IEC 62601

Edition 1.0 2011-11

INTERNATIONAL STANDARD

colour inside

Industrial communication networks – Fieldbus specifications – WIA-PA communication network and communication profile

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

About the IEC

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

About IEC publications

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

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 It also gives information on projects, withdrawn and replaced publications.

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

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

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

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

IEC 62601

Edition 1.0 2011-11

INTERNATIONAL STANDARD

Industrial communication networks – Fieldbus specifications – WIA-PA communication network and communication profile

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE XH

ICS 25.040.40; 35.100.05

ISBN 978-2-88912-788-7

CONTENTS

FOREWORD 13
1 Scope 15
2 Normative references 15
3 Terms, definitions and abbreviations 15
3.1 Terms and definitions 15
3.2 Abbreviations 19
4 Specification of data types 21
4.1 Representation of boolean type 21
4.2 Representation of unsigned integer type 21
4.3 Representation of octet type 21
4.4 Representation of floating point number type 21
5 WIA-PA overview 22
5.1 Device types 22
5.2 Network topology 22
5.3 Protocol architecture 24
5.4 Interconnection 25
6 System management 25
6.1 General 25
6.2 Framework of system management 27
6.3 Joining process 28
6.3.1 Joining process of routing device 28
6.3.2 Joining process of field device 30
6.3.3 Addressing and address assignment 31
6.4 Virtual Communication Relationship (VCR) 31
6.4.1 Definition 31
6.4.2 Protocol support for VCR 32
6.4.3 VCR establishment 33
6.4.4 VCR release 33
6.5 Routing configuration and communication resource allocation 33
6.5.1 Routing configuration 33
6.5.2 Framework of communication resource allocation 33
6.5.3 DLPDU priority and scheduling rules 34
6.5.4 Communication resource allocation to routing device 35
6.5.5 Communication resource allocation to field device 35
6.6 Aggregation and disaggregation 36
6.6.1 Aggregation 36
6.6.2 Disaggregation 38
6.6.3 An example of the two level aggregation process 39
6.6.4 Management of aggregation and disaggregation objects 41
6.7 Performance monitoring 43
6.7.1 Path failure report 43
6.7.2 Device status report 43
6.7.3 Channel condition report 44
6.8 Leaving process 44
6.8.1 General 44
6.8.2 Leaving process of routing device 44

6.8.3 Leaving process of field device 46
6.9 Management information base and services 47
6.9.1 Management information base 47
6.9.2 MIB services 59
7 Physical Layer 61
8 Data link layer 62
8.1 General 62
8.2 Protocol stack 62
8.3 MAC overview and function extension 62
8.3.1 MAC overview 62
8.3.2 MAC function extension 63
8.4 DLSL function description 66
8.4.1 General 66
8.4.2 Coexistence 67
8.4.3 Timeslot communication 67
8.4.4 WIA-PA superframe 68
8.4.5 Frequency hopping 68
8.4.6 Transmission of long cycle data 70
8.4.7 Retry strategy 71
8.4.8 Management service 71
8.4.9 Radio link quality and channel condition measurement 71
8.4.10 Security 71
8.4.11 DLSL state machine 71
8.5 Data link sub-layer data services 73
8.5.1 General 73
8.5.2 DLDE-DATA.request 73
8.5.3 DLDE-DATA.confirm 74
8.5.4 DLDE-DATA.indication 75
8.5.5 Time sequence of DLSL data service 76
8.6 Data link sub-layer management services 77
8.6.1 General 77
8.6.2 Network discovery services 77
8.6.3 Device joining services 79
8.6.4 Device leaving services 81
8.6.5 DLME-CHANNEL-CONDITION.indication 82
8.6.6 DLME-NEIGHBOR-INFO.indication 83
8.6.7 DLME-COMM-STATUS.indication 83
8.6.8 Keep-alive services 84
8.6.9 Time synchronization services 85
8.7 DLSL frame formats 86
8.7.1 General frame format 86
8.7.2 Date frame format 86
8.7.3 Command frame format 87
9 Network layer 87
9.1 General 87
9.2 Protocol stack 87
9.3 Function description 88
9.3.1 General 88
9.3.2 Addressing 88

9.3.3 Routing 89
9.3.4 Packet lifecycle management 89
9.3.5 Joining and leaving network of device 89
9.3.6 End-to-end network performance monitoring 89
9.3.7 Fragmentation and reassembly 90
9.3.8 Network layer state machine 90
9.4 Network layer data services 91
9.4.1 General 91
9.4.2 NLDE-DATA.request 91
9.4.3 NLDE-DATA.confirm 91
9.4.4 NLDE-DATA.indication 92
9.4.5 Time sequence of NL data services 92
9.5 Network layer management services 93
9.5.1 General 93
9.5.2 Network communication status report services 93
9.5.3 Network joining services 95
9.5.4 Network leaving services 101
9.5.5 Cluster member report services 107
9.5.6 Neighbor information report services 109
9.5.7 Route allocation services 110
9.5.8 Communication resource allocation services 116
9.5.9 Aggregation and disaggregation services 131
9.5.10 Device status report services 132
9.5.11 Channel condition report services 134
9.5.12 Failure path report services 136
9.5.13 Network attribute getting services 137
9.5.14 Network attribute setting services 140
9.6 Network layer packet formats 143
9.6.1 Common packet format 143
9.6.2 Data packet format 144
9.6.3 Aggregated packet format 144
9.6.4 Command packet format 145
10 Application Layer 160
10.1 Overview 160
10.1.1 General 160
10.1.2 AL structure 160
10.1.3 Functions of UAP 161
10.1.4 Functions of ASL 161
10.2 UAP 161
10.2.1 General 161
10.2.2 UAO 162
10.2.3 Method definition 163
10.3 Application sub-layer 166
10.3.1 General 166
10.3.2 Application sub-layer data entity 166
10.4 Application sub-layer packet formats 171
10.4.1 General 171
10.4.2 ASL general packet format 172
10.4.3 Packet formats 173

11 Security 175
11.1 General 175
11.2 Security management framework 175
11.3 Secure communication protocol stack 176
11.3.1 General 176
11.3.2 MAC layer security 177
11.3.3 Data link sub-layer security 177
11.3.4 Application sub-layer security 179
11.4 Key management 180
11.4.1 Key type 180
11.4.2 Key distribution 181
11.4.3 Key update 181
11.4.4 Key status 182
11.5 Secure joining process 182
11.5.1 Secure joining process of a new WIA-PA device 182
11.5.2 Device security material getting services 183
11.6 Secure transportation 187
11.6.1 Process of secure transportation from field device to host configuration computer 187
11.6.2 Process of secure transportation from host configuration computer to field device 188
Annex A (informative) Security strategy for WIA-PA network 189
Annex B (informative) Format description for WIA-PA standard 191
Annex C (informative) Example of UAO 193
Bibliography 195

Figure 1 – Example of WIA-PA physical topology (combination of star and mesh) 23
Figure 2 – Example of WIA-PA physical topology (star-only) 23
Figure 3 – OSI basic reference model mapped to WIA-PA 24
Figure 4 – The architecture of WIA-PA gateway 25
Figure 5 – DMAP in system management 26
Figure 6 – Hybrid centralized and distributed system management scheme 28
Figure 7 – Joining process of routing device through the gateway device 29
Figure 8 – Joining process of routing device through an online routing device 29
Figure 9 – Joining process of field device through a gateway device 30
Figure 10 – Joining process of field device through a routing device 30
Figure 11 – Long address structure of device 31
Figure 12 – Short address structure of routing device 31
Figure 13 – Short address structure of field device 31
Figure 14 – An example of resource allocation 34
Figure 15 – Allocation process of routing device's communication resources 35
Figure 16 – Allocation process of field device's communication resources 36
Figure 17 – Example of aggregation and disaggregation 40
Figure 18 – Process of path failure report 43
Figure 19 – Device status report process of field device 43
Figure 20 – Device status report process of routing device 44

