and confirmation processing based on this service interface, the connection channel ID should be the confirmed channel ID that has been established before the data transmission request. The remote DLS-user implements the data receiving and data identification processing based on the service interface, the remote DLDE should only receive and process the data based on the current designated channel.

DLDE-CMDTNA (
 Dst_NodeID,
 Src_NodeID,
 DataAddr,
 Channel_ID,
 Priority,
 Length,
 DT_Status
)

Table 57 indicates the primitives and parameters of the CMDTNA service. The parameters are described in Table 58.

Table 57 – CMDTNA service primitives and parameters

Parameter name	Request	Indication	Confirm
Dst_NodeID	M	—	M
Src_NodeID	—	M	—
DataAddr	M	M	—
Channel_ID	M	M	M
Priority	M	—	—
Length	M	M	—
DT_Status	—	—	M

Table 58 – CMDTNA service primitive parameter description

Parameter name	Data type	Input /output
Dst_NodeID	UINT16	IN
Src_NodeID	UINT16	IN
DataAddr	UINT32	IN
Channel_ID	UINT32	IN
Priority	UINT8	IN
Length	UINT16	IN
DT_Status	UINT32	OUT

Parameter description:

- Dst_NodeID: Range 0 to 65 535, the destination node address of the data transmission, see Table 44.
- Src_NodeID: Range 0 to 65 535, the source node address of the data transmission, see Table 44.
- DataAddr: Range 0 to $(2^{32}-1)$, the buffer address information of the data
- Channel_ID: Range 0 to $(2^{32}-1)$, connection channel ID
- Priority: Range 0 to 255. It is the priority information to identify the RT level, the smaller value has higher priority.
- Length: Range 0 to 65 535, data length information.
- DT_Status: Range 0 to $(2^{32}-1)$, data transmission service status output value, see Table 59.

NOTE For data transmission in connection mode, it is necessary to create a connection channel before the data transmission request. After the end of the data transmission service, the user can choose to release or continue to keep the connection channel according to the business requirement. If releasing the connection channel is selected, the new connection channel needs to re-create before the next data transmission request, then for data transmission service.

Table 59 – DLS status output value description

Status name	Status coding	Status description
DLS_SUCCESS	0x100	Data-link service success
DLS_FAILED	0x101	Data-link service failure
DLS_ERR_CONNECT	0x102	Data-link connection error
DLS_ERR_PARA	0x103	Data-link service parameter error
DLS_ERR_DATA	0x104	Data-link service data error
DLS_ERR_CRC	0x105	Transmitted data CRC error
DLS_DLDE_NO_EXIST	0x106	DLDE does not exist
DLS_ERR_UNKNOW	0x10F	Unknown error in data-link service

8.2.4 Data-link management service description

8.2.4.1 Config

8.2.4.1.1 Service function description

The data-link configuration management service provides the configuration of the basic parameters, default physical resources and default running parameters for the DLMS-user on the local DLME or remote DLME. This service should ensure the local node or the remote nodes can run correctly and effectively on the AUTOBUS network. The basic configuration parameters of the DLME include version number, transmit mode, working mode, node address (NodeID), Ethernet MAC address and the node corresponding multicast group ID. The default physical resource includes random access physical communication resource allocation and the way that nodes deal with communication resources. The network operation parameters of AUTOBUS include multicast membership table address mapping table and so on.

The DLMS-user sends the config management request to the local DLME or remote DLME, the local or remote DLME identifies the variables of the corresponding entity content according to the received request and response to the local DLMS-user or the remote DLMS-user according to the identification result. The local DLMS-user or the remote DLMS-user shall confirm the status and result of this service according to the received response.

8.2.4.1.2 Primitive process

Figure 40 shows this service primitive process.

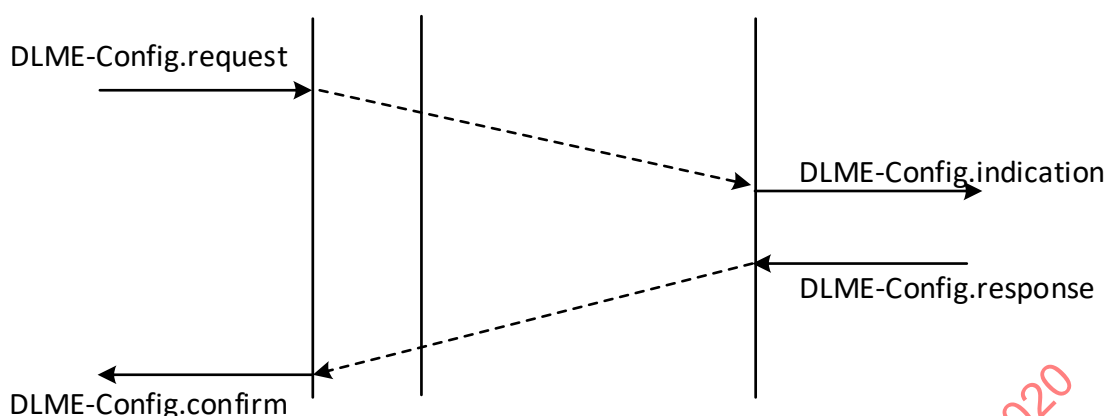


Figure 40 – Data-link config management service primitive process

8.2.4.1.3 Service interface

The DLMS-user sends the config request to the local or remote DLME based on the DLME-Config interface. The DLMS-user also receives the confirmation from the local or remote DLME, and complete the indication processing of the corresponding entity content based on the config request. The indication processing results are responded to the local or remote DLMS-users.

```

DLME-Config (
    CfgType,
    Dst_NodeID,
    MacAddr,
    CfgParamDesc,
    Mgt_Status
)
    
```

Table 60 indicates the primitives and parameters of this service. The parameters are described in Table 61.

Table 60 – Data-link config management service

Parameter name	Request	Indication	Response	Confirm
CfgType	M	—	M	M
Dst_NodeID	M	M	M	M
MacAddr	M	M	—	—
CfgParamDesc	M	M	—	—
Mgt_Status	—	—	M	M

Table 61 – Data-link config management service primitive parameter description

Parameter name	Data type	Input/output
CfgType	UINT16	IN
Dst_NodeID	UINT16	IN
MacAddr	ARRAY	IN
CfgParamDesc	STRUCT	IN OUT
Mgt_Status	UINT32	OUT

Parameter description:

- CfgType: Range 0 to 65 535, configuration type parameter:
 - 0: Invalid config;
 - 1: Basis config;
 - 2: Address config;
 - 4: Resource mapping config;
 - 8: Mapping table config;
 - Others reserved;
- Dst_NodeID: Range 0 to 65 535, configured node address, see Table 44.
- MacAddr: MAC address with 6 octets, the configured MAC address of node device is same as the fixed MAC address of node, if the value is 0, then all nodes should be configured.
- CfgParamDesc: Description for config parameter information. See Table 62 for the description of the structure CFG_PARAM_INFO.
- Mgt_Status: Range 0 to ($2^{32}-1$), config service status return value. See Table 80.

Table 62 – CFG_PARAM_INFO structure description

Member name	Data type	description
Version	UINT32	The version number with dotted decimal: XX.XXXX.XX bit31-bit24: main version number bit23-bit8: Minor version number bit7-bit0: Protocol publishing serial number The protocol version number should be notified to all nodes to ensure that the protocol version is consistent, and the other two version numbers are used as the MN protocol stack software version properties for user program verification reference.
NodeRole	UINT8	Link node's control or terminal property, valid value is 0 ~ 2; 0: MN 1: TN 2: do not distinguish between MN and TN. Other values reserved.
TransmitMode	UINT8	System TM, high 4 bits indicates MN node TM, low 4bit bit indicates TN node TM, 0 ~ 3 is valid (see Table 18). Other values reserved.
WorkMode	UINT16	The node PhL working mode in default, and the physical symbol resources of different modes can carry different effective data. The high 8 bits indicates the default working mode of the upper-side band, and the low 8 bits indicates the default working mode of the lower-side band. Specified in Table 19 and Table 20 for working mode description, the valid range of the working mode is 0 ~ 21
AccessStatus	UINT8	Access status 0: initialization 1: run 2: pause 3: offline Other values reserved
StartNodeID	UINT8	Configured start NodeID that is the corresponding start NodeID value in the MAC address mapping table.
UnicodeNum	UINT8	Numbers of effective nodes in the network, the node number is incremented from the StartNodeID.
GroupNum	UINT8	Numbers of avild multicast groups in this network configuration

Member name	Data type	description
TimeOutCfg	STRUCT	The related config information for DLL timeout processing, see Table 63 for the TIMEOUT_CFG structure.
GroupIDMapLst	STRUCT	Multicast groups list, including valid multicast group ID and multicast member bitmaps, see Table 64 for the description of the GROUP_IDMAP_S structure.
NodeIDMacLst	STRUCT	NodeID and MAC address mapping structure, refer to the description of NODEID_MAC_S structure in Table 65. The number of mapping tables is determined by the parameter UnicodeNum.
CommResCfg	STRUCT	A general configuration structure for the allocation of physical symbol resources reserved for random access by users or nodes, refer to the description of COMM_RES_CFG structure in Table 66.
NOTE For the setting of structure members, see Table 13 and Table 14.		

Table 63 – TIMEOUT_CFG structure description

Member name	Data type	description
T1	UINT16	Timeout configuration, the unit is us
T2	UINT16	Cycle time configuration, unit is us, which should be the integer multiple of the physical SF cycle time.
T3	UINT16	The heartbeat detection cycle time, the unit is us, which should be the integer multiple of the physical SF cycle time.
Max_Discovery_timeout	UINT32	Max discovery timeout, the unit is us.
Max_Heartbeat_timeout	UINT8	Heartbeat detection timeout frame cycle count
Max_Retry_ReqCnt	UINT8	Max retry request count
Max_Retry_ResCnt	UINT8	Max retry response count
Max_Retry_Cnt	UINT8	Max repeat count
Max_ReqIn_Cnt	UINT8	Max random access request count
Max_Establish_Req_Retry_Cnt	UINT8	Max establish request count
Max_Update_Req_Retry_Cnt	UINT8	Max update request count
NOTE For the setting of structure members, see Table 15.		

Table 64 – GROUP_IDMAP_S structure description

Member name	Data type	description
Group_ID	UINT16	Valid multicast number decrease from 253. See Table 44.
Reserved	UINT16	Reserved
NodeID_Map	BITMAP	The multicast group members with bitmap format, NodeID is the corresponding bit serial number.

Table 65 – NODEID_MAC_S structure description

Member name	Data type	description
MACAddr	ARRAY	MAC address with 6 octets. MAC address shall be defined according to the description in ISO/IEC/IEEE 8802-3.

Table 66 – COMM_RES_CFG structure description

Member name	Data type	description
DownStream_Symbol_Num	UINT8	The number of half-side band symbol occupied by a downlink subframe, which should be greater than or equal to 2, default is 2.
Cycle_Frame_Num	UINT8	Configure the number of cyclic frame cycles for processing symbol resources, the user can configure symbol resources within consecutive Cycle_Frame_Num frame cycles. The valid value is 0 ~ 8, the others values are invalid. 0 indicates that the configuration does not do cycle processing, only once. And a value of 8 means that the system circularly processes the symbol resources in 8 consecutive frame cycles (i.e. 512 OFDM symbols).
R_RT_Attr	UINT16	Configure RT properties of randomly access channel resources: 0: Once configured, random access channel resources will be recycled and not released. 1 ~ 512: The number of that the random access channel resource can be duty-cycle operation after being configured during this configuration cyclic frame cycle, the number that the corresponding symbol resource block (a symbol resource block composed of R_Symbol_Num consecutive symbols from R_Symbol_Start) are duty-cycle operation. The maximum value is 512 and other values are not supported; for example, a value of 2 indicates that the physical symbol resource block corresponding to the channel will be released after two consecutive uses.
R_Band_Flag	UINT8	The upper or lower side band of the symbols based on the bit identification, bit0: the upper-side band, bit1: the lower-side band, other bits are invalid, and the corresponding bit position 1 means that the reserved resources use the corresponding side band symbols.
R_Band_Mode	UINT8	The carrier mode of the symbol resource block corresponding to the random-access channel (starting with the OFDM symbol determined by the R_Symbol_Start, with consecutive R_Symbol_Num symbols being the symbol resource block of the random access channel), is shown in Table 19 and Table 20, valid value range is 0 ~ 21
R_Cycle_Symbol_Len	UINT16	The cyclic symbol cycle length of the random-access channel resource, after the channel is created, from the OFDM symbol determined by R_Symbol_Start, the consecutive Cycle_Symbol_Len symbol period is processed once. The current valid range is 1 ~ 512. Others are invalid.
R_Symbol_Start	UINT16	Specify the starting symbol serial number of the reserved cyclic symbol period for the random-access channel, valid range is 0 ~ 511. Others are invalid.
R_Symbol_Num	UINT16	Specify the number of valid resource symbols reserved from R_Symbol_Start for random access channels, valid range of 1 ~ 512. Others are invalid.

8.2.4.2 Discovery

8.2.4.2.1 Service function description

The data-link discovery management service provides support for the MNs to maintain the working state of each node on the AUTOBUS network. The MN can initiate the query for the current working state of each TN. If no response received from the node within the setting timeout (Max_Discovery_Timeout), the node is considered to be offline state, and all node states are updated to the network. The dynamic access TN shall request the access authorization and the communication resource to MN through this service. Once the allocated resource has been confirmed by the dynamic access TN, the MN maybe updates the status of all nodes on the network according to the latest configuration. If no response from MN after the number of request equal to the maximum number of retransmission (Max_Reqln_Cnt), the dynamic access TN shall halt the dynamic access and report the failure status. If the TN actively leaves the network, it should apply to the MN to leave. When the MN confirms that the resource is recycled, it sends the updated state to all nodes on the network.

8.2.4.2.2 Primitive process

There are two cases for the data-link discovery management service primitive process, as shown in Figure 41 and Figure 42.

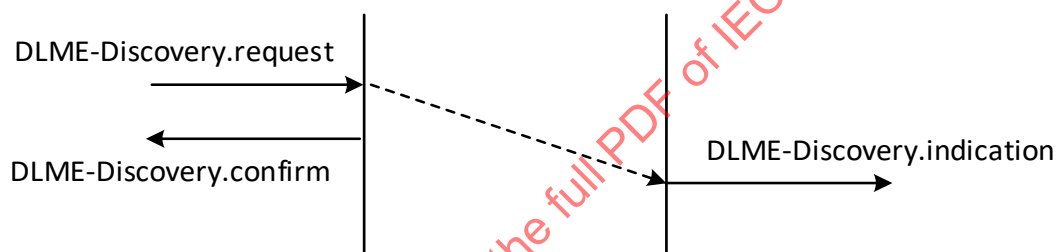


Figure 41 – TN active logout discovery service primitive process

Diagram description:

- For an active logout discovery service of the TN, the local DLMS-user sends a logout request to the local DLME, that is the confirmation is completed, the local DLME encapsulates the request DLPDU and sends it to the MN DLME.

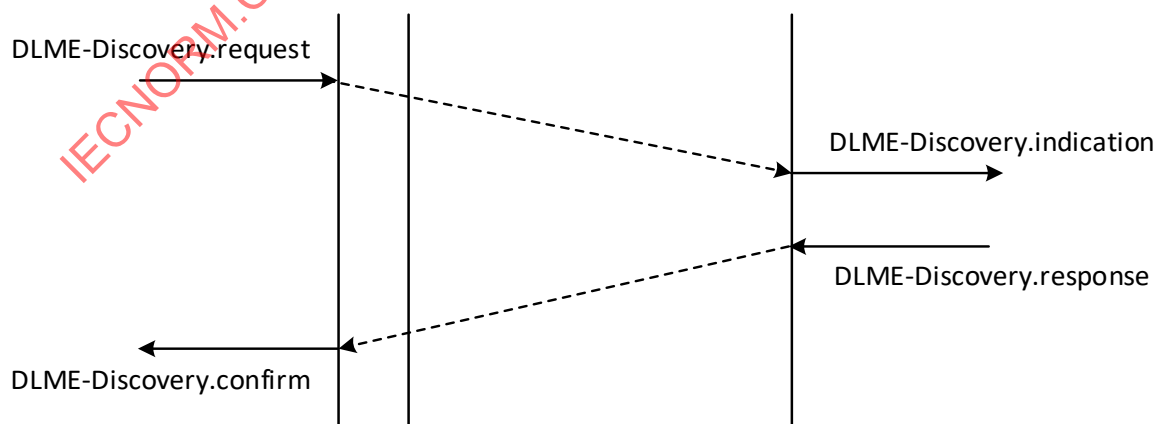


Figure 42 – Data-link discovery management service primitive sequence diagram

Diagram description:

- a) The synchronization operation to the network only for the confirmation primitive of this service is required, the request, indication and response primitives do not require synchronization.
- b) For the MN DLMS-user's request primitive operation, there are many request types, refer to the service interface description.

8.2.4.2.3 Service interface

On the MN or TN, the DLMS-users implement the functions for the data-link discovery management service based on the service interface.

DLME-Discovery (

Dst_NodeID,
AttrID,
Command,
NodeMgtDesc,
Mgt_Status

)

Table 67 indicates the primitives and parameters of this service. The parameters are described in Table 68.

Table 67 – Data-link discovery service primitives and parameters

Parameter name	Request	Indication	Response	Confirm
Dst_NodeID	M	M	M	M
AttrID	M	M	—	—
Command	M	M	M	M
NodeMgtDesc	M	M	—	—
Mgt_Status	—	—	M	M

Table 68 – Data-link discovery management service primitive parameter description

Parameter name	Data type	Input/output
Dst_NodeID	UINT16	IN
AttrID	UINT8	IN
Command	UINT8	IN
NodeMgtDesc	STRUCT	IN OUT
Mgt_Status	UINT32	OUT

Parameter description:

- Dst_NodeID: Range 0 to 65 535, the destination node address of the data-link discovery management service can be unicast, multicast, and broadcast type. See Table 44.
- AttrID: Range 0 to 255, the property ID, identify the properties of discovered management destination node:
 - 0: MN;
 - 1: TN;
 - Others reserved;
- Command: Range 0 to 255, discovery service command word:
 - 0: MN query TN status;

- 1: TN status query response;
 - 2: The TN requests access;
 - 3: MN response access;
 - 4: The TN confirms the access;
 - 5: The TN requests to leave;
 - 6: Configuration parameter update (limited to MN sending requests);
 - 7: Mapping table update (limited to MN sending requests);
 - 8: The MN updates the current active node state;
 - Others reserved;
- NodeMgtDesc: Node management information, see Table 69 for the NODE_MGT_INFO_S structure description.
- Mgt_Status: Range 0 to ($2^{32}-1$), discovery service status return value, See Table 80.

Table 69 – NODE_MGT_INFO_S structure description

Member name	Data type	description
NodeNum	UINT16	Number of nodes that need to be queried or applied
NodeState	UINT8	Node state
ServiceType	UINT8	Carried business data type, which can be planned and deployed in advance, ranges from 32 to 255.
MAC_Addr	ARRAY	The MAC address of discovery management service object node.
APPBW	UINT16	Application bandwidth, unit is octets, this field can be used to confirm whether allocable resources can be reused when applying for random access.
LoopTime	UINT16	Loop time of node, it is sent to MNs as reference information for allocation when applying for random access.

8.2.4.3 Maintenance

8.2.4.3.1 Service function description

The data-link maintenance management service provides local DLMS-user to maintain and manage the running state of the link node, including fault diagnosis, analysis and alarm processing of the local data-link node. The data-link maintenance management service manages the health status of the data-link node based on regular handshake and heartbeat packets, if there is an abnormal error in the link node, the link node shall notify the DLMS-user of the fault type, state code and determines whether to feed back to the MN according to the abnormal state. The MN maintains the state of all nodes in the network, the TN maintains the link state information of the local node and the node that there is cross communication with local node.

The local DLMS-user shall periodically or aperiodically send a maintenance request to the local DLME, and if the requested object is local node, the local DLME processes the request primitive, identifies the corresponding data item, and responds to the request. The local DLMS-user implements the maintenance verification operation based on the response of the local DLME. If the requested object is the remote node, the DLME encapsulates the request as a DLPDU and sends it to the remote node. After receiving the request DLPDU, the remote DLME shall deliver it to the DLMS-user for further processing according to the DLPDU content.

When a remote DLMS-user received a maintenance request, the corresponding remote DLME shall identify variable and encapsulate the request result into a response DLPDU, and it is sent to the DLME of the MN through the PhL service interface. The local DLME should send the response DLPDU immediately or wait for the MN to confirm according to the configuration, and the remote DLMS-user should confirm to receive the response and perform the confirm operation to complete this maintenance service.

8.2.4.3.2 Primitive process

The data-link maintenance service primitive process has local and remote operations, as shown in Figure 43 and Figure 44.

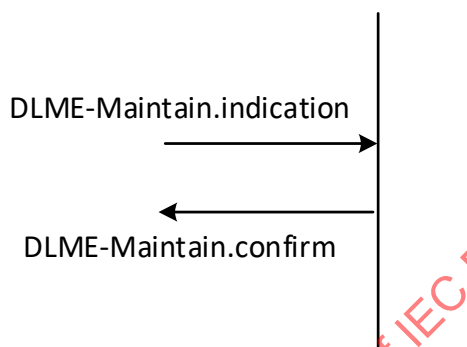


Figure 43 – Local data-link maintenance service primitive process diagram

Description: after the local DLME completes the identification primitive operation, that is, it completes confirmation processing for DLMS-users, then the status of the local maintenance service will be confirmed.