Figure 21 – Process of channel condition report 44
Figure 22 – Active leaving process of routing device 45
Figure 23 – Passive leaving process of routing device 45
Figure 24 – Active leaving process of field device (leaving from gateway device) 46
Figure 25 – Active leaving process of field device (leaving from routing device) 46
Figure 26 – Passive leaving process of field device (leaving from gateway device) 47
Figure 27 – Passive leaving process of field device (leaving from routing device) 47
Figure 28 – WIA-PA DLL protocol stack 62
Figure 29 – WIA-PA DLSL reference model 67
Figure 30 – WIA-PA superframe 68
Figure 31 – R1, R2 and R3 superframe structures 70
Figure 32 – An example of long cycle data transmission 70
Figure 33 – DLSL state machine 72
Figure 34 – Time sequence of data service 76
Figure 35 – Time sequence of network discovery 79
Figure 36 – General frame format 86
Figure 37 – WIA-PA network layer protocol stack 87
Figure 38 – WIA-PA Network layer reference model 88
Figure 39 – Network layer state machine 90
Figure 40 – Time sequence of NL data services 93
Figure 41 – Time sequence for field device joining through routing device 98
Figure 42 – One-hop joining process for routing device 99
Figure 43 – Multi-hop join process of routing device 100
Figure 44 – Active leaving process of field device (leaving routing device) 103
Figure 45 – Passive leaving of field device 104
Figure 46 – Active leaving process of routing device 105
Figure 47 – Passive leaving process of routing device 106
Figure 48 – Cluster member reporting process 108
Figure 49 – Neighbor information reporting process 110
Figure 50 – Time sequence for route adding 112
Figure 51 – Time sequence for route updating 114
Figure 52 – Time sequence for route deleting 116
Figure 53 – Adding a link originated from gateway device to routing device 119
Figure 54 – Adding a link originated from routing device to field device 119
Figure 55 – Updating a link originated by gateway device to routing device 121
Figure 56 – Updating a link originated from routing device to field device 121
Figure 57 – Releasing a link originated from gateway device to routing device 123
Figure 58 – Releasing a link originated from routing device to field device 124
Figure 59 – Adding a superframe originated from gateway device to routing device 126
Figure 60 – Adding a superframe originated from routing device to field device 126
Figure 61 – Updating a superframe originated from gateway device to routing device 128
Figure 62 – Updating a superframe originated from routing device to field device 128
Figure 63 – Releasing a superframe originated from gateway device to routing device 130

Figure 64 – Releasing a superframe originated from routing device to field device 130
Figure 65 – Device status reporting process from field device to routing device 133
Figure 66 – Device status reporting process from routing device to gateway device 134
Figure 67 – Channel condition reporting process from field device to routing device 135
Figure 68 – Channel condition reporting process from routing device to gateway device 136
Figure 69 – Failure path reporting process 137
Figure 70 – Network layer common packet format 143
Figure 71 – Network layer data packet format 144
Figure 72 – Aggregated packet format 144
Figure 73 – Format of NL command packet 145
Figure 74 – AL structure 160
Figure 75 – User application process 161
Figure 76 – C/S communication process 169
Figure 77 – P/S communication process (disable aggregation function) 170
Figure 78 – P/S communication process (enable aggregation function) 171
Figure 79 – R/S communication process 171
Figure 80– Application sub-layer general packet format 172
Figure 81 – ASL data packet format 173
Figure 82 – Acknowledgement packet format 174
Figure 83 – Security management framework of WIA-PA network 175
Figure 84 – Security communication protocol stack 177
Figure 85 – MPDU structure 177
Figure 86 – Security DLPDU structure 178
Figure 87 – Security APDU structure 179
Figure 88 – Key lifecycle 181
Figure 89 – Secure joining process of WIA-PA device 182
Figure 90 – Time sequence for field device joining (Field device to routing device) 185
Figure 91 – Time sequence for field device joining (Routing device to gateway device) 185
Figure 92 – One-hop joining process for routing device 186
Figure 93 – Multi-hop join process of routing device (new routing device to routing device) 186
Figure 94 – Multi-hop join process of routing device (routing device to gateway device) 187
Figure B.1 – Time sequence diagram 191

Table 1 – Protocol support for VCR 32
Table 2 – Relations between VCR and aggregation function 37
Table 3 – Format of aggregated data followed by field device's DAGO 38
Table 4 – Format of aggregated packet followed by routing device's PAGO 38
Table 5 – DAGO class attributes 41
Table 6 – DAGO instance attributes 41
Table 7 – MEM_STRUCT structure 42
Table 8 – PAGO class attributes 42
Table 9 – PAGO instance attributes 42

Table 10 – DGO class attributes 42
Table 11 – DGO instance attributes 43
Table 12 – Unstructured attributes 48
Table 13 – Structured attributes 51
Table 14 – NLRoute_Struct structure 52
Table 15 – Superframe_Struct structure 52
Table 16 – Link_Struct structure 53
Table 17 – Neighbor_Struct structure 54
Table 18 – ChanCon_Struct structure 54
Table 19 – Device_struct structure 55
Table 20 – VCR_Struct structure 57
Table 21 – DevConRep_Struct structure 58
Table 22 – Key_Struct structure 58
Table 23 – ObjList_Struct structure 59
Table 24 – DMAP-MIB-GET.request parameters 59
Table 25 – DMAP-MIB-GET.confirm parameters 60
Table 26 – DMAP-MIB-SET.request parameters 61
Table 27 – DMAP-MIB-SET.confirm parameters 61
Table 28 – MAC extended PIB attributes 63
Table 29 – MAC extended command frame 63
Table 30 – MLME-KEEP-LIVE.confirm parameters 64
Table 31 – MLME-KEEP-LIVE.indication parameters 64
Table 32 – MLME-TIME-SYN.request parameters 64
Table 33 – MLME-TIME-SYN.confirm parameters 65
Table 34 – MLME-TIME-SYN.indication parameters 65
Table 35 – Beacon payload 65
Table 36– Format of keep-alive command frame 66
Table 37– Format of time synchronization command frame 66
Table 38 – Hopping mechanisms 69
Table 39 – DLSL state transitions 72
Table 40 – DLDE-DATA.request parameters 74
Table 41 – DLDE-DATA.confirm parameters 75
Table 42 – Status table 75
Table 43 – DLDE-DATA.indication parameters 76
Table 44 – DLME-DISCOVERY.request parameters 77
Table 45 – DLME- DISCOVERY.confirm parameters 78
Table 46 – Network descriptor list 78
Table 47 – DLME-JOIN.request parameters 79
Table 48 – DLME-JOIN.indication parameters 80
Table 49 – DLME-JOIN.response parameters 80
Table 50 – DLME-JOIN.confirm parameters 81
Table 51 – DLME-LEAVE.request parameters 81
Table 52 – DLME-LEAVE.indication parameters 82

Table 53 – DLME-LEAVE.confirm parameters 82
Table 54 – DLME-CHANNEL-CONDITION.indication parameters 83
Table 55 – DLME-NEIGHBOR-INFO.indication parameters 83
Table 56 – DLME-COMM-STATUS.indication parameters 84
Table 57 – DLME -KEEP-LIVE.confirm parameters 84
Table 58 – DLME -KEEP-LIVE.indication parameters 85
Table 59 – DLME-TIME-SYN.request parameters 85
Table 60 – DLME -TIME-SYN.confirm parameters 85
Table 61 – DLME-TIME-SYN.indication parameters 86
Table 62 – DLSL frame control filed 86
Table 63 – Date frame format 87
Table 64 – General command frame format 87
Table 65 – DLSL command frame 87
Table 66 – Example of a routing table 89
Table 67 – Network layer states 90
Table 68– NL state transitions 90
Table 69 – NLDE-DATA.request parameters 91
Table 70 – NLDE-DATA.confirm parameters 92
Table 71 – NLDE-DATA.indication parameters 92
Table 72 – NLME-COMM-STATUS.request parameters 93
Table 73 – NLME-COMM-STATUS.indication parameters 94
Table 74 – NLME-COMM-STATUS.confirm parameters 94
Table 75 – NLME-JOIN.request parameters 95
Table 76 – NLME-JOIN.indication parameters 96
Table 77 – NLME-JOIN.response parameters 96
Table 78 – NLME-JOIN.confirm parameters 97
Table 79 – NLME-LEAVE.request parameters 101
Table 80 – NLME-LEAVE.indication parameters 101
Table 81 – NLME-LEAVE.response parameters 101
Table 82 – NLME-LEAVE.confirm parameters 102
Table 83 – NLME-RPT-CLRMEM.request parameters 107
Table 84 – NLME-RPT-CLRMEM.confirm parameter 107
Table 85 – NLME-RPT-CLRMEM.response parameters 108
Table 86 – NLME-NEIGHBOR-INFO.request parameters 109
Table 87 – NLME-NEIGHBOR-INFO.confirm parameter 110
Table 88 – NLME-ADD_ROUTE.request parameters 110
Table 89 – NLME-ADD_ROUTE.confirm parameters 111
Table 90 – NLME-UPDATE_ROUTE.request parameters 112
Table 91 – NLME-UPDATE_ROUTE.confirm parameter 113
Table 92 – NLME-UPDATE_ROUTE.request parameters 114
Table 93 – NLME-DELETE_ROUTE.confirm parameters 115
Table 94 – NLME-ADD-LINK.request parameters 117
Table 95 – NLME-ADD-LINK.confirm parameters 118

Table 96 – NLME-UPDATE-LINK.request parameters 120
Table 97 – NLME-UPDATE-LINK.confirm parameters 120
Table 98 – NLME-RELEASE-LINK.request parameters 122
Table 99 – NLME-RELEASE-LINK.confirm parameters 122
Table 100 – NLME-ADD-SFR.request parameters 124
Table 101 – NLME-ADD-SFR.confirm parameters 125
Table 102 – NLME-UPDATA-SFR.request parameters 127
Table 103 – NLME-UPDATE-SFR.confirm parameters 127
Table 104 – NLME-RELEASE-SFR.request parameters 129
Table 105 – NLME-RELEASE-SFR.confirm parameters 129
Table 106 – NLME-AGG.indication parameters 131
Table 107 – NLME-AGO-SEND.request parameters 131
Table 108 – NLME-DAG.indication parameter 132
Table 109 – NLME- DEVICE -STATUS.request parameters 132
Table 110 – NLME- DEVICE -STATUS.indication parameters 133
Table 111 – NLME- DEVICE -STATUS.confirm parameter 133
Table 112 – NLME-CHANNEL-CONDITION.request parameters 134
Table 113 – NLME-CHANNEL-CONDITION.indication parameters 135
Table 114 – NLME-CHANNEL-CONDITION.confirm parameter 135
Table 115 – NLME-PATH_FAILURE.request parameters 136
Table 116 – NLME-PATH_FAILURE.indication parameters 137
Table 117 – NLME-PATH_FAILURE.confirm parameters 137
Table 118 – NLME-INFO_GET.request parameters 138
Table 119 – NLME-INFO_GET. indication parameters 139
Table 120 – NLME-INFO_GET.response parameters 139
Table 121 – NLME-INFO_GET.response parameters 140
Table 122 – NLME-INFO_SET.request parameters 141
Table 123 – NLME-INFO_SET.indication parameters 141
Table 124 – NLME-SET. response parameters 142
Table 125 – NLME-SET.confirm parameters 143
Table 126– Control field format 143
Table 127 – Network layer command packet 146
Table 128 – Execution results of commands 147
Table 129 – Format of joining request packet 147
Table 130 – Format of joining response packet 147
Table 131 – Format of communication status report request packet 148
Table 132 – Format of leaving request packet 148
Table 133 – Value of leaving reason 148
Table 134 – Format of leaving response packet 149
Table 135 – Format of cluster member report request packet 149
Table 136 – Format of cluster member report response packet 149
Table 137 – Format of neighbor information report request packet 150
Table 138 – Format of route adding request packet 150

Table 139 – Format of route adding response packet 150
Table 140 – Format of route update request packet 151
Table 141 – Format of route update response packet 151
Table 142 – Format of route deleting request packet 151
Table 143 – Format of route deleting response packet 152
Table 144 – Format of link adding request packet 152
Table 145 – Format of link adding response packet 152
Table 146 – Format of link update request packet 153
Table 147 – Format of link update response packet 153
Table 148 – Format of link release request packet 154
Table 149 – Format of link release response packet 154
Table 150 – Format of superframe adding request packet 154
Table 151 – Format of superframe adding response packet 155
Table 152 – Format of superframe update request packet 155
Table 153 – Format of superframe update response packet 155
Table 154 – Format of superframe release request packet 156
Table 155 – Format of superframe release response packet 156
Table 156 – Format of device condition report request packet 156
Table 157 – Format of device condition information 157
Table 158 – Format of channel condition report request packet 157
Table 159 – Format of channel quality information 157
Table 160 – Format of path failure report request packet 158
Table 161 – Format of attribute getting request packet 158
Table 162 – Format of attribute getting response packet 159
Table 163 – Format of attribute setting request packet 159
Table 164 – Format of attribute setting response packet 160
Table 165 – Method definition 163
Table 166 – Request format of READ 163
Table 167 – Response format of READ method 163
Table 168 – Request format of WRITE method 164
Table 169 – Response format of WRITE method 164
Table 170 – Format of PUBLISH method 165
Table 171 – Format of REPORT method 165
Table 172 – Format of REPORT ACK method 165
Table 173 – ASLDE-DATA.request parameters 167
Table 174 – ASLDE-DATA.confirm parameters 167
Table 175 – ASLDE-DATA.indication parameters 168
Table 176 – ASLDE-AGG.request parameters 168
Table 177 – ASLDE-DAG.indication parameters 169
Table 178 – Packet control field format 172
Table 179 – Packet type subfield value 172
Table 180 – Structure of DLSL security header 178
Table 181 – Structure of security control field in DLSL security header 179

Table 182 – Structure of security material control field in DLSL security header 179
Table 183 – Structure of ASL security header field 180
Table 184 – DLME-SEC.request parameters 183
Table 185 – DLME-SEC.indication parameters 183
Table 186 – DLME-SEC.response parameters 184
Table 187 – DLME-SEC.confirm parameters 184
Table A.1 – Graded and layered security measures for WIA-PA network 190
Table A.2 – Security levels of data packets 190
Table B.1 – Packet or frame format in octet(s) 191
Table B.2 – Subfield format in bit(s) 192
Table C.1 – AIO Class Attribute 193
Table C.2 – AIO Instance Attributes 194

INTERNATIONAL ELECTROTECHNICAL COMMISSION

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS – WIA-PA COMMUNICATION NETWORK AND COMMUNICATION PROFILE

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.

International Standard IEC 62601 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial process measurement, control and automation.

This standard cancels and replaces IEC/PAS 62601 published in 2009. This first edition constitutes a technical revision.

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

FDIS	Report on voting
65C/663/FDIS	65C/671/RVD

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

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

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

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

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS – WIA-PA COMMUNICATION NETWORK AND COMMUNICATION PROFILE

1 Scope

This International Standard specifies the system architecture and the communication protocol of Wireless networks for Industrial Automation – Process Automation (WIA-PA) built on IEEE STD 802.15.4-2006.

2 Normative references

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

IEC/TS 61804-2: 2006, *Function blocks (FB) for process control – Part 2: Specification of FB concept*

IEEE STD 802.15.4-2006, *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

3.1.1
absolute timeslot number
number of timeslots from the start of the network, generally denoting the current timeslot. Its value increases by one, and does not decrease. Its current value is always the sequence number of the current timeslot. Its maximum value is (2^{48}-1). After the maximum value, it re-counts from zero

3.1.2
active leaving
process by which an online field device is allowed to leave the network through applying to its routing device or by which an online routing device is allowed to leave the network through applying to the gateway device

3.1.3
adaptive frequency hopping
change of communication channels according to actual condition of channels in every timeslot during the intra-cluster period of WIA-PA superframe

3.1.4
adaptive frequency switch
change of communication channels according to the actual condition of channels during the beacon frame and active period in a superframe cycle, and using different channels in different superframe cycles

3.1.5
aggregation
merging of data from multiple user application objects or merging of several packets from cluster members into one packet

3.1.6
application sub-layer

protocol sub-layer that provides data and management services for the application layer

3.1.7
beacon
special frame broadcast by the routing devices or the gateway device in the WIA-PA network

NOTE To join the WIA-PA network, a new routing device or field device should first listen to beacons.

3.1.8
broadcast
sending one packet to all WIA-PA devices simultaneously

3.1.9
channel
RF medium used to convey a packet from a sender to a receiver

3.1.10
cluster
logical group of devices that is comprised of a routing device and field devices

3.1.11
cluster head
manager in a cluster, performed by the routing device

3.1.12
cluster member
data source in a cluster, performed by a field device

3.1.13
coexistence
ability of one network to perform its task in a given shared environment without disturbing or being disturbed by other networks

NOTE These networks may or may not have the same set of rules.

3.1.14
communication resources
channels and timeslots used to transport frames

3.1.15
WIA-PA configuration software
software tools for configuring the WIA-PA network and devices

3.1.16
data link sub-layer
upper layer of the IEEE STD 802.15.4-2006 MAC layer, used to handle the aspects of network topology and communication resources in the WIA-PA network

3.1.17
disaggregation
dividing the merged packet into data of user application objects

3.1.18
field device
device installed in the field, which is connected to or controls the process

3.1.19
frame
format of aggregated bits from the medium access control (MAC) entity that are transmitted together in time

[IEEE STD 802.15.4-2006]

3.1.20
frequency hopping
change of transmitting/receiving frequency to combat interference and fading

3.1.21
gateway device
device connecting the WIA-PA network to other plant networks

3.1.22
handheld device
portable device used for provisioning, firmware updating, and device status monitoring

3.1.23
hop
movement of a packet directly between two adjacent neighbor devices in one network transaction without the participation of any other devices in the WIA-PA network

NOTE Multiple hops are used to lengthen the transmission distance, bypass interference sources or avoid obstructions.