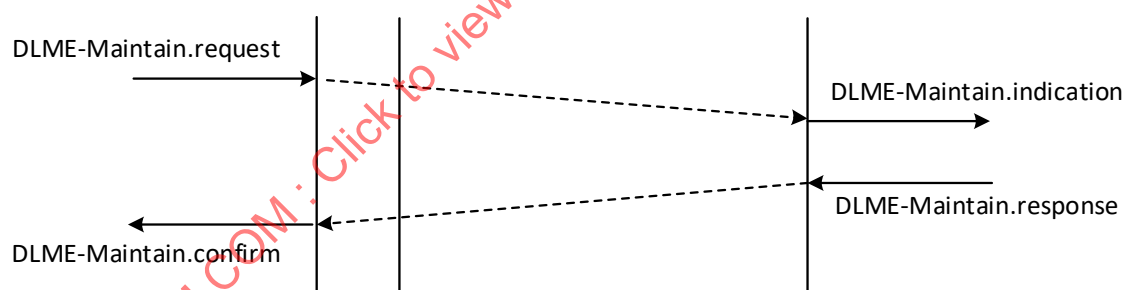


Figure 44 – Remote data-link maintenance service primitive process diagram

8.2.4.3.3 Service interface

Data-link maintenance service provides running state maintenance functions such as fault diagnosis and detection to local link nodes or remote link nodes.

```

DLME-Maintain (
    Dst_NodeID,
    Command,
    DiagInfoDesc,
    Mgt_Status
)
  
```

Table 70 indicates the primitives and parameters of this service. The parameters are described in Table 71.

Table 70 – Data-link maintenance service primitives and parameters

Parameter name	Request	Indication	Response	Confirm
Dst_NodeID	M	—	M	—
Command	M	M	M	M
DiagInfoDesc	M	M	M	—
Mgt_Status	—	M	—	M

Table 71 – Data-link maintenance service primitive parameter description

Parameter name	Data type	Input/output
Dst_NodeID	UINT8	IN
Command	UINT8	IN
DiagInfoDesc	STRUCT	IN OUT
Mgt_Status	UINT32	OUT

Parameter description:

- Dst_NodeID: Range 0 to 65 535, maintained node address, see Table 44.
- Command: Range 0 to 255, Command actions that can be performed to data-link node:
 - 0: Status query request;
 - 1: Status query response;
 - 2: Fault diagnosis request;
 - 3: Fault diagnosis response;
 - 4: Abnormal heartbeat packet;
 - 5: Heartbeat packet detection;
 - 6: Announcement broadcast;
 - Others reserved.
- DiagInfoDesc: Maintenance information description structures, different contents for different command actions, see Table 72 for DIAG_INFO_S structure description.
- Mgt_Status: Range 0 to ($2^{32}-1$), maintenance service status return value, see Table 80.

Table 72 – DIAG_INFO_S structure member parameter description

Member name	Data type	description
The information structure content with Command value is 0 ~ 5		
NodeState	UINT8	Node status information: 0: Initialization state 1: Normal operating state 2: Suspended state 3 ~ 255: Corresponding to various error states
NodeAttr	UINT8	Node control or terminal property information 0: MN; 1: TN Others reserved
HeartBeat	UINT16	Heartbeat count consistent with PhL cyclic symbol count

Member name	Data type	description
ChannelCnt	UINT16	Number of DTCs of node
WarnCnt	UINT32	Number of node alarm information
ErrCode	UINT32	Node abnormal error code
Reserved	UINT16	Reserved for protocol
The information structure content with Command value is 6		
TransmitMode	UINT8	System TM, see Table 62 for details
WorkMode	UINT16	Node PhL default working mode, see Table 62 for detailed
UnicodeNum	UINT8	The number of valid nodes which access to the network, see Table 62 for details
GroupNum	UINT8	Number of valid multicast groups in this network configuration, see Table 62 for details
Cycle_Frame_Num	UINT8	Number of cyclic frame cycles configured to process symbol resources, see Table 66 for details
R_RT_Attr	UINT16	RT properties of configured random access channel resources, see Table 66 for details
R_Band_Flag	UINT8	The upper or lower side band of the symbols based on the bit identification, see Table 66 for details
R_Band_Mode	UINT8	The carrier mode of the symbol resource block corresponding to the random-access channel, see Table 66 for details
R_Cycle_Symbol_Len	UINT16	The cyclic symbol cycle length of the random-access channel resource, see Table 66 for details
R_Symbol_Start	UINT16	Specify the starting symbol serial number of the reserved cyclic symbol period for the random-access channel, see Table 66 for details
R_Symbol_Num	UINT16	Specify the number of valid resource symbols reserved from R_Symbol_Start for random access channels, see Table 66 for details
Reserved	UINT16	Reserved fields.

8.2.4.4 Establish

8.2.4.4.1 Service function description

The data-link establish management service provides the establishment function of the DTC. The parameters of the service interface reflect the quality of this service. Each node maybe use one or more DTCs.

The DLMS-user of TN should send the establish request primitive to the DLME, and the DLME identifies and sets related variables or data based on the request primitive parameters. The DLME of the TN encapsulates the establish request as a DLPDU and send it to the MN to apply for the DTC. The DLME of the MN should send a response DLPDU to the TN after receiving the establish request. If the DLME of the TN does not receive the response DLPDU, it retransmits the request and the maximum number of retransmission (Max_Establish_Req_Retry_Cnt) is set when system is configured. The TN should confirm the result of establish for DTC according to the received response DLPDU from MN.

After the DLME of the MN or the TN receive the confirmation DLPDU from the DLME of MN, the result of the establish service should be determined according to the information of the DLPDU.

8.2.4.4.2 Primitive process

Figure 45 shows this service primitive process.

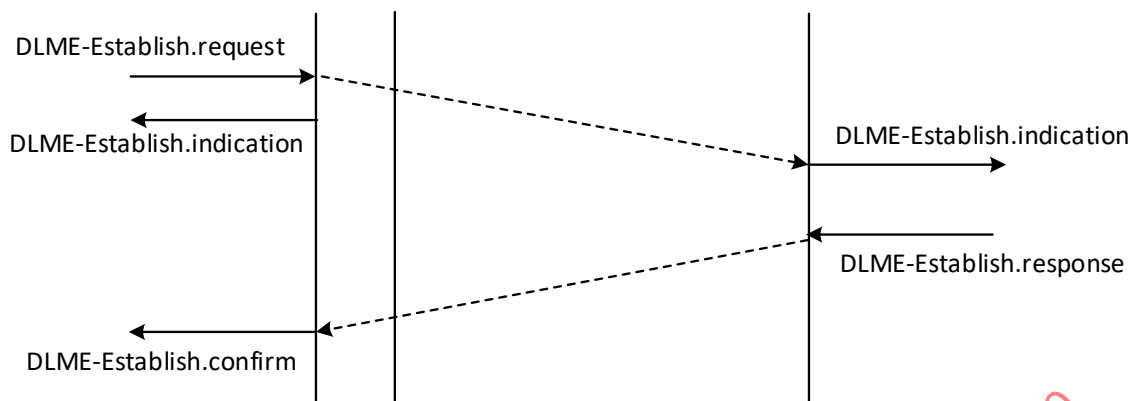


Figure 45 – Data-link establish management service primitive process

8.2.4.4.3 Service interface

The MN or the TN should implement the establishment of DTCs based on this service interface. At least one DTC should be established for each node on the AUTOBUS network.

DLME-Establish (

- Channel_ID,
- Dst_NodeID,
- ChResDesc,
- StateInfo,
- Mgt_Status

)

Table 73 indicates the primitives and parameters of this service. The parameters are described in Table 74.

Table 73 – Data-link establish management service primitives and parameters

Parameter name	Request	Indication	Response	Confirm
Channel_ID	M	M	M	M
Dst_NodeID	M	M	M	—
ChResDesc	M	M	—	—
Mgt_Status	—	—	M	M

Table 74 – Data-link establish management service parameter description

Parameter name	Data type	Input/output
Channel_ID	UINT16	IN
Dst_NodeID	UINT16	IN
ChResDesc	STRUCT	IN
Mgt_Status	UINT32	OUT

Parameter description:

- Channel_ID: Range 0 to 65 535, identify the DTC ID.
- Dst_NodeID: The node address for this channel to be established.
- ChResDesc: DTC related resource description, see Table 75 for CH_RES_INFO_S structure description.

- Mgt_Status: Range 0 to ($2^{32}-1$), establish service status return value. See Table 80.

Table 75 – CH_RES_INFO_S structure member description

Member name	Data type	description
Sequence_No	UINT8	Channel resource block serial number, (Each channel resource block consists of 1 to 2 symbol resource blocks, starting with the symbols specified by the Symbol_Start, the contiguous Symbol_Num symbols form a symbol resource block), when establishing a valid DTC, there should be a valid channel resource, and the valid range of the serial number of the channel resource is 1 ~ 255, 0 is an invalid.
Service_Type	UINT8	Data type carried by channel distinguishes the priority of different service carried by channel, the smaller value has higher priority.
RxTx_Attr	UINT8	Data transmitting and receiving property of bus: 0: Receive data from the bus 1: Transmit data to the bus Others reserved
RT_Attr	UINT16	Real time attribute of channel resources: 0: Once the channel is established, the corresponding resource will always be used by the node to transfer data and will not be released. 1 ~ 512: The channel duty-cycle operation times in the current configuration cycle, that is the corresponding resource block duty-cycle operation times, the maximum value is 512, other values are not supported. For example, 2 indicates that the physical symbol resource block corresponding to the channel will be released after two consecutive uses. Others are invalid. The RT_Attr values for two symbol resource blocks should be the same.
Cycle_Symbol_Len	UINT16	The cyclic symbol cycle length of the channel resource, that is, after the channel is established, from the OFDM symbol specified by the Symbol_Start, the Cycle_Symbol_Len symbol cycle is processed once. Valid range is 1 ~ 512, other values are invalid. Cycle_Symbol_Len values should be greater than or equal to Symbol_Num_1 or Symbol_Num_2. The Cycle_Symbol_Len values of all Sequence_No with the same Channel_ID are the same; the Cycle_Symbol_Len values of the two symbol resource blocks with the same Sequence_No are the same.
Band_Flag_1	UINT8	The upper or lower side band of the first symbol resource block based on the bit identification; bit0: upper-side band, bit1: the lower-side band, the other bits are invalid; and the corresponding bit set to 1 indicates that the symbol resource block uses the corresponding side band symbol.
Band_Mode_1	UINT8	Specifies the working mode for the first symbol resource block, valid value range 0 ~ 21. Other values are not valid.
Symbol_Start_1	UINT16	Start symbol sequence number of the first symbol resource block.
Symbol_Num_1	UINT16	Number of symbols continuously occupied by the first symbol resource block from Symbol_Start_1.
Band_Flag_2	UINT8	The upper or lower side band of the second symbol resource block based on the bit identification; bit0: upper-side band, bit1: the lower-side band, the other bits are invalid; and the corresponding bit set to 1 indicates that the symbol resource block uses the corresponding side band symbol.
Band_Mode_2	UINT8	Specifies the working mode for the second symbol resource block. , valid value range 0 ~ 21. Other values are not valid.
Symbol_Start_2	UINT16	Start symbol sequence number of the second symbol resource block.
Symbol_Num_2	UINT16	Number of symbols continuously occupied by the second symbol resource block from Symbol_Start_2.

8.2.4.5 Release

8.2.4.5.1 Service function description

When the DLMS-user confirms that the established DTC should no longer be used, it should send release request to the DLME of MN. After the DLME of MN receives the request DLPDU, the variable of the DTC to be released should be indicated and the related resources should be recycled. The requester of this service confirms the request primitive operation successful once the request DLSDU has been sent. The DLME of MN should carry out indication primitive successfully and the response to requester is not required.

8.2.4.5.2 Primitive process

Figure 46 shows this service primitive process.

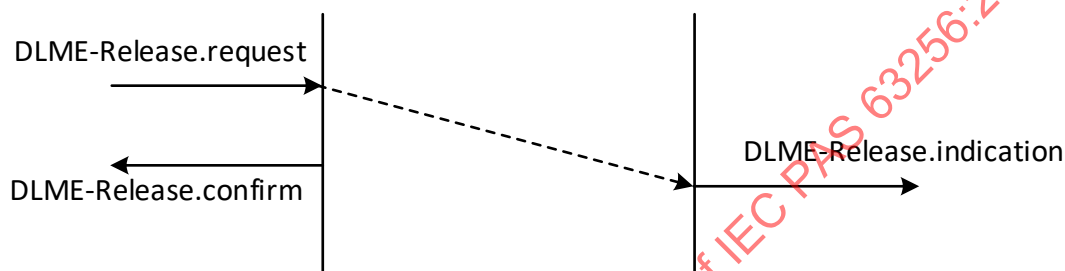


Figure 46 – Data-link release management service primitive sequence diagram

8.2.4.5.3 Service interface

The MN and the TN should use this service interface to release the designated DTC resources. Once the service is finished, the resource of this DTC could be reallocated for any nodes on this AUTOBUS network again.

```

DLME-Release (
    Channel_ID,
    Dst_NodeID,
    ChResDesc,
    Mgt_Status
)
    
```

Table 76 indicates the primitives and parameters of this service. The parameters are described in Table 77.

Table 76 – Data-link release management service primitives and parameters

Parameter name	Request	Indication	Confirm
Channel_ID	M	M	M
Dst_NodeID	M	M	—
ChResDesc	M	M	—
Mgt_Status	—	—	M

Table 77 – Data-link release management service primitive parameter description

Parameter name	Data type	input/output
Channel_ID	UINT16	IN
Dst_NodeID	UINT16	IN
ChResDesc	STRUCT	IN
Mgt_Status	UINT32	OUT

Parameter description:

- Channel_ID: Range 0 to 65 535, the DTC ID to be released.
- Dst_NodeID: The node address for the channel to be released.
- ChResDesc: Descriptor information of DTC related resource to be released, see Table 75 for details of CH_RES_INFO_S structure members.
- Mgt_Status: Range 0 to $(2^{32}-1)$, release service status return value. See Table 80.

8.2.4.6 Update

8.2.4.6.1 Service function description

The DLMS-user should update the parameters of the established DTC based on the update service.

The DLMS-user sends an update request to the DLME, the request DLSDU should contain the DTC channel ID and the parameter information that needs to be updated. After the DLME of MN receives the request, it should identify related variable of the DTC to be updated and encapsulate a DLPDU to send to the remote TN. The DLME of TN should carry out indication operation to update the parameter of the DTC and the operation result should be encapsulated into the response DLPDU to be transmitted to DLME of MN. The DLME of MN should confirm the service status according to the response DLPDU. If the DLME of MN does not receive the response DLPDU from the DLME of TN, it should retransmit the update request DLPDU and the maximum number of retransmission (Max_Update_Req_Retry_Cnt) has been set when the system is configured.

The DLME of TN or MN maybe set to do not respond to the update request according to the requirement of DLMS-user. Based on this setting, the DLME should analyse the received update request DLPDU, confirms whether it is necessary to update the local variables or parameters, and confirms that the update service is completed without concern for the processing result of the DLPDU. When the DLME carry out update indication primitive for the related parameters of the DTC to be update, it should perform protection operations to prevent data loss or damage.

8.2.4.6.2 Primitive process

Figure 47 shows this service primitive process.

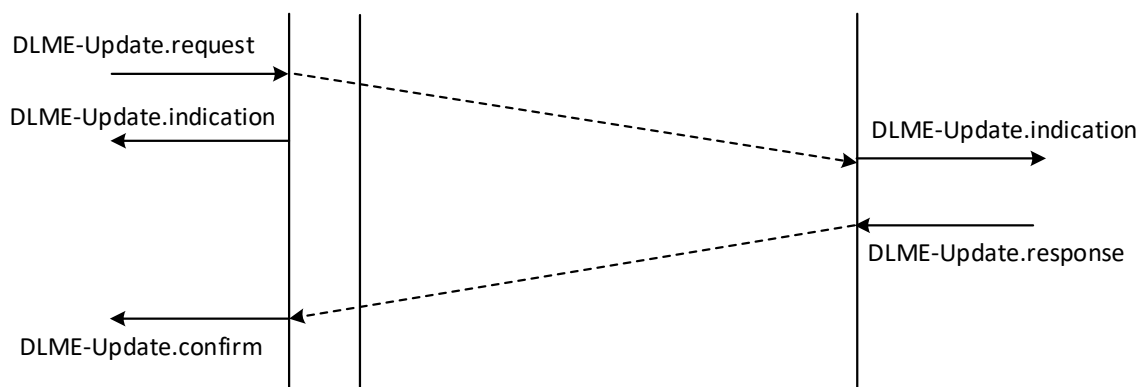


Figure 47 – Data-link update management service primitive sequence diagram

NOTE The DLME may not respond to the update service request operation, the DLMS-user may not need confirmation operation processing.

8.2.4.6.3 Service interface

The DLME of MN or the TN to update the related parameters of the designated DTC based on this service interface.

DLME- Update (
 Channel_ID,
 Dst_NodeID,
 ChResDesc,
 UpdateMode,
 Mgt_Status
)

Table 78 indicates the primitives and parameters of this service. The parameters are described in Table 79.

Table 78 – Data-link update management service primitives and parameters

Parameter name	Request	Indication	Response	Confirm
Channel_ID	M	M	M	M
Dst_NodeID	M	M	M	—
ChResDesc	M	M	—	—
UpdateMode	M	M	—	—
Mgt_Status	—	—	M	M

Table 79 – Data-link update management service primitive parameter description

Parameter name	Data type	Input/output
ChannelID	UINT16	IN
Dst_NodeID	UINT16	IN
ChResDesc	STRUCT	IN
UpdateMode	UINT8	IN
Mgt_Status	UINT32	OUT

Parameter description:

- ChannelID: Range 0 to 65 535,DTC ID to be updated

- Dst_NodeID: The node address of the channel to be updated.
- ChResDesc: Descriptor information of DTC related resource to be updated, see Table 75 for details of CH_RES_INFO_S structure members.
- UpdateMode: Range 0 to 255, update mode which is explained as follows:
 - 0: complete cover and lock protection;
 - 1: complete cover and ignore lock protection;
 - 2: only update parameter and lock protection;
 - 3: only update parameter and ignore lock protection;
 - Others reserved.
- Mgt_Status: Range 0 to ($2^{32}-1$), update service status return value. See Table 80.

Table 80 – DLMS status return value description

Status name	Status coding	Status description
DLMS_SUCCESS	0x200	DLMS processing success
DLMS_FAILED	0x201	DLMS processing failed
DLMS_ERR_PARA	0x202	DLMS parameter errorc
DLMS_ERR_DATA	0x203	DLMS data errorc
DLMS_ERR_CRC	0x204	DLMS data CRC error
DLMS_DLME_NO_EXIST	0x205	DLME do not exist
DLMS_NO_RESOURCE	0x206	DLMS resource is not enough
DLMS_CLASH_CONFIG	0x207	DLMS config clash
DLMS_NOT_SUPPORT	0x208	Current DLMS not supported
DLMS_ERR_UNKNOWN	0x20F	DLMS unknown error

8.2.5 Clock synchronization service description

8.2.5.1 Function description

AUTBUS adopts the communication technology based on OFDM to implement accurate frequency synchronization. The delay measurement mechanism provided on DLL used to implement accurate phase synchronization. The frequency synchronization and the phase synchronization ensure the implementation of clock synchronization.

DLCSS implement delay measurement, clock synchronization and clock interrupt based on the PhL service interface. In order to achieve clock synchronization between nodes, the DLCSS-user should register the service and set the relevant parameters for the DLCE to further meet the time sensitive service requirements. This service also provides clock interrupt signal to the DLS-user, which can further meet the RT data service processing with cyclic or acyclic. The clock synchronization service and the delay measurement service are available for all nodes on the network, but the clock interrupt service is a local service.

8.2.5.2 Delay measurement

The delay measurement service implements the measurement of the transmission delay between the MN and the TNs. After the AUTBUS network initialization is complete, there is only time deviation and transmission delay information between the MN and the TNs. The MN should update the current delay information by sending a broadcast delay measurement request and each TN should respond to this request primitive. When the TN applies for delay measurement service, the MN should send the stored delay information to the corresponding TN. See 8.3.1.4 for working procedure of this service.

The DLCSS-user initiates the request primitive operation based on the delay measurement service interface and implements the local measurement confirmation operation. After the remote DLCSS-user receives the measurement request, the local related parameters of DLCE are identified and updated, and immediately the indication result should be responded to the requester of the current delay measurement service. The DLCSS of AUTOBUS network is always based on the MN, so the time deviation of the time delay measurement is always the delay information of the node relative to the MN.

8.2.5.2.1 Primitive process

Figure 48 shows this service primitive process.

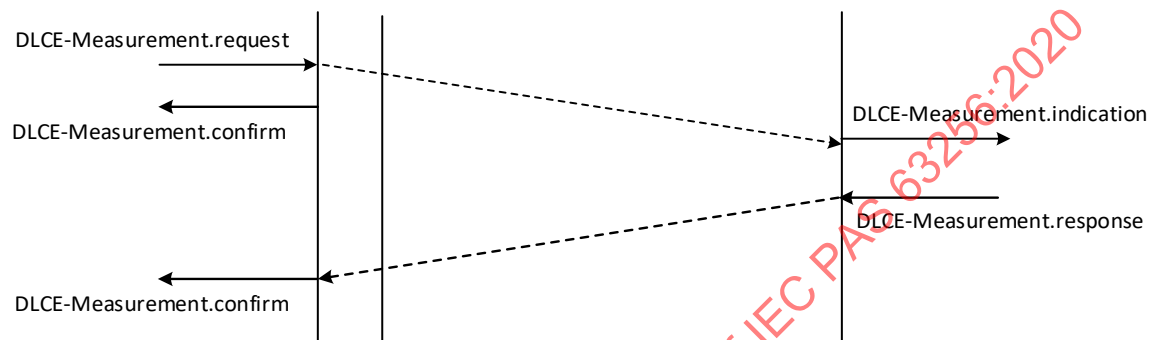


Figure 48 – Delay measurement service primitive process

8.2.5.2.2 Service interface

```
DLCE-Measurement (
    Dst_NodeID,
    Src_NodeID,
    Sequence,
    Action,
    Result,
    Status
)
```

Table 81 indicates the primitives and parameters of this service. The parameters are described in Table 82.