3.1.24
host configuration computer
device through which users and maintenance/management personnel perform transactions on the WIA-PA network and the management networks

3.1.25
interoperability
ability of two or more network systems to exchange information and to make mutual use of the information that has been exchanged

[ISO/IEC TR 10000-1: 1998, 3.2.1, modified]

3.1.26
joining
process by which a WIA-PA device is authenticated and allowed to participate in the WIA-PA network

3.1.27
link
set of communication parameters necessary to transport a frame between adjacent devices in the network

NOTE It includes source/destination address, timeslot, channel, direction, and link type.

3.1.28
mesh
topology formed by routing devices and the gateway device in the WIA-PA network

NOTE One routing device may connect to the gateway device and more than one other routing device.

3.1.29
multicast
sending one packet to a group of WIA-PA devices simultaneously

3.1.30
network address
16-bit unsigned integer uniquely identifying the device in the WIA-PA network; also called short address

NOTE The most significant 8 bits of the network address, assigned by the network manager, identify different clusters.

3.1.31
network manager
logical role for configuring the network, allocating the communication resources, managing the routing tables, and monitoring and reporting the health of the network

3.1.32
packet
formatted, aggregated bits that are transmitted together in time across the physical medium

[IEEE STD 802.15.4-2006, 3.31]

3.1.33
packet lifecycle
maximal packet survival time from being generated to being dropped

3.1.34
passive leaving
process by which an online field device is instructed by its routing device to leave the network or by which an online routing device is instructed by the gateway device to leave the network

3.1.35
physical address
EUI-64 bits uniquely identifying the device in the WIA-PA network; also called long address

NOTE A physical address is assigned by the manufacturers.

3.1.36
routing device
device forwarding packets from one WIA-PA device to another in the WIA-PA network

3.1.37
security manager
logical role for configuring the security strategies of the whole network, managing keys, and authenticating devices

3.1.38
superframe
collection of timeslots repeating over time

NOTE It specifies the transmitting or receiving time of periodic communication.

3.1.39
timeslot
basic time unit of data exchange

NOTE Its duration is configurable in the WIA-PA network.

3.1.40
timeslot hopping
regular change of transmitting/receiving frequency per timeslot to avoid interference and fading

3.1.41
unicast
sending one packet to a single device in the WIA-PA network

3.1.42
virtual communication relation
communication paths and communication resources between two user application objects

3.1.43
WIA-PA device
device in the WIA-PA network, e.g. the host configuration computer, gateway device, routing device, field device or handheld device

3.2 Abbreviations

ACK	Acknowledge
AFH	Adaptive Frequency Hopping
AFS	Adaptive Frequency Switch
AIO	Analog Input Object
AL	Application Layer
AOO	Analog Output Object
ASDU	Application Service Data Unit
ASL	Application Sub-Layer
ASLDE	Application Sub-Layer Data Entity
ASLDE-SAP	ASLDE Service Access Point
ASLME	Application Sub-Layer Management Entity
ASLME-SAP	ASLME Service Access Point
ASLPDU	Application Sub-Layer Protocol Data Unit
ASN	Absolute timeSlot Number
CAP	Contention Access Period
CFP	Contention Free Period
C/S	Client/Server
CSMA	Carrier Sense Multiple Access
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
DAGO	Data AggreGation Object
DGO	DisaGgregation Object
DIO	Digital Input Object
DLDE	Data Link Sub-Layer Data Entity
DLDE-SAP	DLDE Service Access Point
DLL	Data Link Layer
DLME	Data Link Sub-Layer Management Entity
DLME-SAP	DLME Service Access Point
DLPDU	Data Link Sub-Layer Protocol Data Unit
DLSL	Data Link Sub-Layer

DMAP	Device Management Application Process
DOO	Digital Output Object
ENC	ENCryption
EUI-64	Extended Unique Identifier-64 bits
FCS	Frame Check Sequence
FDMA	Frequency Division Multiple Access
FFD	Full-Function Device
FH	Frequency Hopping
GTS	Guaranteed Time Slot
GW	GateWay device
ID	IDentifier
IDS	Intrusion Detection System
KED	Data Encryption Key
KEK	Key Encryption Key
KJ	Join Key
KP	Provision Key
LME-SAP	Layer Management Entity Service Access Point
LSB	Least Significant Bit
LQI	Link Quality Indication
MAC	Medium Access Control sub-layer
MCPS	MAC Common Part Sub-layer
MHR	Medium Access Control Header
MIB	Management Information Base
MIC	Message Integrity Code
MLDE	MAC sub-Layer Data Entity
MLDE-SAP	MLDE Service Access Point
MLME	MAC sub-Layer Management Entity
MLME-SAP	MLME Service Access Point
MPDU	MAC Protocol Data Unit
MSB	Most Significant Bit
NL	Network Layer
NLDE	Network Layer Data Entity
NLDE-SAP	NLDE Service Access Point
NLME	Network Layer Management Entity
NLME-SAP	NLME Service Access Point
NM	Network Manager
NPDU	Network Protocol Data Unit
NSDU	Network Service Data Unit
PAGO	Packet AGgregation Object
PAN	Personal Area Network
PHY	PHYsical layer
PIB	PAN Information Base
P/S	Publisher/Subscriber

RFD	Reduced-Function Device
R/S	Report source/Sink
SAP	Service Access Point
SM	Security Manager
SMK	Symmetric Master Key
TDMA	Time Division Multiple Access
TH	Timeslot Hopping
UAO	User Application Object
UAP	User Application Process
UAPME-SAP	UAP Management Entity SAP
UTC	Universal Time Coordinated
VCR	Virtual Communication Relationship
VCR_ID	Virtual Communication Relationship IDentifier
WIA-PA	Wireless network for Industrial Automation – Process Automation

4 Specification of data types

4.1 Representation of boolean type

Boolean :: = BOOLEAN -- non 0: TRUE

-- 0: FALSE

4.2 Representation of unsigned integer type

Unsigned8 :: = INTEGER(0..255) -- integer range: $0 <= i <= 2^8-1$

Unsigned16 :: = INTEGER(0..65 535) -- integer range: $0 <= i <= 2^{16}-1$

Unsigned24 :: = INTEGER(0.. 2^{24}-1) -- integer range: $0 <= i <= 2^{24}-1$

Unsigned32 :: = INTEGER(0.. 2^{32}-1) -- integer range: $0 <= i <= 2^{32}-1$

Unsigned40 :: = INTEGER(0.. 2^{40}-1) -- integer range: $0 <= i <= 2^{40}-1$

Unsigned48 :: = INTEGER(0.. 2^{48}-1) -- integer range: $0 <= i <= 2^{48}-1$

Unsigned64 :: = INTEGER(0.. 2^{64}-1) -- integer range: $0 <= i <= 2^{64}-1$

4.3 Representation of octet type

Octetstring :: = OCTET STRING -- 8-bit string

4.4 Representation of floating point number type

Float :: = BIT STRING SIZE (4) -- single precision, range references to

IEEE STD 754 Short Real Number (32 Bit)

5 WIA-PA overview

5.1 Device types

The document specifies five types of WIA-PA devices:

a) host configuration computer;

b) gateway device (GW);

c) routing device;

d) field device; and

e) handheld device.

To improve reliability, there may be redundant gateway devices and redundant routing devices in the WIA-PA network. A primary device connects its redundant device in the wired manner. The start-up time of a redundant device is not later than that of the primary device. Redundant devices do not start up their radio modules. When the information of primary devices is changed, the primary devices should timely backup the changed information to their redundant devices in the wired manner. The wired manner is not specified in this document.

5.2 Network topology

WIA-PA network supports two different types of network topologies:

a) a hierarchical network topology that combines star and mesh, and

b) a star-only network topology.

The hierarchical network topology that combines star and mesh is illustrated in Figure 1. The first level of the network is in mesh topology, where routing devices and gateway devices are deployed. The second level of the network is in star topology, where routing devices, field devices, and handheld devices (if they exist) are deployed.

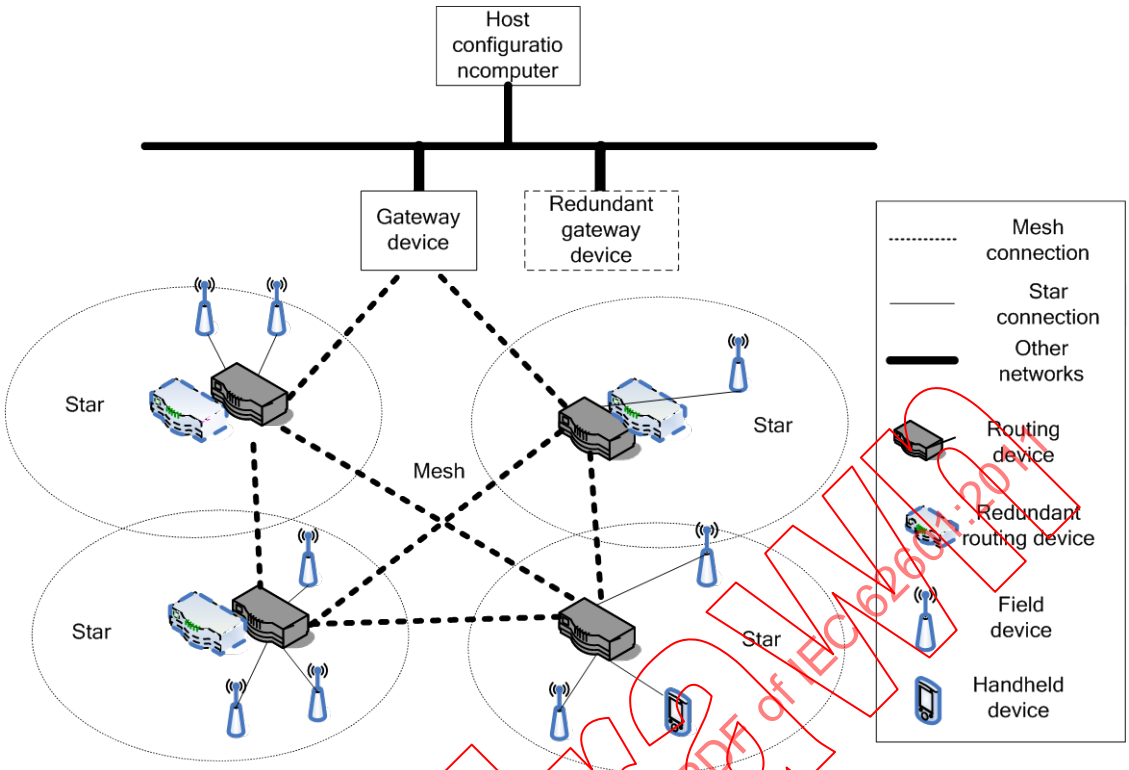

Figure 1 – Example of WIA-PA physical topology (combination of star and mesh)

The star-only network is comprised of a gateway device, field devices, and handheld devices (if they exist). The star network topology is illustrated in Figure 2.

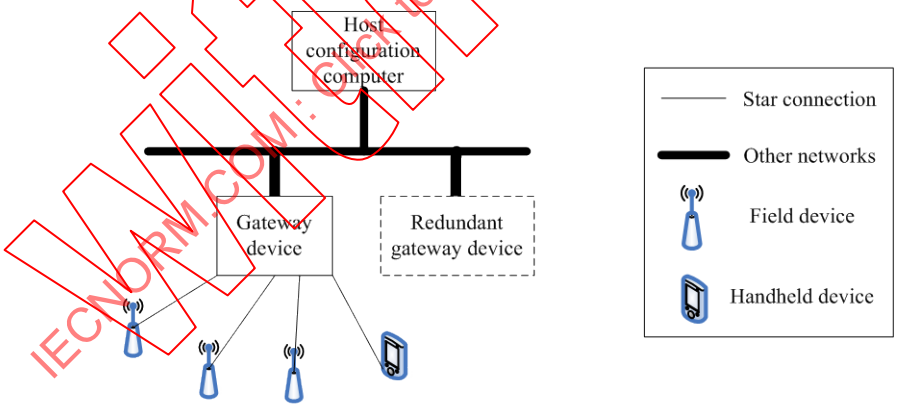

Figure 2 – Example of WIA-PA physical topology (star-only)

NOTE 1 The star-only network is a special case of the hierarchical network. In the rest of this document, the specification of the hierarchical network covers that of the star-only network. Therefore, the specification of the star-only network will not be specially described.

In order to facilitate management, this document specifies five kinds of logical roles:

a) Gateway

Gateway handles protocol-translation and data-mapping between the WIA-PA network and other networks.

b) Network manager

Network Manager (NM) manages and monitors the entire network (see 6.2). There should be one and only one network manager per WIA-PA network.

c) Security manager

Security Manager (SM) deals with security key management and security authentication of gateway devices, routing devices, field devices, and handheld devices (if they exist).

d) Cluster head

Cluster head manages and monitors field devices and handheld devices (if they exist). Cluster head also merges and securely forwards packets from local cluster members and other cluster heads.

e) Cluster member

Cluster member collects field data and sends the data to its cluster head.

The NM and the SM that are used for system management should reside in a gateway device.

One physical device may perform the functions of several logical roles. In the hierarchical network that combines star and mesh, a gateway device may perform the logical roles of gateway, NM, SM, and cluster head. A routing device should act as a cluster head. A field device/handheld device should only act as a cluster member.

NOTE 2 The primary device is marked by the parameter "PrimaryDevAddr" and the redundant device is marked by the parameter "RedundantDevFlag" in Table 19.

5.3 Protocol architecture

The WIA-PA protocol architecture, which is illustrated in Figure 3, is based on ISO/IEC 7498-1. The WIA-PA protocol architecture defines the Data Link Sub-Layer (DLSL), Network Layer (NL) and Application Layer (AL), while its PHYsical layer (PHY) and Medium Access Control sub-layer (MAC) are based on IEEE STD 802.15.4-2006.

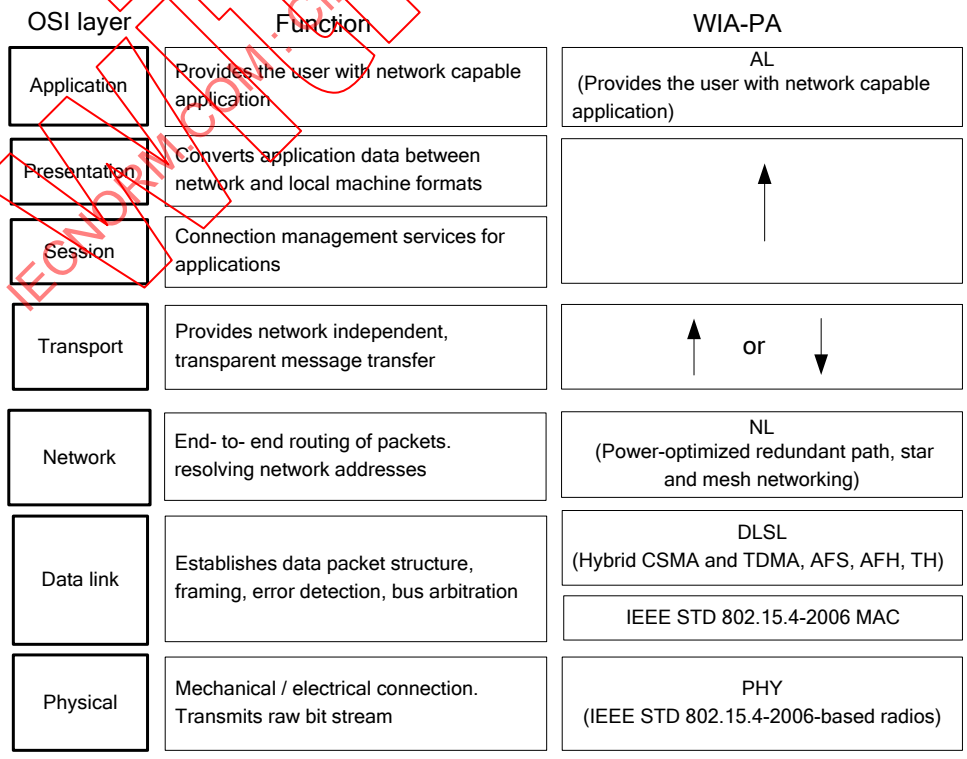

Figure 3 – OSI basic reference model mapped to WIA-PA

5.4 Interconnection

The WIA-PA network interconnects with other networks through the WIA-PA gateway. Besides the communication to the WIA-PA NM and SM, the WIA-PA gateway may communicate with other WIA-PA devices in order to exchange information between devices. Meanwhile, the WIA-PA gateway may connect other networks, such as wired fieldbuses. The architecture of the WIA-PA gateway is shown in Figure 4.