Table 81 – Delay measurement service primitives and parameters

Parameter name	Request	Indication	Response	Confirm
Dst_NodeID	M	M	M	M
Src_NodeID	M	M	M	—
Sequence	M	M	M	M
Result	—	—	M	M
Status	—	—	M	M

Table 82 – Delay measurement service primitive parameter description

Parameter name	Data type	Input/output
Dst_NodeID	UINT16	IN
Src_NodeID	UINT16	IN
Sequence	UINT16	IN
Action	UINT8	IN
Result	UINT64	OUT
Status	UINT32	OUT

Parameter description:

- Dst_NodeID: Range 0 to 65 535, the address of the measurement service destination node can be unicast, multicast or broadcast type, see Table 44.
- Src_NodeID: Range 0 to 65 535, the address of measurement service source node, by default the node address of this node.
- Sequence: Range 0 to $(2^{16}-1)$, each time delay measurement request corresponds to a sequence number, the corresponding TN responds to the MN with this sequence number, and the MN also corresponds to the delay information of the TN with this sequence number.
- Action: Range 0 ~ 255, delay measurement action identification:
 - 0x00: Delay measurement request;
 - 0x01: Delay measurement identification;
 - 0x02: Delay measurement response processing;
 - 0x04: Delay measurement confirmation operation;
 - Others reserved.
- Result: 64-bits integer value, Range is 0 to $(2^{64}-1)$, the output delay measurement results are 32 bits average delay and 32 bits current delay respectively.
- Status: Range 0 to $(2^{32}-1)$, status return value of the delay measurement service. See Table 87.

NOTE The parameter Result has different values at different phases of delay measurement, for request and indication primitive operation, the result as output value may be null. For the response primitive, the Result is the time information of the requested TN, for the confirmation primitive, the Result is the time delay value between the MN and the corresponding TN.

8.2.5.3 Clock synchronization

On the AUTBUS network, the clock and frequency information of the MN as reference standard, the clock and frequency information of all TNs need to be synchronized with the MN's. When the TN initiates a clock synchronization service request, the DLCE of MN should encapsulate the synchronization information of the current PhL SF into a DLPDU and send it to the remote TN by means of response primitive. The DLCE of TN should update the local clock information according to the locally saved relative to MN delay information and the time deviation value in the DLPDU, which can implement the clock synchronization process with the MN.

The DLCSS-user of TN performs the clock synchronization request primitive operation, and the DLCE should encapsulate the request information into a DLPDU. The encapsulated DLPDU should be sent to the DLCE of MN DLDE through the corresponding PhL service interface. After receiving the request DLPDU, the DLCE of MN should encapsulate the start accurate synchronization information of the PhL current SF into a DLPDU to respond to the TN. If the DLCE of TN receives the response DLPDU, it updates the local clock information and confirms the status of this service primitive processing. When the MN initiates the clock synchronization service request, the start accurate synchronization of the current PhL SF is encapsulated in a DLPDU and sent to all TNs. After receiving this DLPDU, the TN utilizes the locally stored delay

information and transfer difference information, calculates and updates local time, completes this service processing.

8.2.5.3.1 Primitive process

Figure 49 shows this service primitive process.

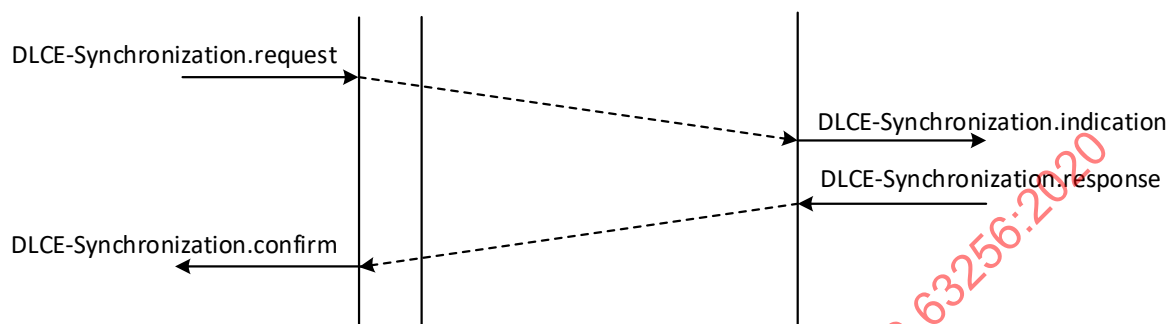


Figure 49 – Clock synchronization service primitive process

8.2.5.3.2 Service interface

DLCE- Synchronization (

- Dst_NodeID,
- Src_NodeID,
- TimeInfo_Senond,
- TimeInfo_Ns,
- Precision,
- Action,
- Status

)

Table 83 indicates the primitives and parameters of this service. The parameters are described in Table 84.

Table 83 – Clock synchronization service primitives and parameters

Parameter name	Request	Indication	Response	Confirm
Dst_NodeID	M	—	M	M
Src_NodeID	M	M	M	M
TimeInfo_Senond	M	M	M	—
TimeInfo_Ns	M	M	M	—
Precision	M	M	—	—
Action	M	M	M	M
Status	—	—	M	M

Table 84 – Clock synchronization service primitive parameter description

Parameter name	Data type	Input/output
Dst_NodeID	UINT16	IN
Src_NodeID	UINT16	IN
TimeInfo_Second	UINT64	IN
TimeInfo_Ns	UINT64	IN
Precision	UINT8	IN
Action	UINT8	IN
Status	UINT32	OUT

Parameter description:

- Dst_NodeID: Range 0 to 65 535, the destination node address of synchronous processing can be unicast, multicast or broadcast type, and the specific value refer to Table 44
- Src_NodeID: Range 0 to 65 535, the source node address of synchronous processing.
- FrameID_S: The minimum 8bit value for the current SF ID of the trigger clock synchronization service.
- TimeInfo_Second: Range 0 to $(2^{48}-1)$, time information above the second of the RT clock at the start of the current SF, second as unit.
- TimeInfo_Ns: Range 0 to $(2^{48}-1)$, time information below the second order of the RT clock at the start of the current SF, ns as unit.
- Precision: Range 0 to 255, synchronous source device (MN) clock accuracy information. The unit is ns.
- Action: Range 0 to 255, clock synchronization actions, including read or write time information, etc.
 - 0x00: Clock synchronization request;
 - 0x01: Clock synchronization identification;
 - 0x02: Clock synchronous response;
 - 0x04: Clock synchronization confirmation;
 - Others reserved.
- Status: Range 0 to $(2^{32}-1)$, status return value of the synchronization service. See Table 87.

8.2.5.4 Clock interrupt

AUTBUS DLCSS provides clock interrupt function based on PhL symbol resources to meet different requirements of users, and provides interrupt signal with minimum granularity is an OFDM symbol cycle, DLS-users could implement real-time cyclic (RTC) or real-time acyclic (RTA) service data processing based on the clock interrupt provided by DLCE. On AUTBUS network, the clock interrupt is off by default, which needs to be turned on when configure the system and different clock interrupt signals should be set according to the requirements of the service.

PhL symbol generates interrupt signal according to set time to inform DLCSS-user to carry on identification operation. DLDE and DLME of DLL can implement related RT service function based on this identification result.

8.2.5.4.1 Primitive process

Figure 50 shows this service primitive process.

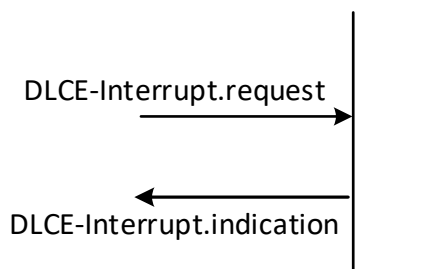


Figure 50 – Clock interrupt service primitive process

8.2.5.4.2 Service interface

The DLCSS-user completes the clock interrupt service registration through this service interface. After the DLCSS-user completed the registration, the PhL entity should generate the interrupt signal at the determined time based on the set type, and notify the DLCSS-user. The DLS-user could process different RTC or RTA services by judging the interrupt signal.

```

DLCE-Interrupt (
    IntType,
    IntUnit,
    IntCount,
    Status
)
    
```

Table 85 indicates the primitives and parameters of this service. The parameters are described in Table 86.

Table 85 – Clock interrupt service primitives and parameters

Parameter name	Request	Indication
IntType	M	M
IntUnit	M	—
IntCount	M	—
Status	—	M

Table 86 – Clock interrupt service primitive parameter description

Parameter name	Data type	Input/output
IntType	UINT8	IN
IntUnit	UINT8	IN
IntCount	UINT8	IN
Status	UINT32	OUT

Parameter description:

- IntType: Range 0 to 255, defined interrupt type by system or user, in which:
 - 0: No interruption;
 - 1: Single real time interrupt;
 - 2: Single ofdm symbol interrupt;
 - 3: Continuous real time interrupt;

- 4: Continuous OFDM symbol interrupt;
 - Other values are invalid.
- IntUnit: A unit of time interruption, in which:
- 1: Real Time is output interrupt for the integer hour;
 - 2: Real Time is output interrupt for the integer minute;
 - 3: Real Time is output interrupt for the integer second;
 - 4: Real Time is output interrupt for the integer millisecond;
 - 5: Real Time is output interrupt for the integer microsecond;
 - 6: OFDM symbol start time output interrupt;
 - 7: SF start time output interrupt;
 - 8: Config cycle start time output interrupt;
 - Other values are invalid.
- IntCount: Number of interrupts in continuous interruptions, valid values is 0 ~ 255, 0 means no output.
- Status: range 0 to ($2^{32}-1$), status return value of the clock interrupt. See Table 87.

Table 87 – DLCSS status return value description

Status name	Status coding	Status description
DLCSS_SUCCESS	0x400	DLCSS processing succeeded
DLCSS_FAILED	0x401	DLCSS processing failed
DLCSS_ERR_PARA	0x402	DLCSS parameter error
DLCSS_ERR_TIME	0x403	DLCSS time error
DLCSS_ERR_CRC	0x404	Data for DLCSS with incorrect CRC value
DLCSS_NO_CLOCK	0x405	No clock synchronization entity clock
DLCSS_NO_RESOURCE	0x406	DLCSS resource is not enough
DLCSS_CLASH_SYNC	0x407	DLCSS synchronization clash
DLCSS_NOT_SUPPORT	0x408	Current DLCSS operation is not supported
DLCSS_ERR_UNKNOWN	0x40F	Unknown error in DLCSS

8.3 DLL protocol

8.3.1 Working procedure

8.3.1.1 Initialization procedure

8.3.1.1.1 Basic configuration

On the AUTOBUS network, the MN confirms the NodeID resource planning and sends the network configuration to all the TNs. At the same time, the SF sequence number is synchronized. The TN receives the numbers of global NodeID nodes, its own NodeID identification, and the SF sequence number information. After determining its resources, TN will send the access notification DLPDU to the MN. If the DLL does not complete the basic configuration in one allocation cycle, the basic configuration will continue in the next allocation cycle.

The MN sends the global basic configuration DLPDU to each TN through the downlink subframe. Each TN shall receive the configuration DLPDU and response access notification to MN through available physical symbol resource. This procedure duration is N frame cycles (N is system configuration parameter, and default value is less than or equal to 8). After this time expires and the corresponding bit of Node status is set, the system automatically enters the next phase.

After the basic configuration is completed, the NodeID of all nodes and NodeID and MAC mapping table is ready. Clock of all nodes are synchronized.

8.3.1.1.2 Resource mapping configuration

After the MN completes the basic configuration of the system, each node completes the resource mapping of the DTC by DLME-establish service.

Data buffer processed by DLL distinguishes RT data buffer from nRT data buffer. RT data buffer is used to process service data with high RT requirement, fixed packet length and fixed cycle period; nRT data buffer is used to process low RT requirement, unfixed packet length and cycle. Data in the RT data buffer needs to be mapped to the determined physical symbol resource through DTC for cyclic processing because of the determined cycle and the fixed packet length; however for the nRT data, since the RT requirement of transmission is not high, and the length of the packet data to be processed is not fixed, using the infinite loop DTC that has been applied for and obtained, or applying for and using the time-limited loop DTC or temporary DTC. The physical symbol resources are dynamically applied for before the data transmission, and the PhL symbol resources are scheduled and allocated according to the service data types that need to be transmitted. This physical symbol resource does not conflict with the physical symbol resources used by the RT data buffer. The resource mapping configuration diagram is shown in Figure 51.

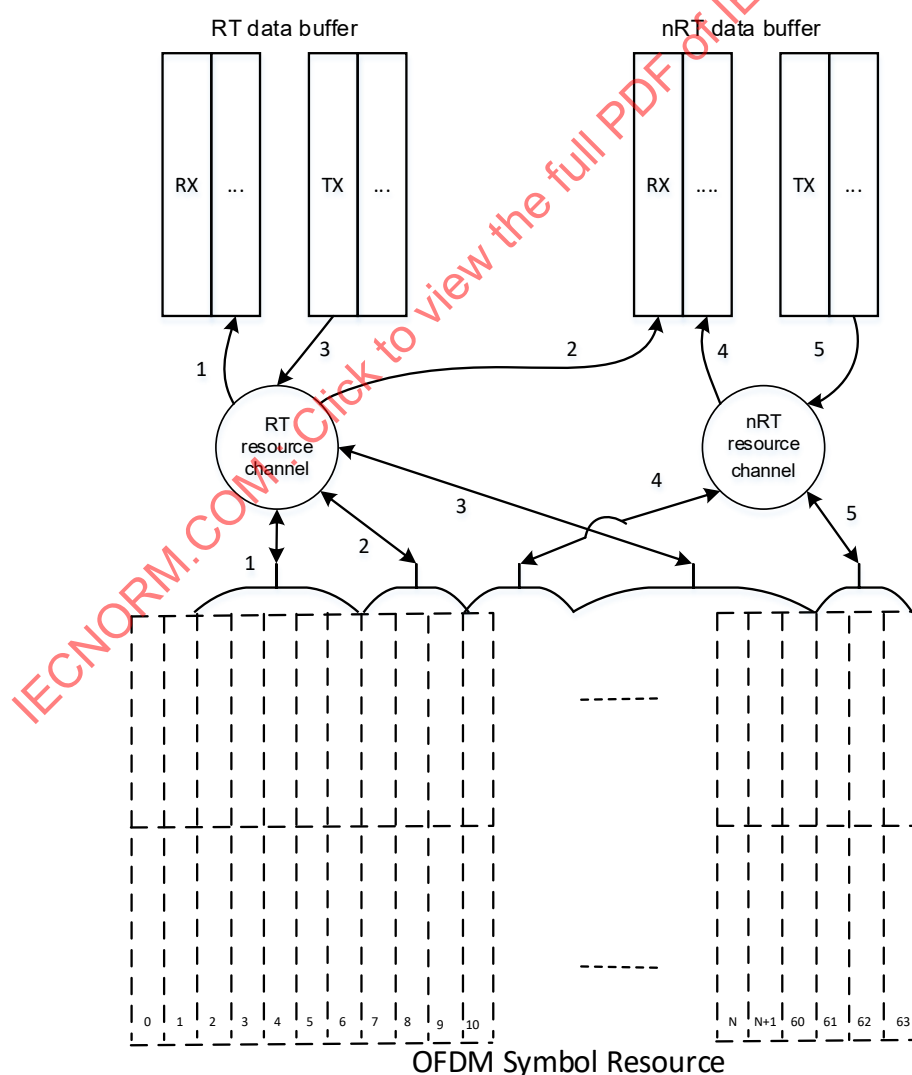


Figure 51 – Resource mapping configuration diagram

The AUTOBUS network MN performs resource mapping configuration immediately after basic configuration negotiation processing. After the completion of resource mapping configuration, all TNs send the final access notification to the MN, the MN confirms whether the corresponding TN has completed its basic configuration and resource mapping phase or not. When MN confirms all the TNs' access notification, and the network system enters the running phase and starts to process the data transmission.

The initialization procedure of the AUTOBUS network DLL is shown in Figure 52:

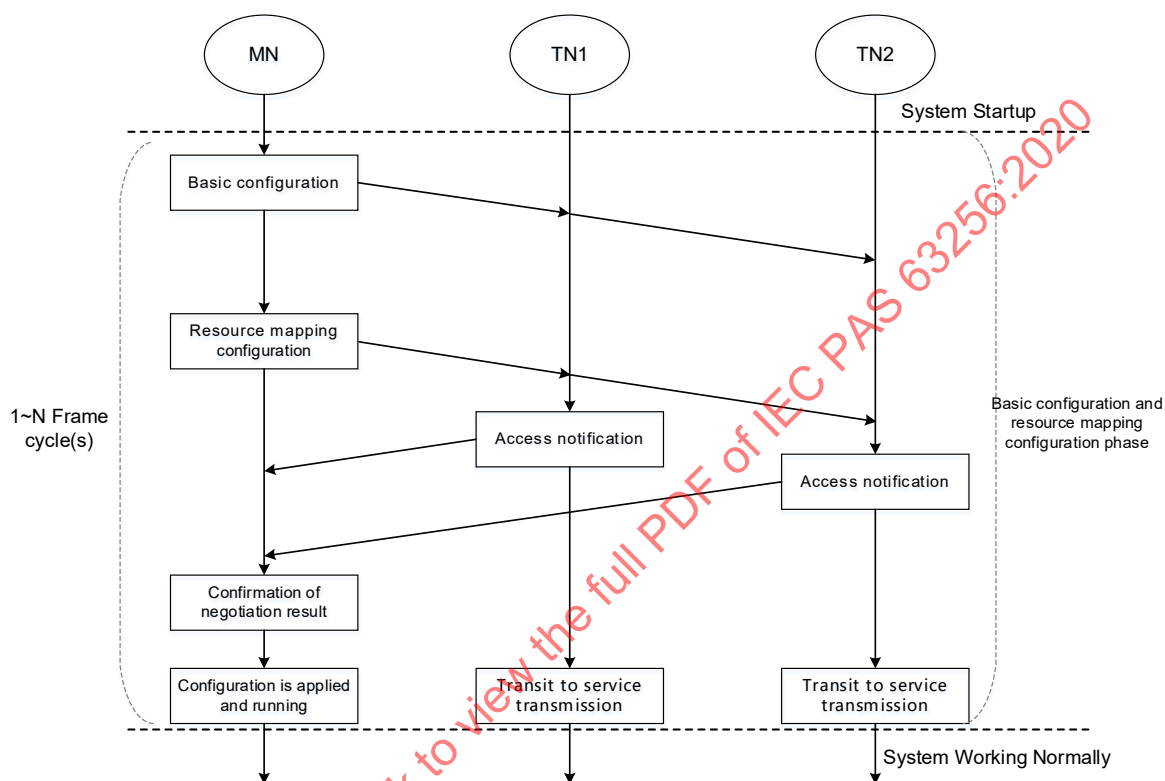


Figure 52 – Initial access configuration procedure diagram

Procedure description:

- Basic configuration: MN configure transmit mode, working mode and address of the TNs.
- Resource mapping configuration: Each node's configuration for PhL symbol resources is updated to each node after the resource mapping configuration is finished.
- Access notification: After the node confirms the basic configuration, address assignment and resource mapping configuration. Access notification has been sent to the MN to confirm that the node is authorized to access network.
- Configuration is applied and working: the MN confirms that the configuration of all TNs is successful and switch to running working mode.
- Transit to service transmission: the TN confirms that the configuration is correct and switch to normal working mode, and carries on service data transmission.

8.3.1.2 DLL node management procedure

8.3.1.2.1 DLL maintenance

In the running phase of AUTOBUS network, the DLMS-user of MN uses the downlink sub-frame to detect the heartbeat of the nodes in the configured cyclic frame period, node with abnormal heartbeat detection, responds to the MN and reports the status and heartbeat information of the node. The MN will broadcast the announcement DLPDU to all nodes using the downlink sub-frame mapping channel, for the random access node, the node random access can be implemented by parsing the broadcast DLPDU. The sending cycle of heartbeat detection DLPDU is set by the user when the system is configured, and the announcement DLPDU is processed by the system in default, and cyclicly sent on the channel of downlink subframe.

8.3.1.2.2 Node join

After receiving the basic configuration and resource mapping of the MN, all the TNs in the AUTOBUS network complete the initialization of the configuration and transit to the running node. There are two ways for TNs to access the network: normal access and random access. For nodes that MNs have correctly configured and allocated valid resources, the mode of accessing the network in the initialization phase or the running phase of the network is normal access. For nodes that MNs have not correctly configured and allocated valid resources, the mode of accessing the network during the running phase of the network is random access. In the initialization phase of network, because the network is not stable, the MN does not accept the random access, and the random access node should wait for the system running normally, then sending the random access request.

For the normal access of the TN, referring to the initial access configuration, the TN confirms that it has received the basic configuration, address assignment and resource allocation information, and sends an access notification DLPDU to the MN, confirming that the TN is correctly accessed to the network, as shown in Figure 52.