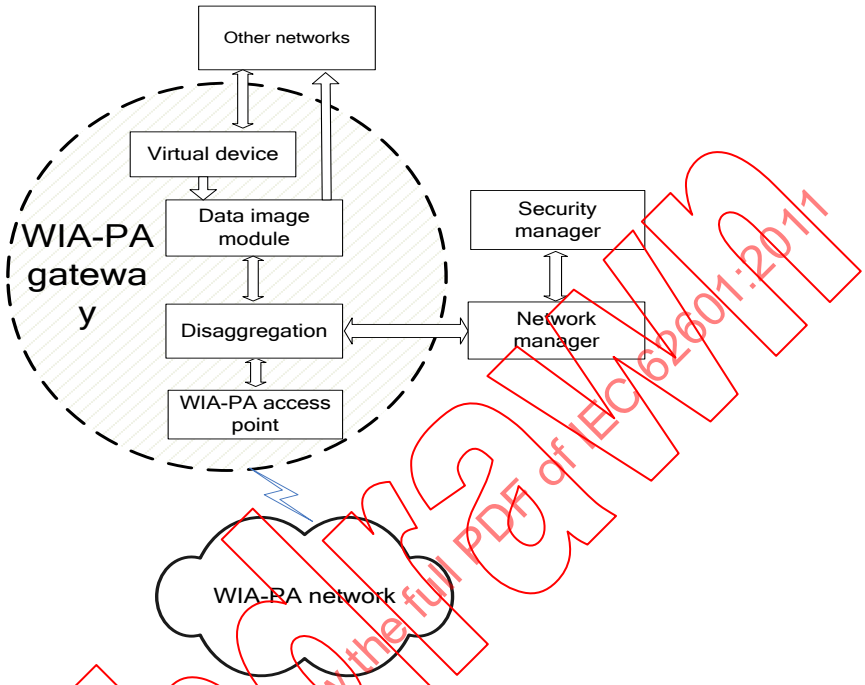

Figure 4 – The architecture of WIA-PA gateway

The WIA-PA gateway includes the following components:

a) WIA-PA access point

The WIA-PA access point realizes the physical connections of the WIA-PA network and transmission of the management information and data.

b) Virtual device

The virtual device defines a communication interface for other networks. The interface is used to map a data source from other networks into a WIA-PA device in order to fulfil communication between the WIA-PA network and other networks.

c) Disaggregation object

This object is used to disaggregate the packets that are aggregated within routing devices and field devices.

d) Data image module

The data image module stores the data of devices in the WIA-PA network and provides an access interface for other networks.

6 System management

6.1 General

The system management in the WIA-PA network includes both network management and security management. The functions of system management are implemented by the Device

Management Application Process (DMAP) in each device. The DMAP, as a system management entity, includes the network management module, the security management module, and the Management Information Base (MIB) module. DMAP is shown as the grey area in Figure 5. The white blocks within the grey area are the function components in the DMAP.

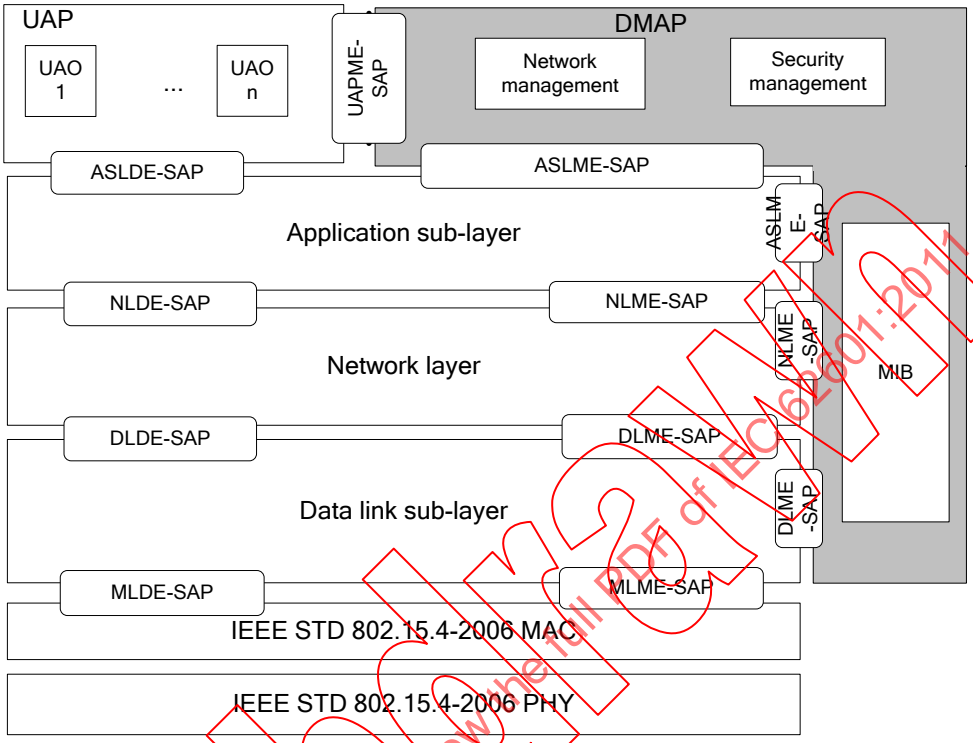

Figure 5 – DMAP in system management

The network management specifies the management of device attributes and attributes that are related to communication and network configuration. The network management functions are accomplished by the DMAPs in the NM, cluster heads, and cluster members. The network management functions include the following actions:

a) Joining and leaving the network

A routing device should join or leave the network through the gateway device or an online routing device. A field device should join or leave the network through an online routing device or the gateway device. The new joining device should be authenticated by the SM.

b) Network address allocation

Each device in the WIA-PA network has a 64-bit Extended Unique Identifier (EUI-64) address and a 16-bit network address. The EUI-64 address is also called long address. The long address of each device is assigned by vendors according to the IEEE EUI-64 address. The 16-bit network address is also called short address. The short address of each routing device is assigned by the NM. The short address of each field device is assigned by the routing device.

c) Routing configuration

Static routing is implemented in the WIA-PA network. The routing table in each routing device is configured by the NM.

d) Communication resource configuration

Communication resources of the WIA-PA network are organized as superframes (see 8.4.4). When a routing device has successfully joined the network, it should be allocated

communication resources by the NM. After a field device has successfully joined a cluster, the cluster head should allocate communication resources to it.

e) Time source configuration and services of system time

The WIA-PA network sets one base time source, which is performed by the gateway device. In order to recognize the order of occurring events, the WIA-PA devices relatively synchronize with the gateway device.

f) Performance monitoring

Performance monitoring is necessary to collect the performance of the WIA-PA network, which includes device status, path failure, and channel condition.

g) MIB maintenance

The MIB module should configure the MIB of the WIA-PA protocol stack.

h) Interfaces for plant operators and maintenance personnel

The NM should provide an interface to allow plant operators and maintenance personnel to monitor and control the performance/activities of the network and devices. The definition of this interface is out of the scope of the current WIA-PA standard.

i) Firmware upgrading

Firmware upgrading should update the protocol stack of WIA-PA devices. Because on-line firmware upgrading is a power-hungry operation, WIA-PA does not recommend this mode and does not specify it in this document. WIA-PA recommends off-line firmware upgrading through handheld devices. Handheld devices connect devices that are to be upgraded by wire or wireless connections. Wired connections may use the serial communication methods. Wireless connections are not specified in this document.

The security management should manage the attributes associated with network security. The security management is performed by the SM and by the security management modules in cluster heads and cluster members. The SM handles the centralized authentication of the whole network, and needs coordination with the NM. See 10.1 for the functions of security management.

The MIB module stores all attributes used in the WIA-PA network, including both structured attributes and unstructured attributes (see 6.9).

6.2 Framework of system management

The WIA-PA network supports the hybrid centralized and distributed management scheme.

The hybrid centralized and distributed management framework is illustrated in Figure 6. The system management is implemented by the NM, the SM, and cluster heads. Cluster heads are directly managed by the NM and the SM. In addition, they are provisioned with the privilege of managing cluster members.

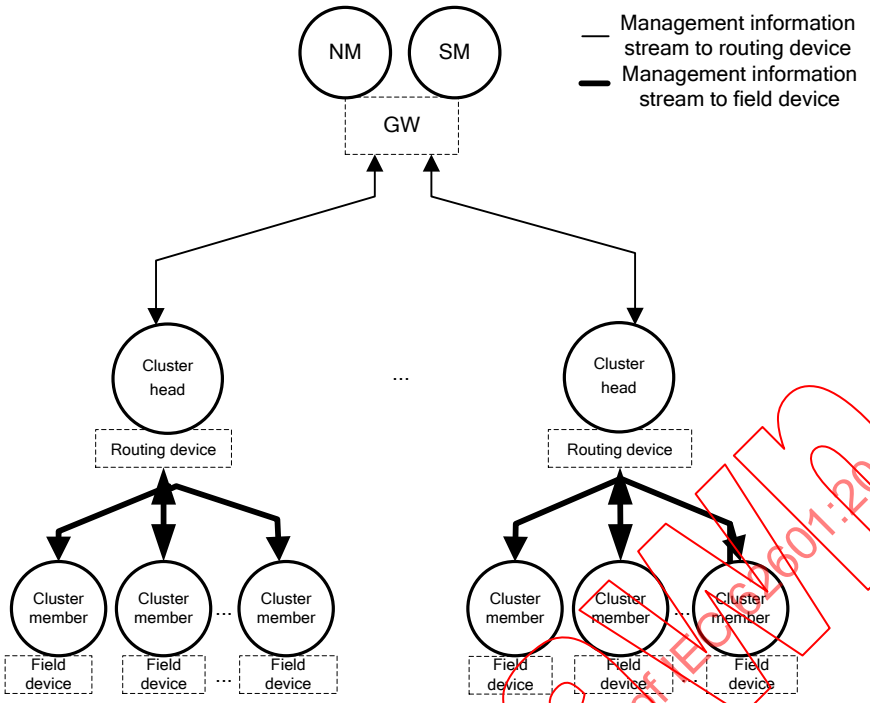

Figure 6 – Hybrid centralized and distributed system management scheme

The NM should perform the following tasks:

a) Constructing and maintaining the mesh topology comprised of cluster heads, and star topology comprised of cluster heads and cluster members;

b) Allocating communication resources for cluster heads in mesh topology, and allocating communication resources for cluster members to the cluster heads in star topology;

c) Monitoring the performance of the WIA-PA network, including device status, path failure, and channel condition.

The SM should perform the following tasks:

a) Authorizing the cluster heads and cluster members that are attempting to join the WIA-PA network;

b) Managing keys in the entire network, including key generation, key distribution, key recovery, and key revocation;

c) Authorizing the relationship of the end-to-end communication.

The cluster head should perform the following tasks:

a) Constructing and maintaining the star topology; allocating communication resources that are allocated to the star topology by the NM to cluster members in the cluster; providing the monitoring results of the star topology to the NM;

b) Storing and forwarding keys used in the star topology; authorizing the communication relationship among cluster heads; authorizing the communication relationship between a cluster head and cluster members.

6.3 Joining process

6.3.1 Joining process of routing device

When a routing device intends to join the network, it should be provisioned a Join Key by a handheld device. See clause 11 for the detailed provisioning process.

The joining process of a routing device includes the following actions:

a) A routing device scans the available channels to get beacons from either the online routing devices or the gateway device.

b) The routing device chooses an online routing device or the gateway device as the temporal parent and synchronizes with the network according to the received beacon.

c) The routing device sends a joining request to its temporal parent, which will then forward the request to the NM.

d) When receiving a joining request, the NM should communicate with the SM to complete the authentication process. Then the NM returns a joining response.

e) The routing device receives the joining response relayed by its temporal parent. If the response is negative, the routing device should restart this joining process; otherwise, the joining process should be finished.

The joining process of a routing device through the gateway device is shown in Figure 7.

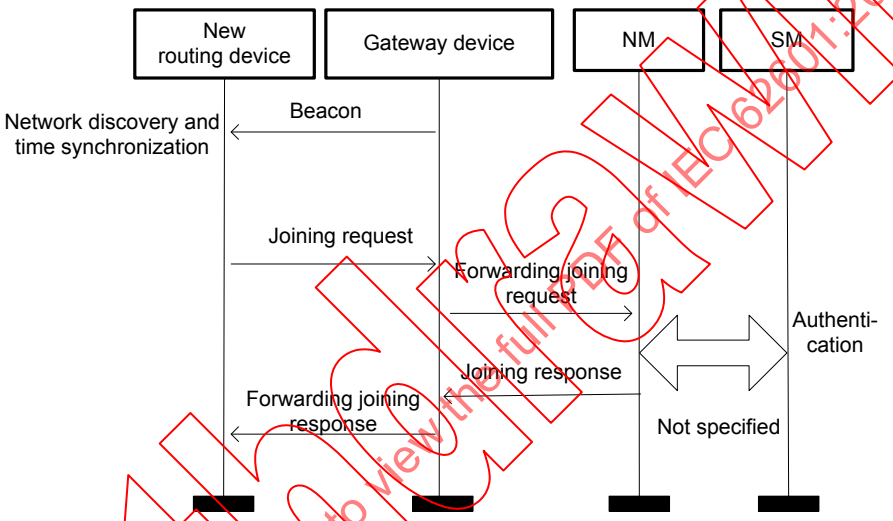

Figure 7 – Joining process of routing device through the gateway device

The joining process of a routing device through an online routing device is shown in Figure 8.

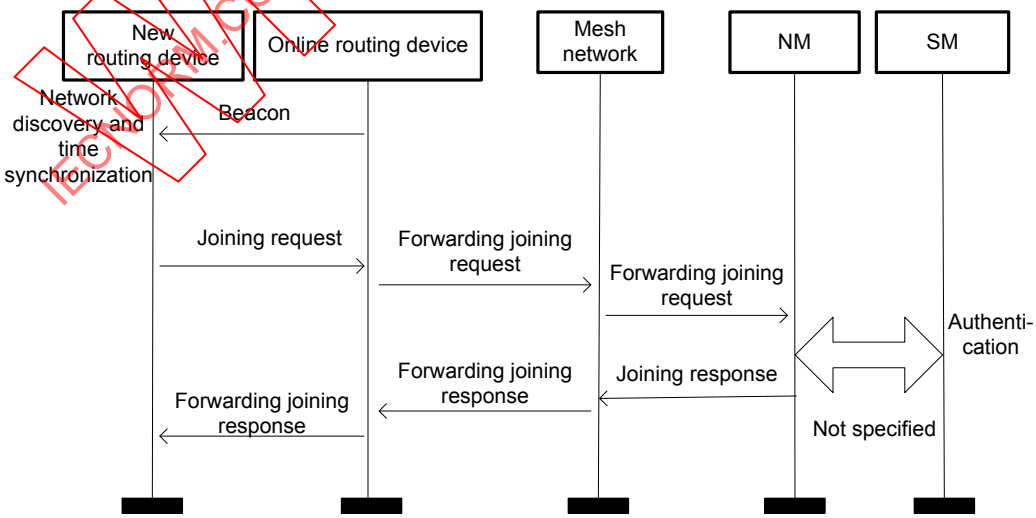

Figure 8 – Joining process of routing device through an online routing device

See 9.5.8 for the allocation of communication resources.

6.3.2 Joining process of field device

When a field device intends to join the network, it should be provisioned a Join Key by a handheld device. See Clause 11 for the detailed provisioning process.

The joining process of a field device includes the following actions:

a) A field device scans the available channels to get beacons from the online routing devices or the gateway device;

b) The field device chooses one online routing device or the gateway device as the cluster head and synchronizes with the network according to the received beacon;

c) The field device sends the joining request to the cluster head;

d) When receiving the joining request, the cluster head forwards it to the NM;

e) After receiving the joining response from the NM, the cluster head returns the joining response to the field device according to the type of network topology and the available communication resources of the cluster head. If the network topology is mismatched, the status in the response is set to FAILURE_TOP_DISMATCH; if there are other types of failures, the status in the response is set to FAILURE_ELSE; if the joining is successful, the status in the response is set to SUCCESS. See Table 49 for definition of “status”;

f) The field device receives the joining response from the cluster head. If the status in the response is FAILURE_TOP_DISMATCH, the field device will choose one routing device as its cluster head and restart the joining process as above; if status in the response is FAILURE_ELSE, the field device will restart this joining process; otherwise, if status in the response is SUCCESS, the field device has joined in the network.

The joining processes of a field device through the gateway device and through a routing device are shown in Figure 9 and Figure 10 respectively. See 9.5.8 for the allocation of communication resources.