In the running phase of network, the new access node is also called random access node. The random access node can confirm the basic configuration information of the system network based on the DLME-Discovery service, and send the access request on the random access resource allocated by the MN. The random access procedure is shown in Figure 53:

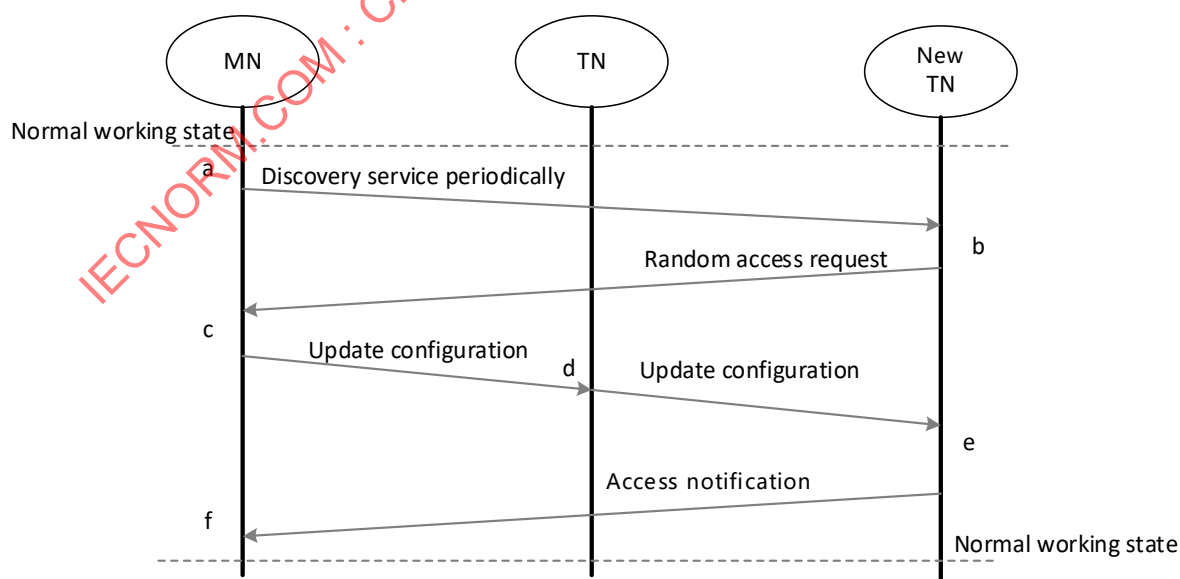


Figure 53 – The random access configuration procedure diagram

Procedure description:

- a) MNs periodically send control information packet including network announcement to all nodes, all TNs including random access nodes can receive this packet.
- b) The random access node analyses the announcement of the network sent by MN, and sends the access request to the MN based on the random access symbol resource.
- c) After receiving the request information packet of the random access node, the MN analyses and confirms the relevant information, completes the identification and security authentication, assigns the address and the resource to the random access node.
- d) The MN sends the updated address and resource allocation information to the network, and other running TNs refresh the local configuration information and continue to work.
- e) After receiving the configuration information sent by the MN, the random access node updates the local configuration information and sends the confirmed access notification to the MN after confirming that both the basic configuration and the resource allocation are correct, and transit to the running state.
- f) According to the confirmed access information of the random access node, the MN continues to update the local configuration table and information, completes the joining process of the random access node, and manages the new node regularly.

8.3.1.2.3 Node query

Based on the bus topology, all nodes including MNs, can receive information DLPDUs from all nodes. Using this feature, the MNs in AUTOBUS network can query and monitor the status and information of any TN. If the MN does not receive the information DLPDU sent by a TN within a certain period of time, it should send a query DLPDU to the TN to confirm whether it is still in the network and run normally. If the node does not respond to the query request within the timeout limit, it is considered to have failed or has left the network.

For the query DLPDU sent by the MN, all the nodes will receive it. If the nodes of the query do not include this node, the receiving node does not respond to the access notification, but shall continue to listen to the access notification DLPDU of the queried node. If MN does not receive an access notification DLPDU of response from the queried node during the timeout (which is set when the system is configured), then other nodes should delete the non-responsive nodes from their configuration information table according to the configuration update DLPDU published by the MN to complete the online network node configuration synchronous update.

8.3.1.2.4 Node leave

Based on whitelist management rules, a node is still a legitimate device before it leaves, so the fixed resources such as NodeID address resources pre-allocated to the node, are not recovered. However, the resources for dynamic application by the leaving node because of the service data transmission shall be recovered normally at the end of the service transmission, which does not affect the data transmission of other nodes in the network.

Node leave procedure is distinguished with active mode and passive mode. Active mode means that a node should send a request to the MN to leave the network before exiting the network, and at the same time actively release related resources, or the node does not send a packet to the MN due to device exception or error to leave the network, at this time, the MN should also confirm that the node has exited the network based on the maintenance service, and the MN will update the network mapping table and related configuration information, and synchronize to all the nodes online on the network. For passive mode, the MN notifies the TN to leave the network, according to the difference of the command word given by the MN, the TN can choose to restart the device, reset all basic configuration and resource mapping parameters, or keep silent without sending or receiving any service data. Passive mode node leave mechanism is usually used in network dynamic configuration or online upgrade, application scenarios, etc.

The node leave procedure is shown in Figure 54.

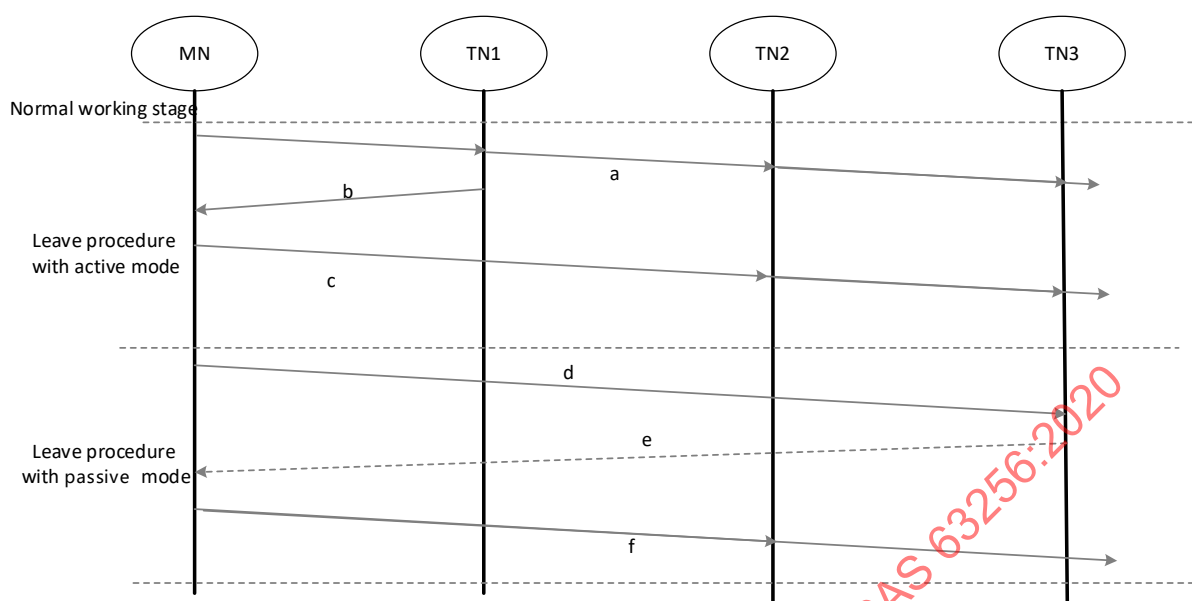


Figure 54 – Node leave procedure diagram

- The MN confirms whether all nodes in the network are working normally based on periodic maintenance service and the current network operation status.
- The TN2 sends the leave network request to the MN, and the TN2 actively releases the related resources.
- The MN receives the request from the TN2, updates the mapping table and configuration information saved by this node, confirms the recovery of related resources, and synchronizes the updated network and resource allocation information to other TNs on the network.
- The MN notifies the TN3 to leave the current network, and the MN updates the mapping table and configuration information saved by the node, and recovers the related resources.
- After receiving the request of the MN, the TN3 releases the resource and decides whether to restart or keep silent according to the command. At the same time, the TN responds to the MN according to the command to confirm leave from the network.
- The MN synchronizes the updated network and resource allocation information to all the nodes of the network again.

8.3.1.3 Data transmission procedure

After the AUTOBUS network confirms that all TN basic configuration and resource mapping is finished, the network can work correctly based on configuration to data transmission. AUTOBUS fieldbus and other fieldbus can process the same type of service data, according to the RT and nRT service data, correspondingly the packet data buffers of the DLL also have two types with RT and nRT.

When the data in RT data buffer is processed in the DLL service data sending phase, the data is encapsulated and sent to the network based on the DTC that has been configured or applied. The transmission of nRT data need to dynamically apply for DTC, based on the successful applied physical symbol resources and the established channels, the data is encapsulated and sent to the network. For nRT data resource applications, the scheduling is based on priority, and the data types with high priority are scheduled first. For nRT data, the resource can also be applied for once and a determined channel can be established at the system initialization phase. If the user does not actively release the resource, it can be used all the time. The DLL service data transmission procedure is shown in Figure 55.

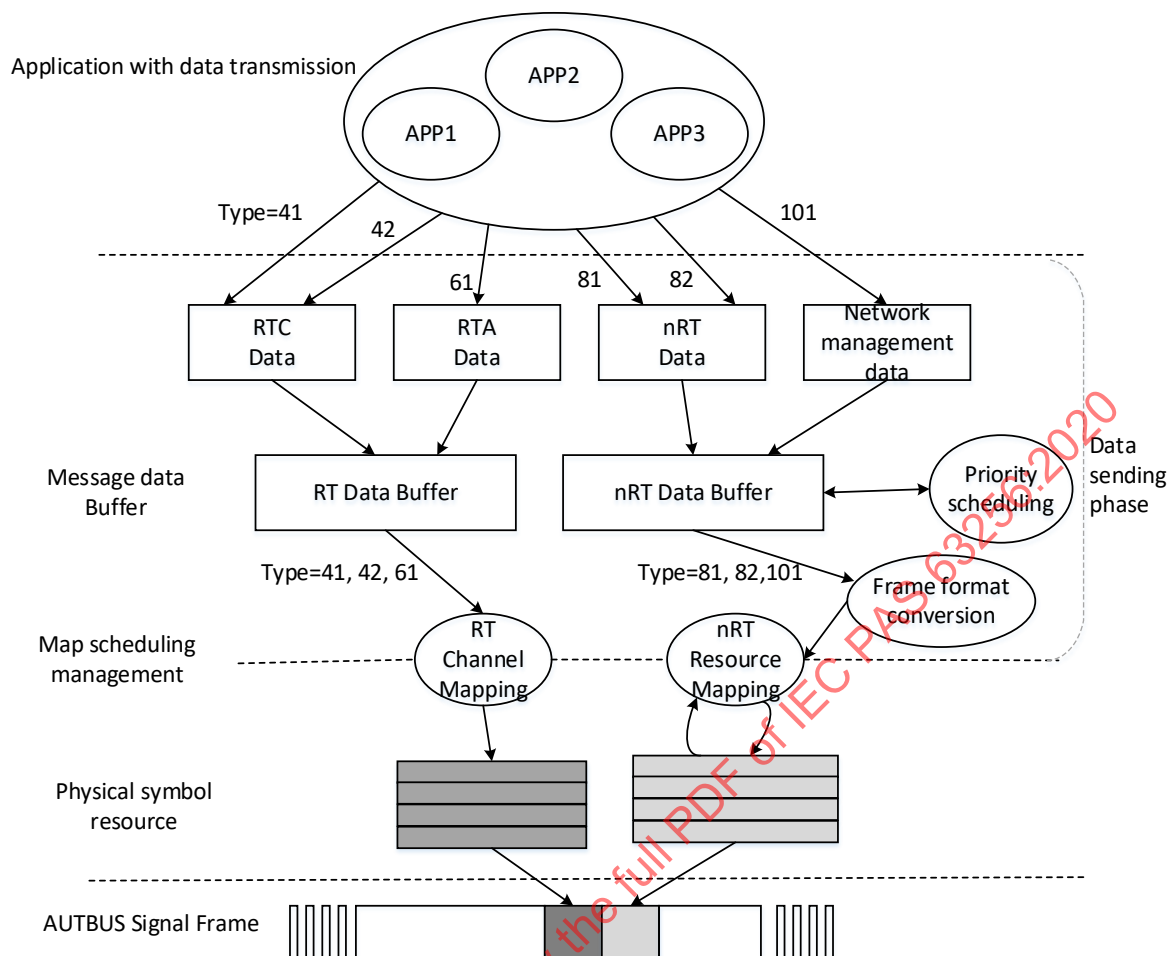


Figure 55 – DLS data sending procedure diagram

After receiving the data from the channel established by the mapping physical symbol resource, the DLL puts the data into the RT data buffer and the nRT data buffer respectively. For nRT data buffers, format conversion is carried out according to configuration and service requirements, and then sent to the corresponding application service module for further processing in the DLPDU structure. For RT data buffer, it should be taken out by the service module in cyclic mode (fixed period) or acyclic mode (non-fixed period) and then continue to process. After the data buffer is processed by the service module, the buffer space is released. The DLL service data receiving procedure is shown in Figure 56.

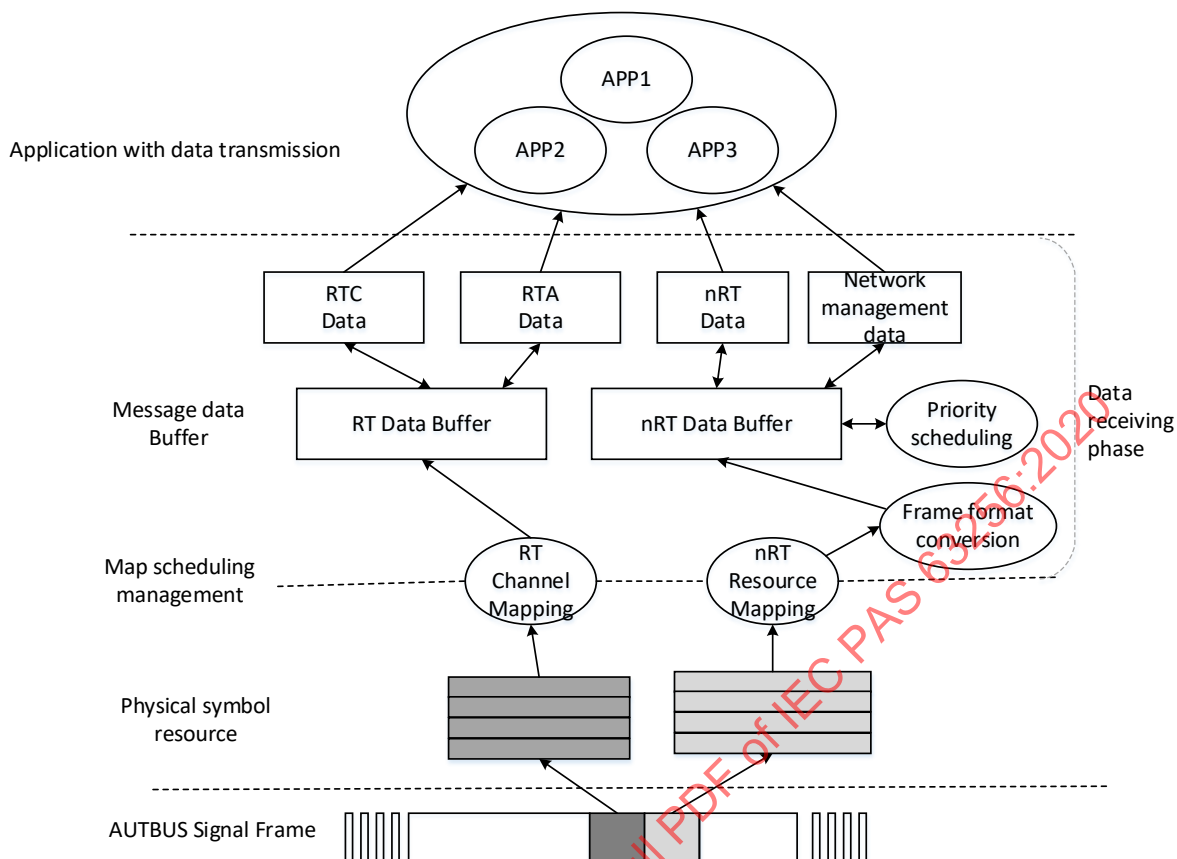


Figure 56 – DLS data receiving procedure diagram

8.3.1.4 Clock synchronization procedure

The DLL clock synchronization service provides delay measurement, clock synchronization and clock interrupt. The delay measurement function is a service function based on the PhL mechanism of AUTBUS protocol. The delay measurement and clock synchronization services require the interaction of the MNs and the TNs on the AUTBUS network.

The DLS-user triggers the measurement of the PhL delay time between nodes in the AUTBUS network based on the delay measurement service.

The DLS-user completes the clock synchronization of all nodes on the network based on delay measurement and synchronization service.

The DLS-user could register the interrupt signal based on the clock interrupt service interface which can make DLS-user to implement the RT service processing with different clock cycles.

See Figure 57 for an example of delay measurement between a MN and a TN:

- The symbol information sent out by the MN at the T₀ time, and the TN will receive the symbol at the corresponding T₁ time after the transmission delay of the physical medium.
- Similarly, the symbol information sent out by the TN at T₂ time, it will be received by the MN at T₃ time after the transmission delay of the physical line.
- Since the above delays are generated in physical signal transmission and are symmetrical to each other, the transmission delay between the MN and the TN is:

$$\Delta t = T_1 - T_0 = T_3 - T_2 \quad (14)$$

d) Because the MN can monitor the expected time and the actual arrival time of the packet sent back by the TN in real time, this time deviation is the transmission delay of the line Δt .

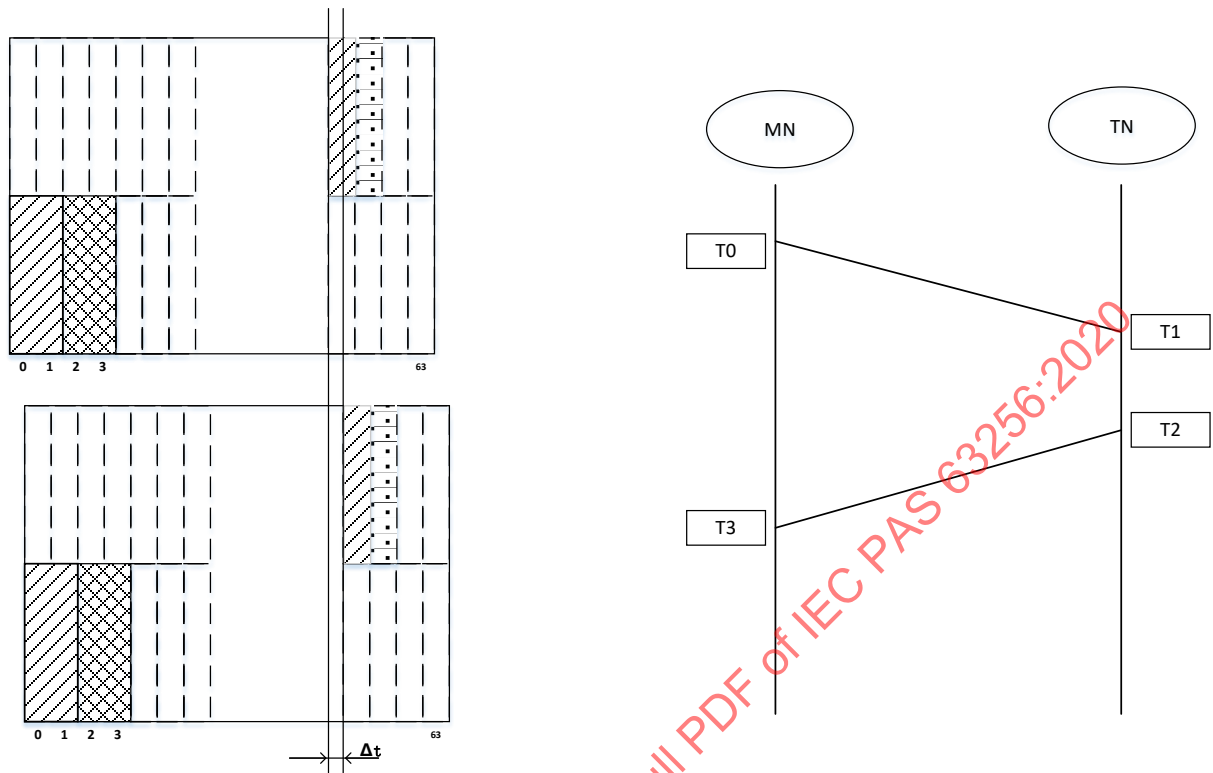


Figure 57 – Clock synchronization delay measurement procedure diagram

Based on the above delay measurement results, the DLLCSS could provide further clock synchronization services to the high layer users of DLL. On the AUTOBUS network, the time information of the MN is always taken as the basic time information. There is a 64-bit data variable on each node in the AUTOBUS network to save time information of the corresponding OFDM symbols, as shown in Figure 58:



Figure 58 – Clock register structure diagram

On the AUTOBUS network, the Frame Index and Symbol Index in the clock registers of all nodes are obtained by synchronous parsing of the physical signals based on the MNs, and only the Cycle Index is calculated according to the local physical signals. This structure can ensure that the clock information of all TNs is consistent with the MN.

On the AUTOBUS network, in addition to the transmission delays between the nodes, there are three timing information that need to be maintained:

- OFDM Timing, the clock registers on each node on the PhL, including Frame Index, OFDM symbol index and Cycle Index.
- Real Timing, at the MN, this time is the time information expressed in hours, minutes, seconds, milliseconds and microseconds as determined by the system clock source, which needs to be transmitted to other node devices on the bus; at the TN, this real timing is the time information recovered from the PhL.

- Trigger Time, it is the Real Timing value at the start of the Frame on the MN which is used to synchronize the MN time for the TN.

The PhL of the MN generates an interrupt signal at the start of the SF, and then copies the current value of Real Timing to the Trigger Time register. The MN uses the corresponding DLL clock synchronization DLPDU to transmit the current Frame index, Trigger Time to other TN devices through the DLL clock synchronization DLPDU. In this way, the TN can synchronize Frame index and Symbol Index with the MN when the system runs stably. When the TN receives the clock synchronization packet from the MN, the Trigger Time, Frame index are placed in the local register in the current SF. When the TN interrupt at the start of the next SF, the calculated time is synchronized to the Real Timing of this node, thus the complete clock synchronization is realized between the TN and the MN. Formula for calculating local time by TN is:

$$\text{Real_Time_TN} = \text{Trigger_Time_MN} + \text{Delay} + T_{\text{Frame}} \quad (15)$$

T_{Frame} is a SF time length.

The Clock synchronization procedure between the MN and the TN is shown in Figure 59.



Figure 59 – Clock synchronization procedure

8.3.2 DLPDU structure

8.3.2.1 Universal DLPDU structure

AUTBUS DLL DLPDUs have a general structure, as shown in Figure 60.

	Bit Offset							
OcteT Offset	7	6	5	4	3	2	1	0
0	TYPE							
1	Fragment_No				LEN			
2	LEN							
3~n	PAYLOAD							
n+1~n+2	CRC							

Figure 60 – Universal DLPDU structure

Each field is defined as:

- TYPE: 1 octet, the type of DLPDU, maximum support 256 different types. The type value 0 ~ 31 is a protocol control DLPDU reserved by the system, including:
 - TYPE=0x00/0x01: Reserved for equipment manufacturers;
 - TYPE=0x02: Basic configuration DLPDU;

- TYPE=0x03: Address allocation DLPDU;
- TYPE=0x04: Multicast address allocation DLPDU;
- TYPE=0x05: RT resource allocation DLPDU;
- TYPE=0x06: nRT resource allocation DLPDU;
- TYPE=0x07: Access notification DLPDU;
- TYPE=0x08: Resource request DLPDU;
- TYPE=0x09: Resource release DLPDU;
- TYPE=0x0A: Status query DLPDU;
- TYPE=0x0B: Status response DLPDU;
- TYPE=0x0C: Announcement DLPDU;
- TYPE=0x10: Clock synchronization DLPDU;
- TYPE=0x0D ~ 0x0F, 0x11 ~ 0x1F: Reserved
- TYPE with 0x20 ~ 0xFF belong to the ordinary DLPDU defined by the user;

TYPE field not only distinguishes different DLPDU types, but also represents the higher DLPDU priority of the DLPDU with smaller value of TYPE.

- Fragment_No: 4bits, Packet fragment sequence number; field is 0 means that the packet is no fragment, the field is not 0 means the sequence number of the packet fragment, and the value 1 means the last piece of packet fragment.
- LEN: 12bits, the length of a DLPDU, unit is octet, indicating that the DLPDU length from the start of the TYPE field to the beginning of the CRC field. AUTOBUS DLL supports a valid DLPDU length range is 4 ~ 2 048 octets.
- PAYLOAD: The data payload of a DLPDU, its length is related to type. For different types of DLPDUs, the data with different structures and lengths are defined in the payload section.
- CRC: 2 octets of CRC-16 check bit, the CRC DLPDU automatically calculated and valued by the hardware, the field will not be described again in this document.

8.3.2.2 Basic configuration DLPDU

In the initialization phase of AUTOBUS network, the MN broadcasts the basic configuration to all the TNs on the network, so that the basic configuration is consistent, and the protocol state machine should be negotiated in the next step. This type of DLPDU is sent once per physical SF period, and synchronously realizes SF number alignment in the network.

The basic configuration includes: working mode, SF structure configuration, node random access resource allocation, NodeID and multicast address, access mode, SF sequence number alignment in initialization phase. The DLPDU structure is shown in Figure 61.

	Bit Offset							
Octet Offset	7	6	5	4	3	2	1	0
0	TYPE							
1	Fragment_No				LEN			
2	LEN							
3	VER							
4	Reserved							
5 ~ 10	FrameID_Seq							
11 ~ 24	COMM_Cfg							
25	Reserved							
26 ~ 27	CRC							

Figure 61 – Basic configuration DLPDU structure

Each field is defined as:

- TYPE: 1 octet, set as 0x2, basic configuration DLPDU type.
- Fragment_No: 4 bits, Packet fragment sequence number, refer to universal DLPDU structure description.
- LEN: 12 bits, the length of DLPDU octets from the start of the TYPE field to the beginning of the CRC field, refer to universal DLPDU structure description.
- VER: 1 octet, Version number of AUTOBUS Protocol.
- Reserved: The 4th octet and the 25th octet are reserved fields.
- FrameID_Seq: 6 octets, MN to the physical SF relative count, the valid value range is 0 ~ (2⁴⁴-1), for the user node alignment count. This number is unique for each cycle of the current network.
- COMM_Cfg, Equal to or greater than 14 octets, general configuration block information for the network, as shown in Figure 62.

	Bit Offset							
Octet Offset	7	6	5	4	3	2	1	0
0	TransmitMode				Default_USB_WorkMode			
1	Default_USB_Work Mode		Default_LSB_WorkMode					
2	AccessStatus				Frame_Fmt			
3	Ucn_Num							
4	Mcn_Num							
5	DownStream_Symbol_Num							
6	Cycle_Frame_Num				App_Mode		Reserved	
7	R_RT_Attr							
8	R_RT_Attr				R_Cycle_Symbol_Len			
9	R_Cycle_Symbol_Len							
10	R_Band_Flag		R_Band_Mode					
11	R_Symbol_Start							
12	R_Symbol_Start				R_Symbol_Num			
13	R_Symbol_Num							

Figure 62 – General configuration block structure

- TransmitMode: 4 bits, system TM, value is 0, 1, 2, 3.
- Default_USB_WorkMode: 6 bits, default working mode of upper-side band.
- Default_LSB_WorkMode: 6 bits, default working mode of lower-side band.
- AccessStats: 4 bits, access status.
- Frame_Fmt: 4 bits, SF structure, default 0 identifies a physical SF with 64 symbols.
- Ucn_Num: 1 octet, total number of NodeID including MN.
- Mcn_Num: 1 octet, the total number of multicast group NodeID, the maximum number of multicasts is 16, the default is 0.
- DownStream_Symbol_Num: 1 octet, the number of symbols of the downlink subframe, with a default value of 2, see Table 66.
- Cycle_Frame_Num: 4 bits, the number of cycles of cyclic frame, see Table 66.
- App_Mode: 2 bits, Access mode configuration, 0: fixed algorithm access mode; 1: user algorithm access mode.
- Reserved: 2bits, reserved field.

- R_Cycle_Symbol_Len: 12 bits, cycle length of cyclic symbol of Random access symbol, see Table 66.
- R_RT_Attr: 12 bits, RT attributes of random access symbols, see Table 66.
- R_Band_Flag: 2 bits, the side band occupation identification of the random access symbol, see Table 66.
- R_Band_Mode: 6 bits, the side band occupation working mode of the random access symbol, see Table 66
- R_Symbol_Start: 12 bits, Random access symbol start number, see Table 66.
- R_Symbol_Num: 12 bits, Number of random access symbols, see Table 66.

8.3.2.3 Address assignment DLPDU

The address assignment DLPDU is used to realize the NodeID allocation function. In the initial phase, the MN broadcasts the NodeID resource allocation to all TNs on the network. The TN self-configures the allocated NodeID, and realizes the node identification and addressing based on NodeID.

The structure of the address assignment DLPDU is shown in Figure 63.

	Bit Offset							
Octet Offset	7	6	5	4	3	2	1	0
0	TYPE							
1	Fragment_No				LEN			
2	LEN							
3 ~ 4	Start_NodeID							
5	Ucn_Num							
6 ~ 11	MAC_Addr_1							
...								
(6*n) ~ (6*n+5)	MAC_Addr_n							
(6*n+6)~(6*n+7)	CRC							

Figure 63 – Address assignment DLPDU structure

Each field is defined as:

- TYPE: 1 octet, set as 0x3, NodeID address allocation DLPDU.
- Fragment No: 4 bits, Packet fragment sequence number, refer to universal DLPDU structure description.
- LEN: 12 bits, DLPDU length, the length of DLPDU octets from the start of the TYPE field to the beginning of the CRC field, refer to universal DLPDU structure description. The allocation information of address resources can be processed in batches after the consecutive NodeID sections are allocated.
- Start_NodeID: 2 octets, Start NodeID of the NodeID resource allocation section.
- Ucn_Num: 1 octet, the length of the NodeID resource allocation section, that is, the number of NodeID that the section contains.
- MAC_Addr_1 ~ MAC_Addr_n: MAC addresses list with Ucn_Num nodes initially assigned by Start_NodeID. Each consecutive 6 octets from the 6th octet as a valid MAC address for a total of Ucn_Num MAC addresses.

8.3.2.4 Multicast assignment DLPDU

Multicast address allocation function is used to implement multicast services in the network. In the initialization phase of the network, the MN broadcasts the multicast address allocation results to all nodes.

The structure of the multicast assignment DLPDU is shown in Figure 64.

	Bit Offset							
Octet Offset	7	6	5	4	3	2	1	0
0	TYPE							
1	Fragment_No				LEN			
2	LEN							
3 ~ 4	MC_NodeID							
5	Group_Nodes_Num							
6	Group_Nodes_Mem							
...	...							
38	Group_Nodes_Mem							
39	Reserved							
40 ~ 41	CRC							

Figure 64 – Multicast assignment DLPDU structure

Each field is defined as:

- TYPE: 1 octet, set as 0x04, multicast groups address allocation DLPDUs.
- Fragment_No: 4 bits, packet fragment sequence number, refer to universal DLPDU structure description.
- LEN: 12 bits, refer to universal DLPDU structure description.
- MC_NodeID: 1 octet, the corresponding multicast group NodeID, which broadcast the address assignment result of one multicast group at a time.
- Group_Nodes_Num: 1 octet, the number of multicast members included in a MC_NodeID multicast group.
- Group_Nodes_Mem: 33 octets, based on the bitmap format, the corresponding bit is 0 indicates that the corresponding NodeID node is not in the current multicast group and the 1 indicates the corresponding NodeID node is in the current multicast group.
- Reserved: 1 octet, reserved field.

8.3.2.5 Resource allocation DLPDU

The data processed in the DLL is divided into RT data and nRT data (see 6.3.1.1.2), and the corresponding resource allocation can be configured before the network starts, the MN sends to all nodes in the network through the bandwidth resource allocation DLPDU to complete the configuration and mapping of the related resources in the initialization access configuration phase. Or during the operation of the system, the TN can apply for the resources dynamically from the MN according to the service requirement, and the MN can assign the corresponding resources to this node to carry on the data transmission processing. The resource allocation DLPDUs for RT and nRT resources have the same format, except that the corresponding type values are different. For RT resource allocation DLPDUs, TYPE=0x05, nRT resource allocation DLPDUs, TYPE=0x06, corresponding DLPDU formats is shown in Figure 65.

	Bit Offset							
Octet Offset	7	6	5	4	3	2	1	0
0	TYPE							
1	Fragment_No				LEN			
2	LEN							
3 ~ 4	Channel_ID							
5	Sequence_No							
6 ~ 7	Dst_NodeID							
8	Service_Type							
9	RxTx_Attr		RT_Attr					
10	RT_Attr						Res_Flag	
11	Cycle_Symbol_Len							
12	Cycle_Symbol_Len				Reserved1			
13	Band_Flag_1		Band_Mode_1					
14	Symbol_Start_1							
15	Symbol_Start_1				Symbol_Num_1			
16	Symbol_Num_1							
17	Band_Flag_2		Band_Mode_2					
18	Symbol_Start_2							
19	Symbol_Start_2				Symbol_Num_2			
20	Symbol_Num_2							
21	Reserved2							
22 ~ 23	CRC							

Figure 65 – Resource allocation DLPDU structure

Each field is defined as:

- TYPE: 1 octet, 0x05: RT resource allocation DLPDU; 0x06: nRT resource allocation DLPDUs.
- Fragment_No: 4 bits, packet fragment sequence number, refer to universal DLPDU structure description.
- LEN: 12 bits, Refer to universal DLPDU structure description.
- Channel_ID: 2 octets, the corresponding connection resource channel ID number.
- Sequence_No: 1 octet, resource allocation DLPDU sequence number, when there are more than two channel resource blocks at the same DTC Channel_ID, multiple resource allocation DLPDUs should be used for allocation.
- Dst_NodeID: 2 octet; Gets the NodeID value of the node of the channel resource.
- Service_Type: 1 octet; the service data type TYPE ID used by the resource channel, see Table 75.
- RxTx_Attr: 2 bits, bus data processing mode on the channel resources, 0x0 for read and 0x1 for write, see Table 75.
- RT_Attr: 12 bits, the RT attribute of the resource channel, see Table 75.
- Res_Flag: 2 bits, flag whether the first and second groups of symbol resource block configurations are valid or not, bit0 flag the first group configuration and bit1 flag the second group configuration; the corresponding bit location 1 flag that the group configuration is valid.
- Cycle_Symbol_Len: 12 bis; flag the cycle length of the symbol, see Table 75.
- BandFlag1 ~ 2: 2 bits, the upper and lower side band of symbol based on bit, see Table 75.
- Band_Mode_1 ~ 2: 6 bits, the working mode of allocated the first and second groups of symbol resources, see Table 75.

- Symbol_Start_1 ~ 2: 12 bits, start symbol number of allocated the first and second groups of symbol resources; see Table 75.
- Symbol_Num_1 ~ 2: 12 bits, the number of symbols of allocated the first and second groups of symbol resources; see Table 75.
- Reserved1: 4 bits, reserved field.
- Reserved2: 1 octet, reserved field.

8.3.2.6 Access notification DLPDU

When TN starts up, if related resource for sending is allocated, it broadcasts the access notification DLPDU to the network. The content of the transmission includes destination node NodeID, own NodeID, the status of this node and the MAC address information, the numbers of notification DLPDUs received from different NodeID. The access notification DLPDU is the basis for the communication between nodes of the entire network. The MN perceives the access of the TN in the network in real time.

The access notification DLPDU structure is shown in Figure 66.

	Bit Offset							
Octet Offset	7	6	5	4	3	2	1	0
0	TYPE							
1	Fragment_No				LEN			
2	LEN							
3 ~ 4	Dst_NodeID							
5 ~ 6	Src_NodeID							
7	Nodes_Num							
8 ~ 13	Mac_Addr							
14	Node_Status							
15	Reserved							
16 ~ 17	CRC							

Figure 66 – Access notification DLPDU structure

Each field is defined as:

- TYPE: 1 octet, set as 0x07, access notification DLPDU.
- Fragment_No: 4 bits, packet fragment sequence number, refer to universal DLPDU structure description.
- LEN: 12 bits, refer to universal DLPDU structure description.
- Dst_NodeID: 2 octets, Destination node address NodeID value.
- Src_NodeID: 2 octets, source node address NodeID value.
- Nodes_Num: 1 octet, the numbers of access notification DLPDUs received from different nodes correctly.
- Mac_Addr: 6 octets, identify the MAC address of the source node.
- Node_Status: 1 octet, the running state of this node, 0x0 is initialization phase, 0x1 is ready; other values reserved; see Table 69.
- Reserved: 1 octet, reserved field.

8.3.2.7 Resource application DLPDU

Before the AUTOBUS network works, the MN transmits the physical symbol resources allocation result to each TN in the network. After the network starts to work, the TN dynamically applies for nRT resources to process the sending and receiving of corresponding nRT data (see 8.3.1.2.2). For the MN, the dynamic application should be completed locally; for the TN, the application information should be encapsulated as a dynamic resource application DLPDU and sent to the MN for allocation. If the application is successful, the transmission is processed based on the corresponding resource. If the allocation failed, the corresponding nRT data will be discarded and the DLS-user should be notified.

The structure of the resource application DLPDU is shown in Figure 67.

	Bit Offset							
Octet Offset	7	6	5	4	3	2	1	0
0	TYPE							
1	Fragment_No				LEN			
2	LEN							
3 ~ 4	Dst_NodeID							
5 ~ 6	Src_NodeID							
7 ~ 8	Channel_ID							
9	ServiceType							
10 ~ 11	AppBw							
12 ~ 13	LoopTime							
14 ~ 15	Reserved							
16 ~ 17	CRC							

Figure 67 – Resource application DLPDU structure

Each field is defined as:

- TYPE: 1 octet, set as 0x08, dynamic resource application DLPDU.
- Fragment_No: 4 bits, packet fragment sequence number, refer to universal DLPDU structure description.
- LEN: 12 bits, refer to universal DLPDU structure description, the resource request DLPDU length is fixed to 18.
- Dst_NodeID: 1 octet, Destination node address NodeID value, default is MN's NodeID.
- Src_NodeID: 1 octet, Source node address NodeID value for resource application, default is the NodeID of this node
- Channel_ID: 2 octets, the corresponding resource channel ID number; the ID number is a full F invalid value when the user applied for the first time.
- Service_Type: 1 octet, the service data type values that dynamically request resources to be loaded, see Table 69.
- AppBW: 2 octets, the size of the amount of service data that needs to be applied for resource to be loaded, the unit is octet, see Table 69.
- LoopTime: 2 octets, the cycle time to apply for resource carrying data, the unit is us, see Table 69.
- Reserved: 2 octets, reserved field.

8.3.2.8 Resource release DLPDU

The resource dynamically requested by the TN can be used within the time of application. When the number of times and time of application meet the set threshold, the resource should be automatically released and recycled by the system. In some cases, the TN can proactively release the requested resource by sending the resource release DLPDU to the MN.

The structure of the resource release DLPDU is shown in Figure 68.

	Bit Offset							
Octet Offset	7	6	5	4	3	2	1	0
0	TYPE							
1	Fragment_No				LEN			
2	LEN							
3 ~ 4	Dst_NodeID							
5 ~ 6	Src_NodeID							
7 ~ 8	Channel_ID							
9	Sequence_No							
10	ServiceType							
11	Reserved							
12 ~ 13	CRC							

Figure 68 – Resource release DLPDU structure

Each field is defined as:

- TYPE: 1 octet, set as 0x09, dynamic resource request DLPDU.
- Fragment_No: 4 bits, packet fragment sequence number, refer to universal DLPDU structure description.
- LEN: 12 bits, refer to universal DLPDU structure description.
- Dst_NodeID: 2 octets, NodeID value corresponds to the destination node address for the release service.
- Src_NodeID: 2 octets, NodeID value corresponds to the source node address for the release service.
- Channel_ID: 2 octets, the released resource channel ID number.
- Sequence_No: 1 octet, resource allocation DLPDU number, if the field is 0, delete all symbol resources in the channel.
- ServiceType: 1 octet, based on the service data type values loadedd by the released resource channel, see Table 69.
- Reserved: 1 octet, reserved field.

8.3.2.9 Status query DLPDU

The maintenance service mainly completes the specific function by the status query DLPDU and the status response DLPDU, the maintenance service function can refer to the description of the command parameter in Table 71. When the MN detects that no DLPDU has been sent by a specified node within a certain period of time, or when a new node randomly accesses, the MN broadcasts the status query DLPDU, and the normal nodes should respond with the access notification DLPDU. The node that runs abnormally or has been offline should not respond.

The structure of the status query DLPDU is shown in Figure 69.

	Bit Offset							
Octet Offset	7	6	5	4	3	2	1	0
0	TYPE							
1	Fragment_No				LEN			
2	LEN							
3 ~ 4	Dst_NodeID							
5 ~ 6	Src_NodeID							
7	Command							
8 ~ 9	CRC							

Figure 69 – Status query DLPDU structure

Each field is defined as:

- TYPE: 1 octet, set as 0x0A, status query DLPDU.
- Fragment_No: 4 bits, packet fragment sequence number, refer to universal DLPDU structure description.
- LEN: 12 bits, refer to universal DLPDU structure description.
- Dst_NodeID: 1 octet, destination node address NodeID value for query.
- Src_NodeID: 1 octet, source node address NodeID value for query.
- Command: 1 octet, the command word for query operation at the destination node is shown in Table 71.

8.3.2.10 Status response DLPDU

The destination node of the status query confirms whether or not response according to the command word in the status query DLPDU, and sends the status response DLPDU to the status query source node. The status response DLPDU contains the result of the query and the status information.

The structure of the status response DLPDU is shown in Figure 70.

	Bit Offset							
Octet Offset	7	6	5	4	3	2	1	0
0	TYPE							
1	Fragment_No				LEN			
2	LEN							
3 ~ 4	Dst_NodeID							
5 ~ 6	Src_NodeID							
7	Command							
8	NodeState							
9 ~ 10	ChannelCnt							
10 ~ 13	WarnCnt							
14 ~ 17	ErrCode							
18 ~ 19	Reserved							
20 ~ 21	CRC							

Figure 70 – Status response DLPDU

Each field is defined as:

- TYPE: 1 octet, set as 0x0B, status query DLPDU.
- Fragment_No: 4 bits, packet fragment sequence number, refer to universal DLPDU structure description.
- LEN: 12 bits, refer to universal DLPDU structure description.
- Dst_NodeID: 2 octets, NodeID value of destination node address for response.
- Src_NodeID: 2 octets, NodeID value of source node address for response.
- Command: 1 octet, response command values that identify the response source node. See Table 72.
- NodeState: 1 octet, the status of the response source node, see Table 72.
- ChannelCnt: 2 octets, the number of DTCs that are being used by the source node. See Table 72.
- WarnCnt: 4 octets, the number of alerts generated by the response source node, see Table 72.
- ErrCode: 4 octets, the last error code on the reply source node, see Table 72.
- Reserved: 2 octets, reserved field.

8.3.2.11 Announcement DLPDU

During link maintenance, the MN sends the announcement DLPDU to the AUTOBUS network, and the TN confirms whether the local configuration is consistent with the MN based on the DLPDU. The announcement DLPDU structure is shown in Figure 71.

	Bit Offset							
Octet Offset	7	6	5	4	3	2	1	0
0	TYPE							
1	Fragment_No				LEN			
2	LEN							
3 ~ 4	Dst_NodeID							
5 ~ 6	Src_NodeID							
7	Command							
8	Ucn_Num							
9	Mcn_Num							
10	TransmitMode							
11	Default_USB_WorkMode						Default_LSB_WorkMode	
12	Default_LSB_WorkMode				Cycle_Frame_Num			
13	R_RT_Attr							
14	R_RT_Attr				R_Cycle_Symbol_Len			
15	R_Cycle_Symbol_Len							
16	R_Band_Flag		R_Band_Mode					
17	R_Symbol_Start							
18	R_Symbol_Start				R_Symbol_Num			
19	R_Symbol_Num							
20 ~ 21	CRC							

Figure 71 – Announcement DLPDU structure

Each field is defined as:

- TYPE: 1 octet, set as 0x0C, announcement DLPDU.
- Fragment_No: 4 bits, packet fragment sequence number, refer to universal DLPDU structure description.
- LEN: 12 bits, refer to universal DLPDU structure description.
- Dst_NodeID: 2 octets, NodeID value of destination node address for response.
- Src_NodeID: 2 octets, NodeID value of source node address for response.
- Command: 1 octet, the response command value that identifies the response source node, see Table 66.
- Ucn_Num: 1 octet, Total number of NodeID including MN.
- Mcn_Num: 1 octet, the total number of multicast group NodeID, the maximum number of multicasts is 16, default is 0.
- TransmitMode: 4 bits, system TM, value is 0x0, 0x1, 0x2, 0x3.
- Default_USB_WorkMode: 6 bits, upper-side band default work mode.
- Default_LSB_WorkMode: 6 bits, lower-side band default work mode.
- Cycle_Frame_Num: 4 bits, cyclic frame cycle number, see Table 66.
- R_Cycle_Symbol_Len: 12 bits, cyclic symbol length of random access symbols, see Table 66.
- R_RT_Attr: 12 bits, the RT attributes of random access symbols, see Table 66.
- R_Band_Flag: 2 bits, the side band occupation identification of the random access symbol, see Table 66.
- R_Band_Mode: 6 bits, the sideband mode of random access symbols, see Table 66.
- R_Symbol_Start: 12 bits, Random access symbol start number, see Table 66.
- R_Symbol_Num: 12 bits, Number of random access symbols, see Table 66.

8.3.2.12 Clock synchronization DLPDU

Clock synchronization provides delay measurement and clock synchronization functions for DLS-user, the MN synchronizes clock information to designated TNs or all TNs based on the DLPDU. The TN should respond to the clock synchronization of the MN immediately. This DLPDU sent by the MN to the TN, and this DLPDU sent by the TN to the MN has the same structure, the specific contents of the DLPDU are different according to the difference of synchronous action.

The clock synchronization DLPDU format is shown in Figure 72.

	Bit Offset							
Octet Offset	7	6	5	4	3	2	1	0
0	TYPE							
1	Fragment_No				LEN			
2	LEN							
3 ~ 4	Dst_NodeID							
5 ~ 6	Src_NodeID							
7 ~ 8	Sequence							
9	Action							
10 ~ 15	Delay							
16	FrameID_S							
17 ~ 22	TimeInfo_Second							
23 ~ 28	TimeInfo_Ns							
29 ~ 32	Status							
33 ~ 34	CRC							

Figure 72 – Clock synchronization DLPDU structure

Each field is defined as:

- TYPE: 1 octet, set as 0x10, clock synchronization DLPDU.
- Fragment_No: 4bits, packet fragment sequence number, refer to universal DLPDU structure description.
- LEN: 12 bits, refer to universal DLPDU structure description.
- Dst_NodeID: 2 octets, NodeID value of destination node address for clock synchronization.
- Src_NodeID: 2 octets, NodeID value of source node address for clock synchronization.
- Sequence: 2 octets, the sequence number of the delay measurement service, see Table 82.
- Action: 1 octet, clock synchronization actions, see Table 82 and Table 84 for action parameters description.
- Delay: 6 octets, the result of delay measurement between nodes, 32 bits average delay and 32 bits current delay.
- FrameID_S: 1 octet, the lowest 8bit value for the current SF ID of the trigger clock synchronization service.
- TimeInfo_Second: 6 octets, range 0 to $(2^{48}-1)$, the time information of above the second of the real time clock at the start of the current SF.
- TimeInfo_Ns: 6 octets, range 0 to $(2^{48}-1)$, the time information below the second of the real time clock at the start of the current SF, unit is ns.
- Status: 4 octets, clock synchronizes state parameters, see Table 87.

8.3.2.13 Common DLPDU

Common DLPDUs are used to carry the transmission service of application data, including RT field process data and nRT network management data. The corresponding DLPDU structure is shown in Figure 73.

	Bit Offset							
Octet Offset	7	6	5	4	3	2	1	0
0	TYPE							
1	Fragment_No				LEN			
2	LEN							
3 ~ 4	Dst_NodeID							
5 ~ 6	Src_NodeID							
7 ~ n	Payload							
n+1 ~ n+2	CRC							

Figure 73 – Common DLPDU structure

Each field is defined as:

- TYPE: 1 octet, the octet of data service DLPDU, value range is 32 ~ 255.
- The TYPE field in the AUTOBUS DLPDU structure not only identifies the different service, but also identifies the priority of the corresponding service. Corresponding to RT service data, TYPE should select a smaller value because of its high priority, while for nRT service data, it has a lower priority and a larger value of TYPE.
- Fragment_No: 4 bits, packet fragment sequence number, refer to universal DLPDU structure description.
- LEN: 12 bits, refer to universal DLPDU structure description.
- Dst_NodeID: 2 octets, destination node address NodeID value.
- Src_NodeID: 2 octets, source node address NodeID value.
- PAYLOAD: The length of the data loaded by the DLPDU is related to the TYPE. For different types of DLPDUs, the data of different structures and lengths are defined in the load part.

8.3.3 State machine

8.3.3.1 DLDE state machine

Figure 74 shows the DLDE state machine.

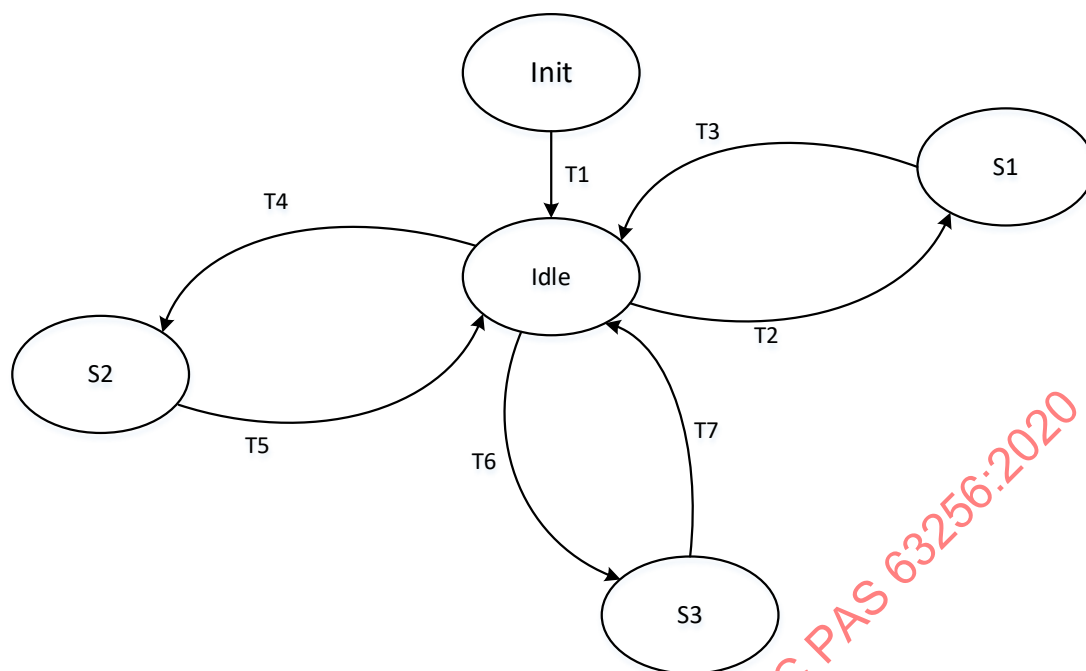


Figure 74 – DLDE state machine

The DLDE state machine has five states:

- Init: The DLDE in the DLL initialization state, the DLDE shall enter Idle state after the configuration of the access initialization.
- Idle: The normal working state of a DLDE at the DLL, the DLDE switches among the three states of S1, S2 and S3. The Idle state is a superstate, that is if the DLDE has multiple session connections at the same time, then it allows multiple Idle states to exist at the same time, or for continuous cyclic data sending and receiving processes, that is DLDE will always transit between S2 and Idle and Idle and S3, the idle states can be ignored at this moment.
- S1: DLDE configuration state, the configuration mapping of DLDE is usually triggered by the DLMS interface, the DLDE completes the related resource allocation and mapping in this state.
- S2: In this state, the DLDE in the DLL encapsulate and parse the DLPDU structure, the data that needs to be sent is encapsulated into the correct DLPDU format, and for the received DLPDUs, the valid data is obtained after parsing for the next step.
- S3: In this state, the DLDE completes the data sending and receiving based on the PhL service interface.

Table 88 – DLDE state transition description

Number	Current state	description	Next state
T1	Init	After initialization, the DLDE with the Idle state, and the DLDE does not transfer data before the DLDE is configured and mapped.	Idle
T2	Idle	The system management user or the DLS-user calls the DLMS interface will trigger the configuration operation of the DLDE, and the DLDE shall enter this state after completing the resource mapping configuration. The following call of the service interface triggers the state transition of the DLDE: DLME-Config DLME-Discovery DLME-Maintain DLME-Establish DLME-Release DLME-Update	S1
T3	S1	The DLDE configuration end or release operation triggers the DLDE transition from the configuration state to the Idle state. In theory, the service operation involved in T2, triggering the DLDE to the configuration state, the configuration completion let the state of data entity migrate back to the Idle state. In addition, if the configuration failed or an exception occurred the state machine should migrate from the configuration state to the Idle state.	Idle
T4	Idle	The DLDE receives the data transmission request of the DLS-user, which will trigger the DLPDU structure encapsulation process of the DLDE, and the data respond to the request of the PhL service interface to receive the packet, the DLDE will parse the DLPDU. The service interfaces that can trigger this state transition are: DLDE-CLMDTA DLDE-CLMDTNA DLDE-CLMRDTA DLDE- CLMDTRRNA DLDE-CMDTA DLDE-CMDTNA	S2
T5	S2	The DLDE notifies the physical service interface to send the data after completing the encapsulation of the data. The DLDE needs to enter the data sending state S3 before sending with the determined physical resource, the DLDE is recovered to a temporary Idle state before the physical resources are scheduled in time, the service interface which make data entity migrate the Idle state to S2 state in the T4 phase, it also causes the state transition. For the data receiving process, after the DLPDU parsing is finished, the DLDE actually enters the idle state and waits for the next receiving data process, while the data put into the data buffer will be received and processed by the user.	Idle

Number	Current state	description	Next state
T6	Idle	After completing the data encapsulation, the DLDE will immediately notify the physical service interface to send the data, and the DLDE will call the actual PhL service interface to process the data. The DLDE receives the signal from the PhL service interface will enter the data receiving state, completes the data receiving, and the service interface that leads to the T4 phase state transition will lead to the state transition.	S3
T7	S3	After calling the PhL service interface to complete a send and receive process, the DLDE will enter the idle state if there is no more data processing. As above, the service interface that lead to the T4 phase state transition will eventually lead to this state transition.	Idle

8.3.3.2 DLME state machine

Figure 75 shows the DLME state machine.

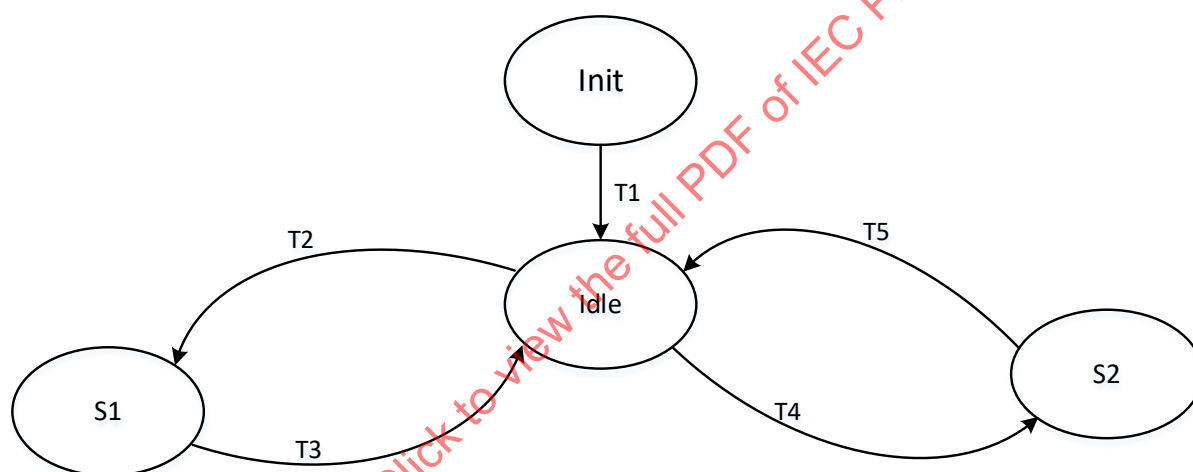


Figure 75 – DLME state machine

The processing of the DLME can trigger the state transition of the DLDE, there are four states for the DLME itself:

- Init: DLME initialization status, which can be configured and maintained after successful initialization.
- Idle: The normal state of the DLME, based on the DLME, the configuration and maintenance operations of the DLDE can be triggered to enter the configuration state and the maintenance state respectively.
- S1: The configuration state of the DLME, the configuration operation triggered to the DLME will make the DLME enter the state, and the configuration state of the DLME will confirm the protection state of the DLME according to the configuration requirements.
- S2: The maintenance state of the DLME, the maintenance operation of the DLME during normal operation after configuration will make the DLME enter the maintenance state.

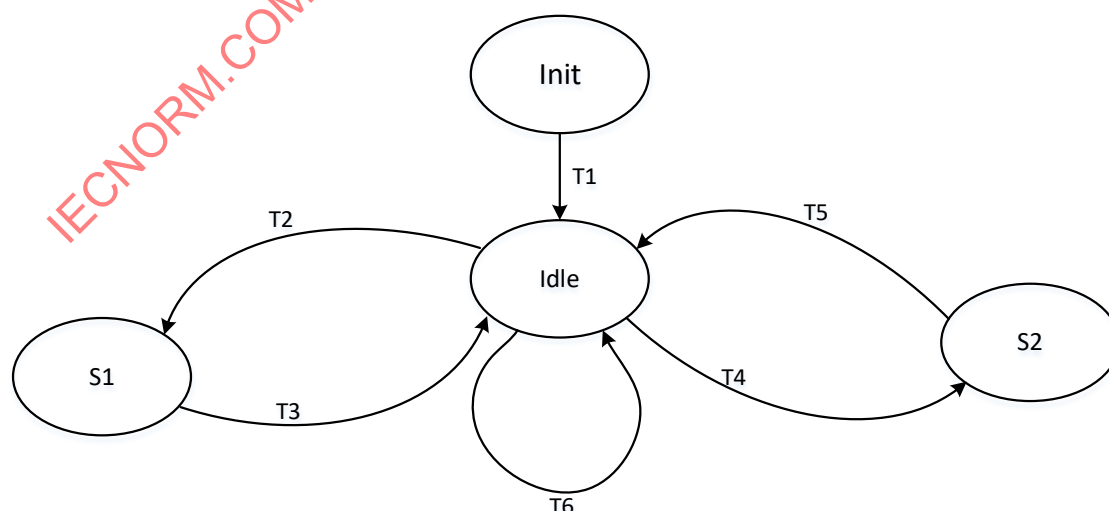
Table 89 defines the DLME state machine state transitions.

Table 89 – DLME state machine state transition description

Number	Current state	description	Next state
T1	Init	For successfully initialized management entities, it automatically enters the state by default, waiting for the user or system to configure and maintain the DLME	Idle
T2	Idle	The DLS-user or system management user call management service interface triggers the DLME state migrate to the configuration state, and the service operation that trigger the state transition are: DLME-Config DLME-Discovery DLME-Establish DLME-Release	S1
T3	S1	The finish of the DLME configuration service operation will make the DLME state machine migrate from the configuration state to the Idle state. If there are errors or exceptions in the configuration process, the state machine will migrate directly to the Idle state.	Idle
T4	Idle	During the operation of the device, the state of the DLME is always switched between idle and S2, and the frequency of the switch depends on the clock cycle of the management service or the frequency of the application sublayer trigger. The management service operations that can trigger this state transition are: DLME-Maintain DLME- Update	S2
T5	S2	The maintenance state of the DLME will enter the Idle state due to the service operation errors and exceptions, and the normal end of the maintenance service operation will also make the DLME migrate from the maintenance state to the Idle state.	Idle

8.3.3.3 DLCE State Machine

Figure 76 shows the DLCE state machine.

**Figure 76 – DLCE state machine**

The DLCE state machine contains four states:

- Init: The initialization state of clock synchronization, after which the clock synchronization related service is ready.
- Idle: The default working state of the clock synchronization service.
- S1: Measure state, based on this state, complete delay measurement correlation processing.
- S2: Synchronization state, which provides the user with the status of clock synchronization, based on this state, clock synchronization is completed.

Table 90 defines the DLCE state machine state transitions.

Table 90 – DLCE state machine state transition description

Number	Current state	Description	Next state
T1	Init	After the clock synchronization related initialization settings are complete, the clock synchronization service enters the normal working state, that is, the Idle state	Idle
T2	Idle	The delay measurement can trigger the DLCE state machine to enter the S1 state. In the S1 state, the clock synchronization service completes the delay measurement process, including broadcast the delay measurement results to all nodes. The service interface that triggers the state transition is: DLCE-Measurement	S1
T3	S1	The finish of the delay measurement makes the state machine back to the idle working state, and the service interface that triggers the state transition is: DLCE-Measurement	Idle
T4	Idle	The clock synchronization operation can trigger the DLCE state machine to enter the S2 state, in S2 state, complete the related processing of clock synchronization, once clock synchronization processing, including the MN send clock synchronization DLPDU to all TNs, and the possible TN status is confirmed to be end. The service interface that triggers this state transition is: DLCE-Synchronization	S2
T5	S2	The end of the clock synchronization operation makes the DLCE state machine back to its normal working state, that is the Idle state, and the service interface that triggers the state transition is: DLCE-Synchronization	Idle
T6	Idle	By registering the clock interrupt service, the user can periodically obtain the clock interrupt signal at a specified length of time during the normal service of clock synchronization, the interface that triggers the service is: DLCE-Interrupt	Idle

8.3.4 Error handling

Error handling in DLL is the basis of diagnosis. The real error source is usually detected by analysing multiple error features on multiple nodes. According to the error characteristics and error sources, nodes should respond at different levels. Error handling should be simple and easy to implement.

8.3.4.1 Possible sources and characteristics of errors

The following error characteristics of the PhL error source are handled by the MNs and TNs:

- Lost connection.
- CRC error.
- Sending buffer overflow.
- Physical symbol resources clash.

8.3.4.2 Error handling of MN / TN

State transitions should be notified to all nodes and all log events are registered.

Each error characteristic has a corresponding accumulating counter and a threshold counter. If the accumulating counter exceeds the threshold count in a certain time period the action of handling the error is triggered.

The system startup or reset command should not reset the accumulating counter, should use a method that provides a reset accumulating counter.

Threshold counters support configuration.

8.3.4.3 PhL error source

The DLL uses PhL error source to diagnose DLL communication error characteristics.

8.3.4.3.1 Lost connection

If the connection is broken, the connection may be lost. For example, break the wire and unplug the network cable.

Lost connection is a late detected error source, which belongs to the basic error detection. Whenever a DLPDU loss is detected in the DLL, the corresponding MN or TN checks whether there is disconnection on the PhL's MAC controller.

If lost connection is detected, it is recorded in the error history.

Lost connection should be reported regardless of any error log triggered by previous basic error detection.

8.3.4.3.2 CRC error

Transmission errors are detected by the AUTBUS controller's hardware (CRC check), and received DLPDUs with CRC errors are discarded.

8.3.4.3.3 Buffer overflow

If the TN or the MN data buffer overflows, the data transmission will fail or the expected data will not be received. The response error is logged and the corresponding counter value increases.

8.3.4.3.4 Symbol resource conflict

If a large amount of RT aperiodic data is needed to be sent, the RT resource based on the setting cannot handle the data, which causes the DLPDU to be unable to send temporarily, resulting in the symbol resource clash error.

8.3.4.3.5 Symbol resource insufficient

When AUTBUS network working mode is fixed, the corresponding total number of symbol resources is fixed too. For the burst nRT service data transmission application, the actual application corresponding to the PhL symbol resources may be insufficient. An error that causes an application to fail, resulting in data sending failed.

9 Application Layer

9.1 Overview

9.1.1 Protocol architecture

The AUTOBUS AL processes RT and nRT application services based on the AUTOBUS DLL and PhL, provides three kinds of service functions: time service, data service and system management service. The position of AUTOBUS AL in the AUTOBUS protocol stack architecture is shown in Figure 77.

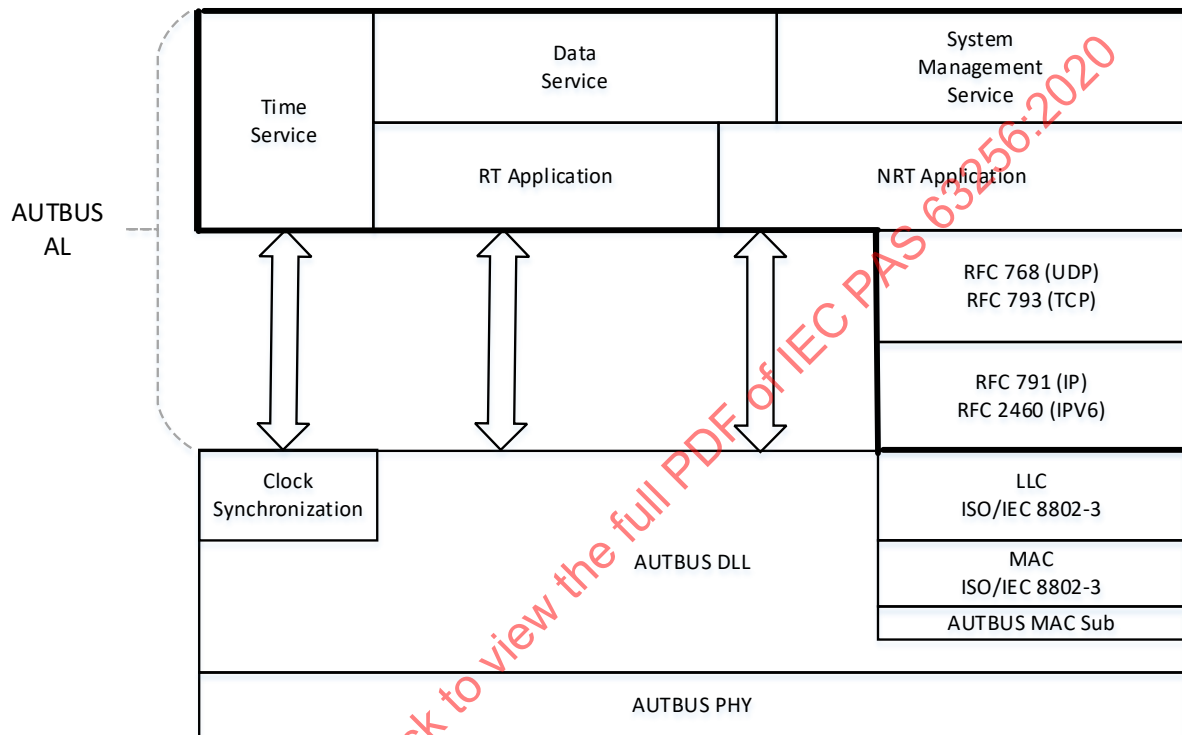


Figure 77 – AL in AUTOBUS protocol stack architecture

The relationship between the International Standards for AUTOBUS application services, AUTOBUS application protocol and systems management is illustrated in Figure 78:

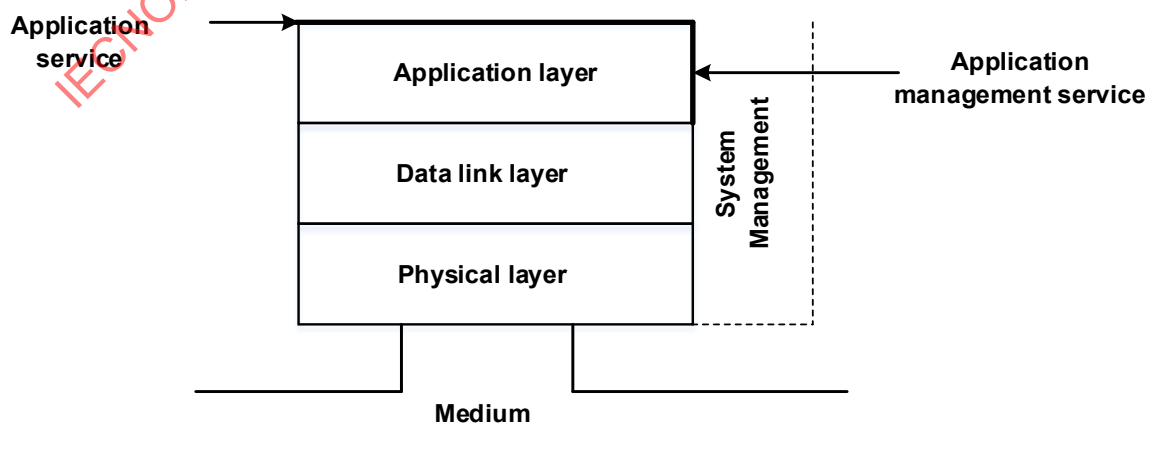


Figure 78 – Relationship of AUTOBUS AL to other fieldbus layers and to users of AUTOBUS application service

A diagram of AUTBUS AL protocol architecture is shown in Figure 79.

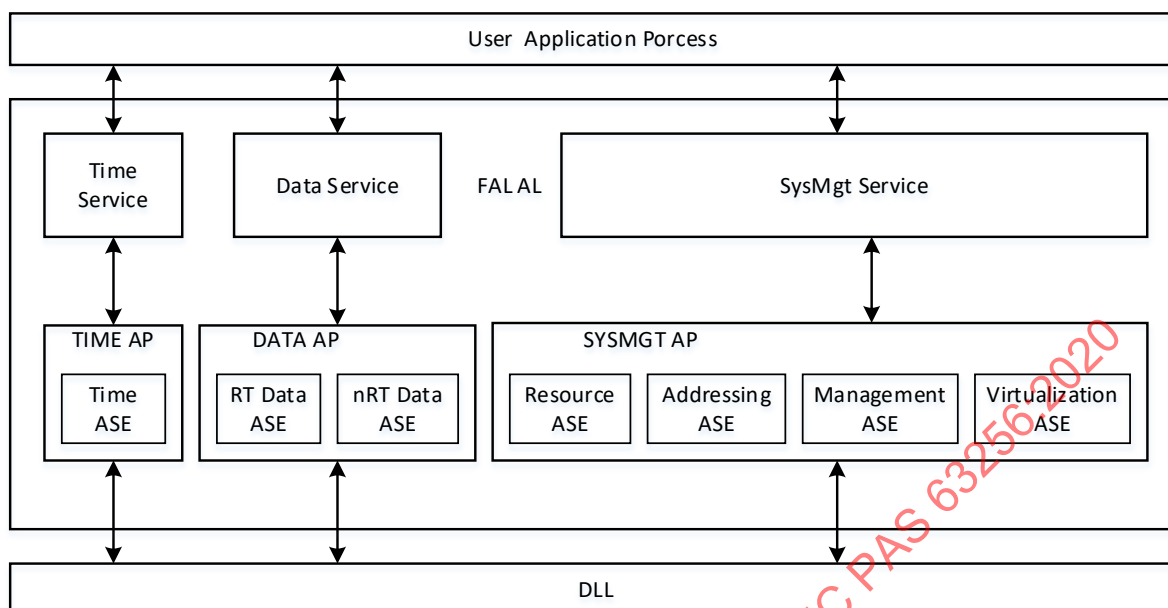


Figure 79 – Diagram of AUTBUS AL protocol architecture

AUTBUS AL provides users with time service, data service and system management service, which are implemented by the application service element (ASE) of the corresponding 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 AUTBUS AL protocol implements the specific AUTBUS AL service functions through the AUTBUS DLL and service interfaces provided by the lower layer.

9.1.2 Data type of AL

9.1.2.1 Structure type

The structure type is defined in 4.3.1. The AUTBUS AL protocol stack contains the following general structure definitions:

9.1.2.1.1 Time information data structure

- Structure name: TIMEINFO_S
- Number of structural members: 5
- Description of structure member: see Table 91

Table 91 – 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.
Date2000	DATE2000	Convert to date value of year, month, day, minute, minute, second, precision is millisecond.
TimeOfDay	TIMEOFDAY	The number of day based on nanosecond.
TimeDiffer	TIMEDIFFER	Time offset value based on nanosecond
NOTE For the use of members of structure TIMEINFO_S, see Table 16.		

9.1.2.1.2 Clock option information structure

- Structure name: CLOCK_OPTION_INFO_S
- Number of structural members: 5
- Description of structure member: see Table 92

Table 92 – 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	ARRAY	Array of UINT8, TCD command for the corresponding clock, the value range of TCD specified in 9.3.2.3
Tid_errcode	ARRAY	ARRAY of UINT8, clock error exception encoding list
NOTE For the use of members of structure CLOCK_OPTION_INFO_S, see Table 16.		

9.1.2.1.3 Network configuration parameter structure

- Structure name: NETWORK_CFG_PARA_S
- Number of structural members: 15
- Description of structure member: see Table 93.

**Table 93 – Description of network configuration parameter structure
NETWORK_CFG_PARA_S**

Member Name	Type of Data	Description
Network_id	UINT32	Identify the NETWORKID of device or module
Ip_switch	UINT8	Whether the IPV4 address or IPV6 address is supported by bit. bit0: whether IPV4 is supported or not; bit1: whether IPV6 is supported, other bits are not used yet; Corresponding bit = 1 indicates support.
Multicast_id0 ~ 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 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
NOTE For the use of members of structure NETWORK_CFG_PARA_S, see Table 17.		

9.1.2.1.4 MAC mapping table data structure

- Structure name: NETWORKID_MAC_MAP_TABLE_S
- Number of structural members: 7
- Description of structure member: see Table 94.

Table 94 – 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, 0 is the control device, 1 is the field device, 2 is the unknown. Other values are reserved.
IP_Flag	UINT8	The device or module supports which type of IP. Bit 0 identifies IPV4 Bit 1 identifies IPV6. The corresponding bit position 1 indicates supporting. Other bits are reserved.
StatusID	UINT8	The status ID of the device or module. The value 0 indicates the initialization status is OK, and the value 1 indicates the normal working status. Other values are reserved.
Network_ID	UINT32	Unique NETWORKID ID of the device or module
MacAddr	ARRAY	MAC address of the device or module

9.1.2.1.5 Mapping table data structure

- Structure name: IP_MAP_TABLE_S
- Number of structural members: 9
- Description of structure member: see Table 95.

Table 95 – 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. 0 is invalid, 1 is valid, 2 is pending. Other values are reserved.
Network_id	UINT32	Unique NETWORKID ID of the device or module
ipv4_addr	UINT32	IPv4 address
ipv4_mask	UINT32	IPv4 mask
ipv6_addr	UINT128	IPv6 address
ipv6_mask	UINT128	IPv6 mask
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.

9.1.3 Data mapping modeling

On the AUTOBUS 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. For the RT data buffer, there are the real-time cyclic (RTC) data buffer and real-time acyclic (RTA) data buffer for the RT data buffer according to the different cyclic processing methods of the data. There are the receiving and transmitting addresses for each type of data buffer. The AP in the AUTOBUS network interacts with the service data, and after data modeling, maps to the corresponding position in the data buffer.

The AUTOBUS AL protocol adopts the object-oriented idea for the APO definition involved in the AP. In the AUTOBUS network environment, 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 80.

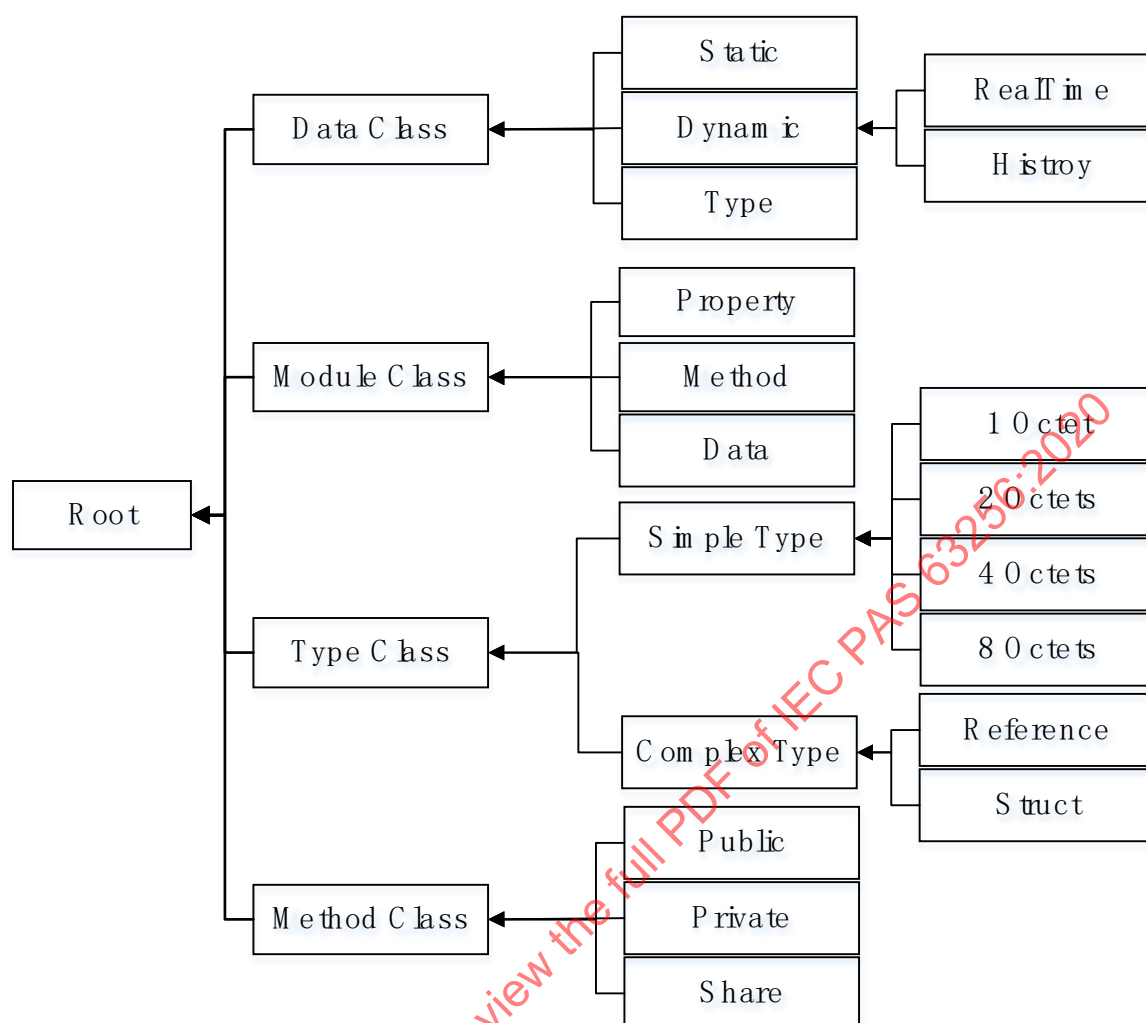


Figure 80 – Data mapping modeling object structure

On the AUTBUS network environment, devices or modules describe specific AP interaction with 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 modeling 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 E for an example of service data mapping for DTU messages.

The data mapping modeling method provided by the AUTBUS 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 AUTBUS 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 F for a data modeling description using a sensor as an example.

Modeling the APO data in the AUTBUS network environment, the actual service data variables are represented by buffer blocks in the data buffer, and each block has certain information such as the block ID and the block length as shown in Figure 81.

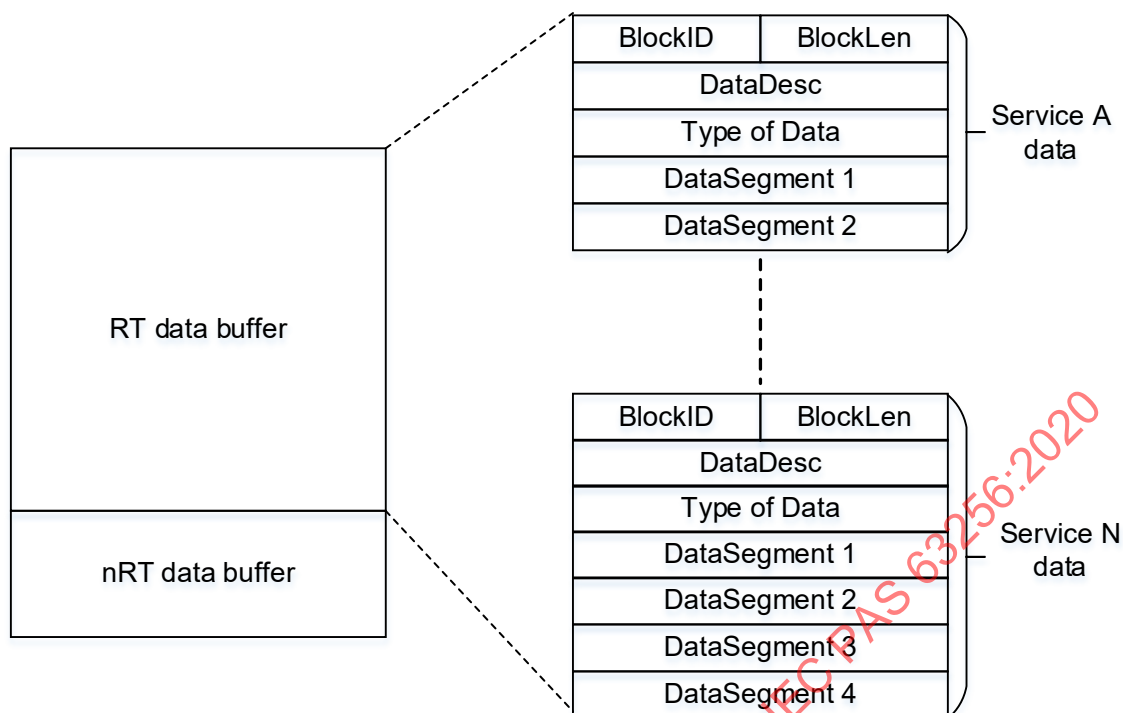


Figure 81 – Diagram of mapping data buffers and service data variables

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 of the block ID 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 AUTOBUS network, the AP interacts with the globally unique DataID to access the buffer block data. The device or module has a unique entry based on the DataID, and the content of the entry is associated with the actual data buffer block data. The entry is a service data mapping table, and its parameters description is shown in Table 96.

Table 96 – 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, 0: RT, 4: nRT, other values are reserved.
DataState	UINT8	Status of the data block, 0: valid, 4: in use, 15: invalid, other values are reserved.
BlockID	UINT32	Data buffer block ID
NetworkID	UINT32	NETWORKID of device or module to which the service data belongs

9.1.4 Communication model

9.1.4.1 General

The Application Process (AP) in the AUTOBUS network environment implements the determined service function objectives through interaction. The interaction between the APs can be implemented based on the request/response method or implemented based on the data/event mode. The communication models of these interaction methods are publisher/subscriber model (P/S) and client/server model (C/S).

9.1.4.2 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 82.

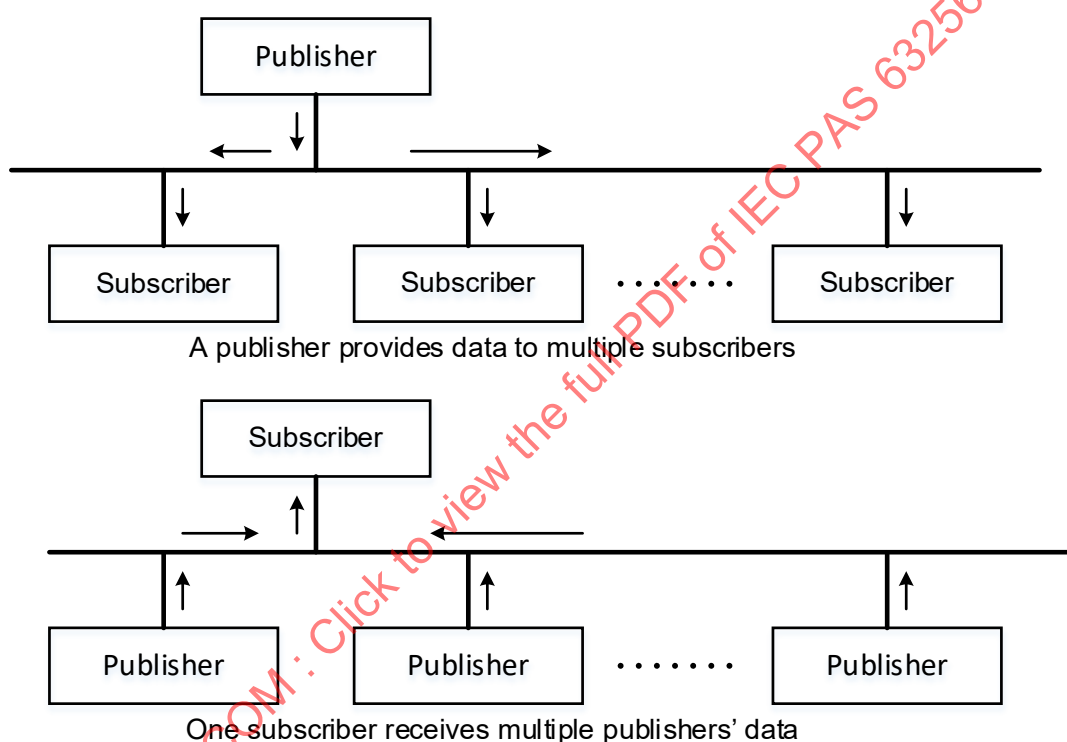


Figure 82 – Diagram of P/S model

On the AUTOBUS 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. On the AUTOBUS network, the P/S-based communication model supports both push (PUSH) mode and pull (PULL) mode as shown in Figure 83 and Figure 84:

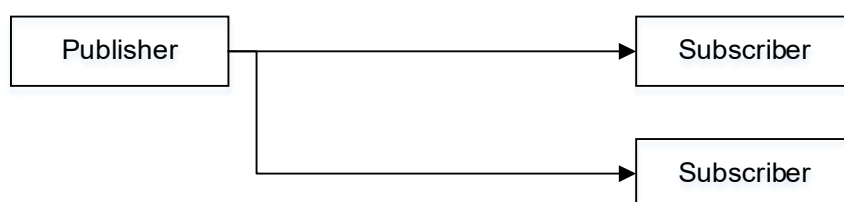


Figure 83 – P/S model of PUSH mode

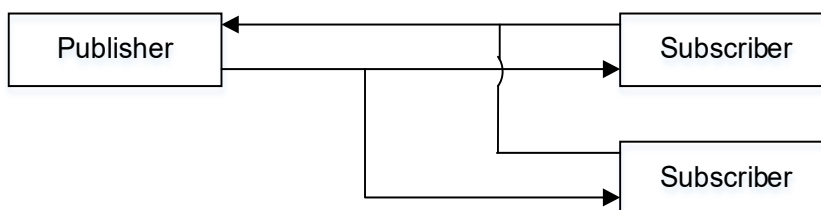


Figure 84 – P/S model of PULL mode

In the 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.

9.1.4.3 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 with confirmed and non-confirmed. If the interaction is 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; if the interaction is a 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 method depends on the specific application interaction service. A C/S communication model diagram is shown in Figure 85.

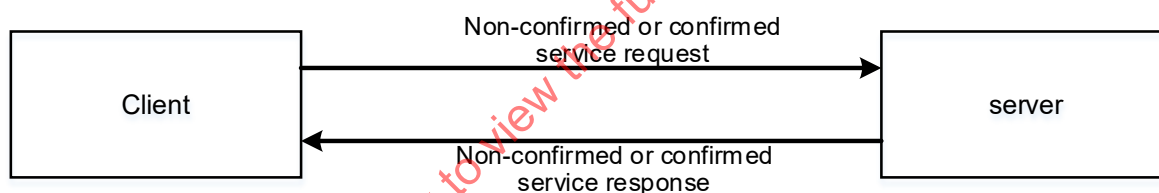


Figure 85 – Diagram of C/S communication model

The server usually stores data centrally (the control command is data too), and the client requests data from the server and receives the response data of the server. In the AUTOBUS network, in the case of different AP interactions, a server can respond to requests from one or more clients, and a client can request and receive data from one or more servers.

9.2 AL Service

9.2.1 Overview

AUTOBUS AL provides users with time service, data service and system management service based on the AUTOBUS DLL service interface. The AL user addresses and accesses devices or modules in the AUTOBUS network through the 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, and virtualization management through resource ASE, addressing ASE, management ASE and virtualized ASE. These application entities implement specific data communication transmission services through C/S or P/S communication models, Table 97 shows the description of the communication model used by AUTOBUS AL ASEs.

Table 97 – Communication models supported by AUTBUS 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 98, the AL user determines whether the service is successful and the reason for the failure according to the return value.

Table 98 – Application service interface return value list

Wrong name	Wrong coding	Wrong description
SUCCESS	0	Service processing succeeded
FAILED	1	Service processing failed
ERR_NOTSUPPORT	2	Device or module does not support
ERR_NOTINITIALIZED	3	Device or module has not been initialized
ERR_NOTEXIST	4	Device or module does not exist, service is unavailable
ERR_NOTREADY	5	Device initialization finishes but no valid status is set
ERR_INVALIDPARA	6	Invalid parameter information
ERR_UNKNOWN	7	Unknown error
ERR_TIMEOUT	8	Service processing timeout

9.2.2 Application Service Elements

9.2.2.1 RT data ASE

9.2.2.1.1 General

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

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 carrying the RT data transmission is changed, the system needs to be restarted to meet the latest RT data service requirement.

9.2.2.1.2 Interface

```
ALS-RT-DATA-Proc (
    Dst_NetworkID,
    Src_NetworkID,
    ProcessID,
    DataID,
    OptionRTLst,
    ALS_Result
)
```

Table 99 defines the RT Data service interface parameters.

Table 99 – 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, as follows:
 - Update (bit 0): Compares and updates data in based on the bit position;
 - 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, overwriting the original data;
 - Write clear (bit6): 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 RT data service returns the value. See the specific return in Table 98.

9.2.2.1.3 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.

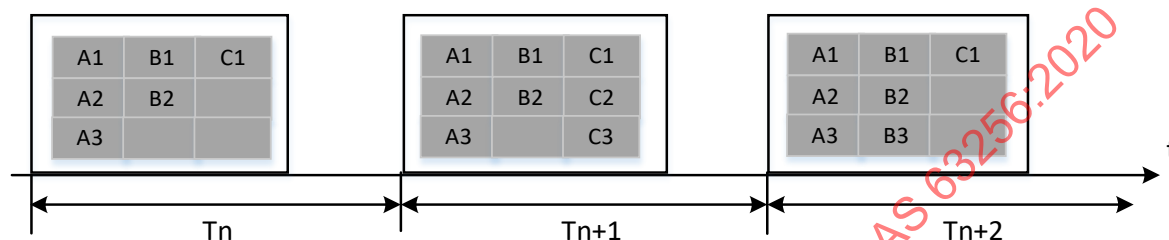


Figure 86 – RT data transmission model

As shown in Figure 86, for the service data blocks A1, A2, A3, B1, B2 and C1, which are always cyclically processed in each fixed period T , that is, RTC data; service data blocks B3, C2 and C3 which are not processed in every cycle. When transmission is required, RT processing is completed within a certain time period; that is RTA data.

Since RTA data are usually control, configuration, alarms or exceptions information, so they are not cyclically processed. See Annex G for an example of a 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, and the server application acquires the request for 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.

9.2.2.2 nRT data ASE

9.2.2.2.1 General

In the AUTBUS network, for nRT application service data, time-critical is not necessary and the lengths of message are 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 number of times the user uses the channel resource is determined by the user when applying, and that number cannot exceed the maximum number which is defined by the MN.

nRT data ASE defines interfaces and related parameters for nRT data processing, providing 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. On the AUTBUS network, whether process nRT data or not is decided by the client. When the nRT data AP interacts based on P/S communication model, the data is transmitted based on the idle default channel resources created by the system. The system schedules the channel resources based on the nRT application service priority, and preferentially allocates the idle channel resources to nRT application service users with high priority.

9.2.2.2.2 Interface

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

Table 100 defines the nRT data service interface parameters.

Table 100 – 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 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 AUTOBUS DLL layer, and see the description of the ProcessID described in RT data ASE chapter for the description of remaining bit.
- DataID: Service data ID, based on which the ID 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, identifies whether the service needs confirmation, 0: the service does not need to be confirmed; 1: 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; 0: no processing; 1: clear release; 2: no clear release;
- ALS_Result: The nRT data service processes the return value. For the specific return values, see Table 98.

9.2.2.2.3 Interaction process

In the AUTOBUS network, when the nRT data AP interaction uses the C/S communication model, the nRT data service interface provides confirmed and non-confirmed methods 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 87, 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.

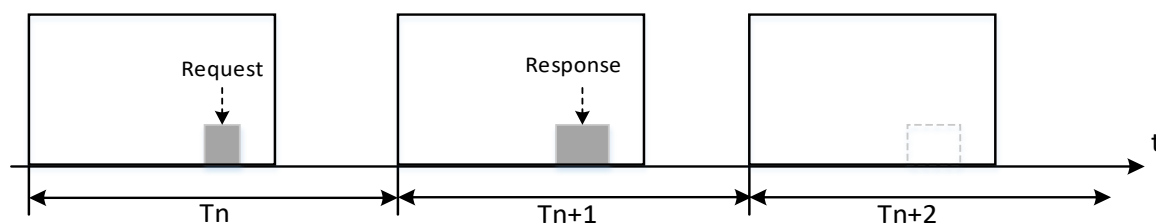


Figure 87 – Diagram of nRT data request response model

When the nRT data AP interaction adopts the C/S communication model, the client application initiates one nRT data service request, and the server application responds to the client application request, 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 88 shows the nRT data AP interaction based on C/S communication mode.

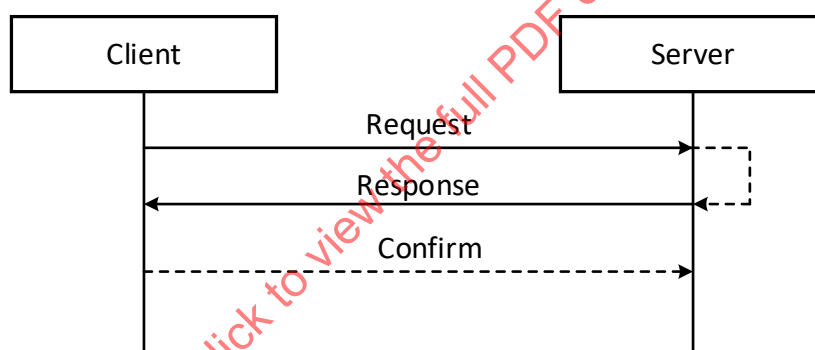


Figure 88 – Diagram of nRT data AP interaction based on C/S communication model

When nRT data AP interaction adopts P/S model, it will schedule based on the priority of nRT data application service, and put nRT data into the idle default channel for data transmission. 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 89, 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.

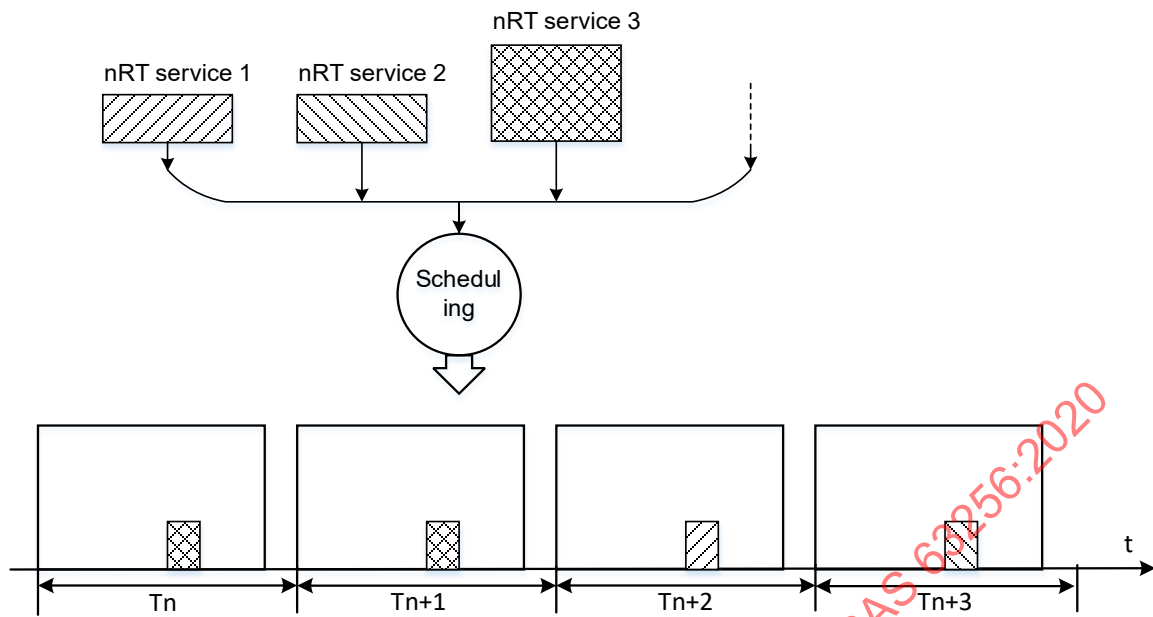


Figure 89 – Diagram of nRT data AP interaction based on P/S communication model

9.2.2.3 Time ASE

9.2.2.3.1 General

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

In the AUTOBUS network, the AL user triggers the time synchronization service at the MN. While the MN will send transmission delay between time information and TN to all TNs, and then the TN will parse the received time information of the MN, 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.

9.2.2.3.2 Interface

```
ALS-TIMING-Proc (
    Dst_NetworkID,
    Src_NetworkID,
    ProcessID,
    OptionClkInfo,
    TimeInfo,
    ALS_Result
)
```

Table 101 defines the timing service interface parameters.

Table 101 – 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:
 - Time synchronization request (0);
 - Time synchronization response (1);
 - Time synchronization exception (2);
 - Time query request (4);
 - Time query response (8);
 - Time query error (10);
- OptionClkInfo: Time option information. When implementing time synchronization service, this parameter is used as an input parameter, 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 92 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 91 for the description of the structure TIMEINFO_S.
- ALS_Result: Time Service processing return value. See Table 98.

9.2.2.3.3 Interaction process

9.2.2.3.3.1 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, 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.

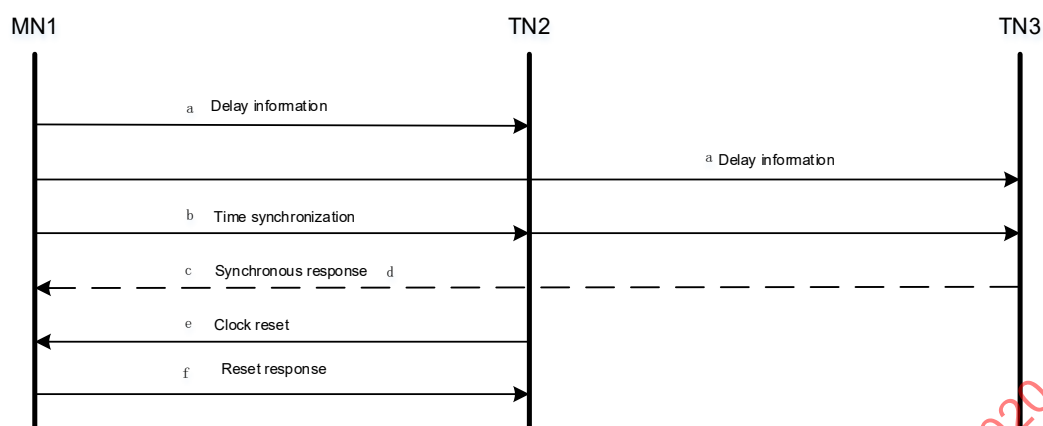


Figure 90 – Time synchronization application interaction process

See Figure 90 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 must 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, 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.

9.2.2.3.3.2 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 91.

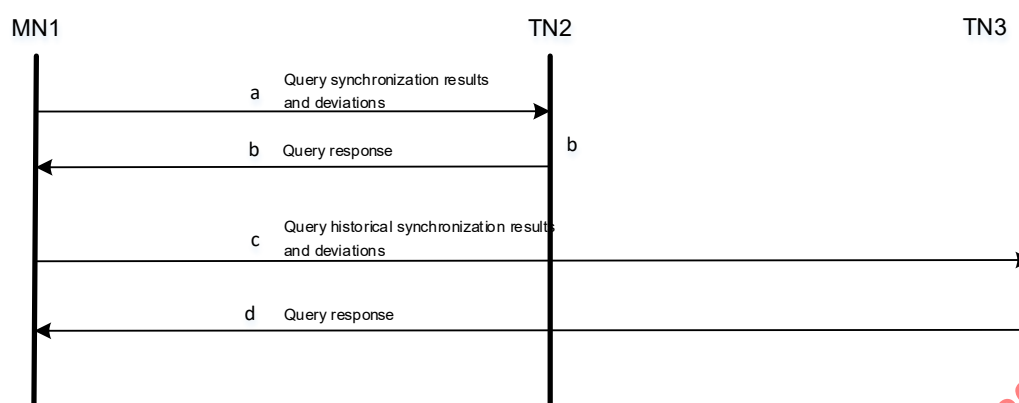


Figure 91 – Time query flow

See Figure 91 the time query is initiated by the MN:

- a) The MN queries the time of the last synchronization of the corresponding TN and the deviation value during synchronization.
- b) 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.
- c) 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 last n synchronization.
- d) The TN replies to the corresponding content of the MN based on the content of the query.

9.2.2.4 Resource ASE

9.2.2.4.1 General

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

Channel resources created based on resource ASE distinguish fixed bandwidth channel resources from variable bandwidth channel resources. The fixed bandwidth channel resources are created during AUTOBUS 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 AUTOBUS 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 and 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.

9.2.2.4.2 Interface

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

Table 102 defines the resource service interface parameters.

Table 102 – 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: A list of resource application service related option information, including the following information:
 - ResType: UINT8 type, resource type, 0: fixed bandwidth resource; 1: 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. For details of return value, see Table 98.

9.2.2.4.3 Interaction process

On the AUTOBUS network as shown in Figure 92, the resource AP interaction is based on the C/S communication model. 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 does not receive the confirmation message from the client application, the server application automatically reclaims resources that have been allocated to the client application.

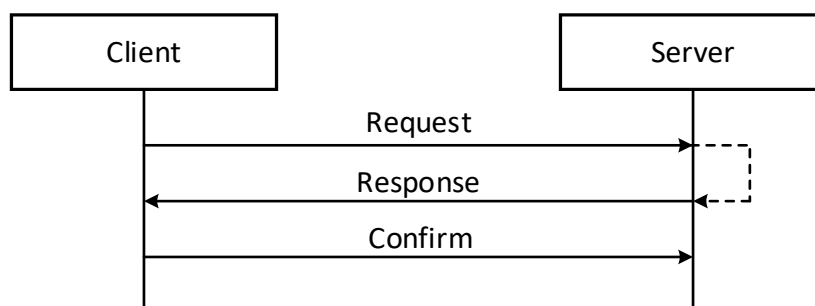


Figure 92 – Resource AP interaction based on C/S communication model

Resource ASE provides the function of writing and reading services based on the DLL service interface to implement local bandwidth resource allocation. Figure 93 shows how the AL user writes and reads the resource application service, triggers the creation, updates and releases of resource channel of the DLL layer.



Figure 93 – Resource ASE local service function Diagram

9.2.2.5 Addressing ASE

9.2.2.5.1 General

The AUTOBUS AL user identifies the device or module by NETWORKID. The AUTOBUS user can also address the device and module based on IP. On the AUTOBUS 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 AUTOBUS AL user implements the interoperation between the AUTOBUS 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 AUTOBUS network environment.

9.2.2.5.1.1 NETWORKID

NETWORKID is a 4-octets value, see Figure 94 Field Description. The AUTOBUS protocol specification AL supports up to eight virtual bus domains. The NETWORKID value is unique for the device or module identified in the specified domain, and the Module ID part is unicast. 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.

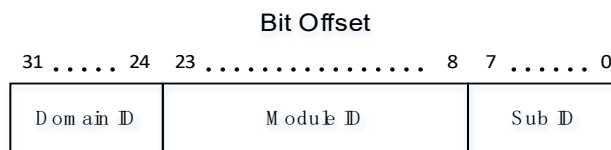


Figure 94 – NETWORKID identification field structure

- 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 ~ 0x08: Identifies the domain ID of a bus domain;
 - 0x09 ~ 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 – 0xDFFF: Identifies valid ID of device or module;
 - 0xE000 – 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. The valid value is 0x00 ~ 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 the AUTOBUS network. For details of the mapping table, see Table 94. This 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.

9.2.2.5.1.2 IP address mapping service

For the AUTOBUS 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 AUTOBUS network and the IP network. For the contents of this mapping table, see Table 95. The device_id in the entry is a keyword. If the device or module is not assigned a NETWORKID, the corresponding entry will 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 manners 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.

9.2.2.5.2 Interface

```
ALS-ADDRESSING-Proc (
    Dst_NetworkID,
    Src_NetworkID,
    ProcessID,
    OptionParaLst,
    ALS_Result
)
```

Table 103 defines the addressing service interface parameters.

Table 103 – Addressing service interface parameters

Parameter name	Data type	Input/output
Dst_NetworkID	UINT32	IN
Src_NetworkID	UINT32	IN
ProcessID	UINT32	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:
 - Set NETWORKID(0);
 - Get NETWORKID(1);
 - Add a MAC mapping table entry (2);
 - Delete the MAC mapping table entry (3);
 - Modify the MAC mapping table entry (4);
 - Query MAC mapping table entries (5)
 - Add an IP mapping table entry (6);
 - Delete the IP mapping table entry (7);
 - Modify the IP mapping table entry (8);
 - Query the IP mapping table entry (9);
- 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. For details about related configuration parameters, see.
 - 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 95 For related member description;

– ALS_Result: Address the service return value, see Table 98 for details of return value.

9.2.2.5.3 Interaction process

9.2.2.5.3.1 NETWORKID interaction process

The addressing ASE implements a Write service function and a Read service 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 95 shows the NETWORKID interaction process diagram.

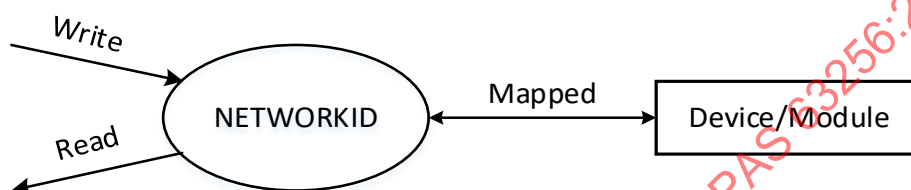


Figure 95 – NETWORKID interaction process diagram

During the initialization and operation of the AUTOBUS 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.

9.2.2.5.3.2 MAC mapping table interaction process

The C/S communication model is used in the AUTOBUS 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 96.

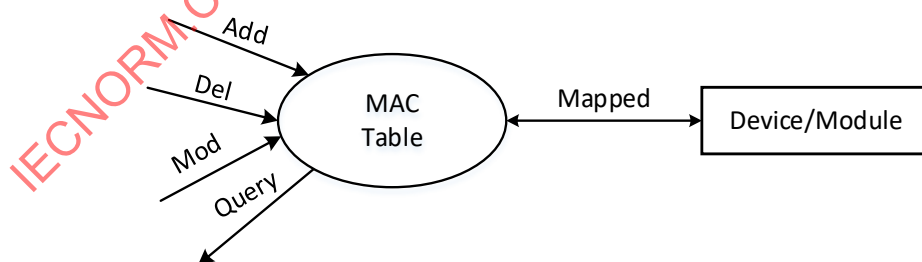


Figure 96 – Diagram of MAC mapping table service function

9.2.2.5.3.3 IP and NETWORKID interaction process

In the AUTOBUS 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 97.