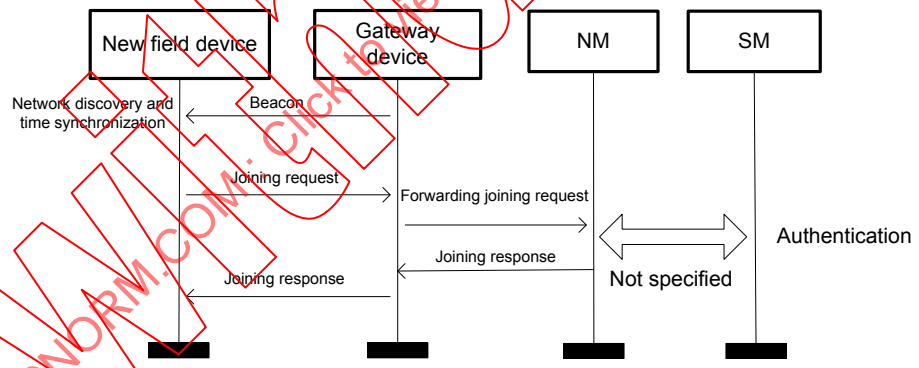

Figure 9 – Joining process of field device through a gateway device

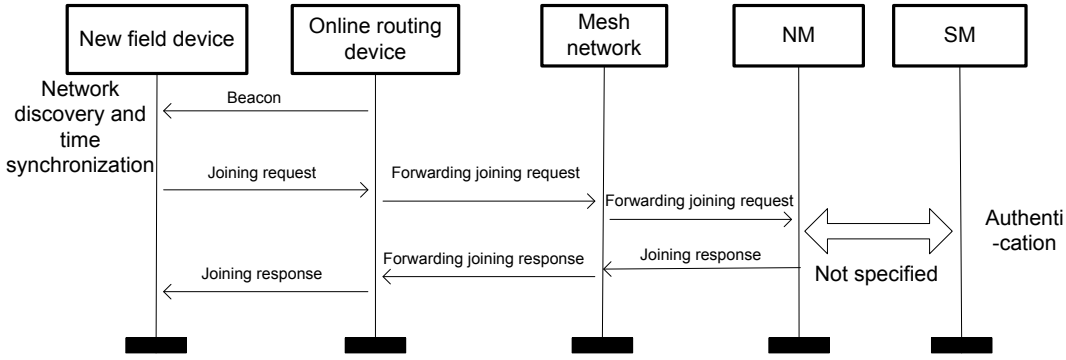

Figure 10 – Joining process of field device through a routing device

NOTE 1 A WIA-PA handheld device connects to the WIA-PA network via the gateway device or a routing device. The joining process of the handheld device is the same as other WIA-PA devices. After joining in the network, the

WIA-PA handheld device configures the WIA-PA devices and collects the network performance/health information during the Contention Access Period (CAP) period of the WIA-PA superframe.

NOTE 2 See clause 11 for detailed information about security.

6.3.3 Addressing and address assignment

In the WIA-PA network, each routing device or field device has a global unique 64–bit "long address" and a 16-bit "short address". The long address, as shown in Figure 11, is set by manufacturers according to the IEEE EUI-64 address.

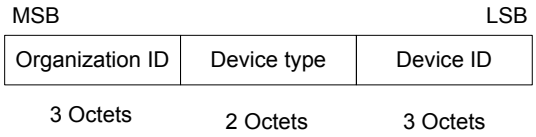

Figure 11 – Long address structure of device

The short address is 16-bit long. The most significant 8 bits of the short address that are assigned by the NM are used to identify different clusters.

The "short address" of a routing device is shown in Figure 12. For the short address of the routing device, the least significant 8 bits are set to 0.

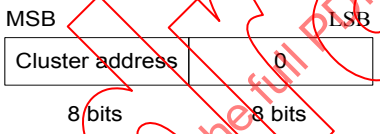

Figure 12 – Short address structure of routing device

The "short address" of a field device is shown in Figure 13. The least significant 8-bit part of the field device's short address is the intra-cluster address. The intra-cluster address is assigned by the cluster head.

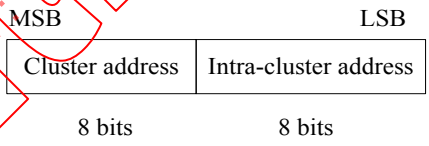

Figure 13 – Short address structure of field device

6.4 Virtual Communication Relationship (VCR)

6.4.1 Definition

Access to the User Application Objects (UAOs) is defined by the VCRs and the endpoints of a VCR are two UAOs. VCRs distinguish the routing and communication resources among different UAOs. Each VCR is identified by a VCR IDentifier (VCR_ID). The attributes of a VCR include UAO ID at source side, UAO ID at destination side, addresses of source device/destination device, VCR type, valid scope, etc (see Table 20).

VCRs are classified into three types according to the application modes:

a) Publisher/Subscriber (P/S) VCR, which is used to publish periodic data;

b) Report source /Sink (R/S) VCR, which is used to support aperiodic events and trend reports;

c) Client/Server (C/S) VCR, which is used to transfer aperiodic and dynamic paired unicast messages.

VCRs are classified into three types according to its aggregation functions:

a) Non-aggregation VCR

If the aggregation is not supported by a cluster member, the non-aggregation VCR is used by the cluster member to send non-aggregation packets to its cluster head. If the cluster head does not support the aggregation function, it will forward the non-aggregation packets to the gateway through the non-aggregation VCR.

b) Data aggregation VCR

If the aggregation is supported by a cluster member, the data aggregation VCR is used by the cluster member to send the aggregated data from the UAOs to its cluster head. If the cluster head does not support the aggregation function, it will forward the aggregated data to the gateway through the data aggregation VCR. The effective interval of the data aggregation VCR originates from the Data Aggregation Object (DAGO) of the cluster head to the DisaGgregation Object (DGO) of the gateway.

c) Packet aggregation VCR

If the aggregation is supported by a cluster head, the packet aggregation VCR is used by the cluster head to send aggregated packets to the gateway. The effective interval of the packet aggregation VCR originates from the Packet Aggregation Object (PAGO) of the cluster head to the DisaGgregation Object (DGO) of the gateway.

NOTE PAGOs, DGOs, and DAGOs are three special UAOs. See 6.6.1 for DAGOs.

6.4.2 Protocol support for VCR

The protocol supports for VCRs are listed in Table 1.

Table 1 – Protocol support for VCR

Protocol layer support			VCR types		
			P/S VCR	R/S VCR	C/S VCR
AL	User applications		Periodic data transmission of UAO	Aperiodic report e.g. alarm or event	Attribute getting and setting operation in UAO
	Communication modes in Application Sub-Layer (ASL)		P/S	R/S	C/S
	Bidirectional		No	No	Yes
	End-to-end retransmission		No	Optional	Yes
	Packet aggregation supporting		Yes	No	No
NL	Redundant path supporting	Intra-cluster	No	No	No
		Inter-cluster	Yes	Yes	Yes
	Unicast/Broadcast		Broadcast/Unicast	Broadcast/Unicast	Unicast
DLSL	Periodic		Yes	No	No
	Real-time		Yes	Optional	No
	Communication resource	Intra-cluster	Contention Free Period (CFP), exclusive timeslots in intra-cluster communication period	CAP	CAP
		Inter-cluster	Exclusive timeslots in inter-cluster communication period	Shared timeslots	Shared timeslots

6.4.3 VCR establishment

The processes of R/S VCR and C/S VCR establishment are as follows:

a) The NM obtains the UAOs of the field devices when they have joined the WIA-PA network. The UAOs are obtained by the NLME-INFO_GET primitives during the CAP (see 9.5.13 and 9.6.4).

b) Once the NM obtains the UAOs, it allocates reserved R/S VCRs and C/S VCRs for each UAO to field devices.

The process of P/S VCR establishment (if it is needed) is as follows:

a) The NM establishes one P/S VCR for each UAO by using the NLME-INFO_SET primitives (see 9.5.14 and 9.6.4). If a field device does not support the data aggregation, each UAO has one non-aggregation P/S VCR, and different VCRs may use the same routing paths. Otherwise, each field device has one data aggregation VCR. If the routing device supports packet aggregation, one packet aggregation VCR is established for the routing device.

See Table 20 for the VCR information.

6.4.4 VCR release

VCRs are released when routing devices and field devices leave the network (see 6.8).

6.5 Routing configuration and communication resource allocation

6.5.1 Routing configuration

Routing devices use static routing that is configured by the NM. The concrete routing algorithms are not specified in this standard. The details of routing paths are shown in Table 14. The redundant paths are supported and use the same RouteID. The routing configuration is realized by Route allocation services (see 9.5.7).

6.5.2 Framework of communication resource allocation

Communication resources consist of timeslots and channels. Allocation of communication resources should be considered in two dimensions: time and channel.

The process of communication resource allocation is as follows:

a) In the mesh network, the communication resourcess of the cluster heads are allocated by the NM. The communication resources consist of those used by cluster heads in the mesh network and those allocated by cluster heads to cluster members.

b) In the star network, the communication resources are allocated by cluster heads to cluster members. That is, communication resources are bound to cluster members.

An example of the resource allocation in the hierarchical network topology that combines star and mesh is shown in Figure 14. First, the NM residing in the GW allocates communication resources to the routing devices R1, R2 and R3. These communication resources are used for communication among R1, R2 and R3, for communication between routing devices and the GW, and for communication between intra-cluster field devices and routing devices. Second, after receiving communication resources from the NM, routing devices allocate some of the communication resources to their intra-cluster field devices, which are used for intra-cluster communications among field devices and their corresponding routing devices. As shown in Figure 14, after the NM allocates communication resources for R1, R2 and R3, R1 allocates communication resources for F1 and F2; R2 allocates communication resources for F5; and R3 allocates communication resources for F3 and F4.

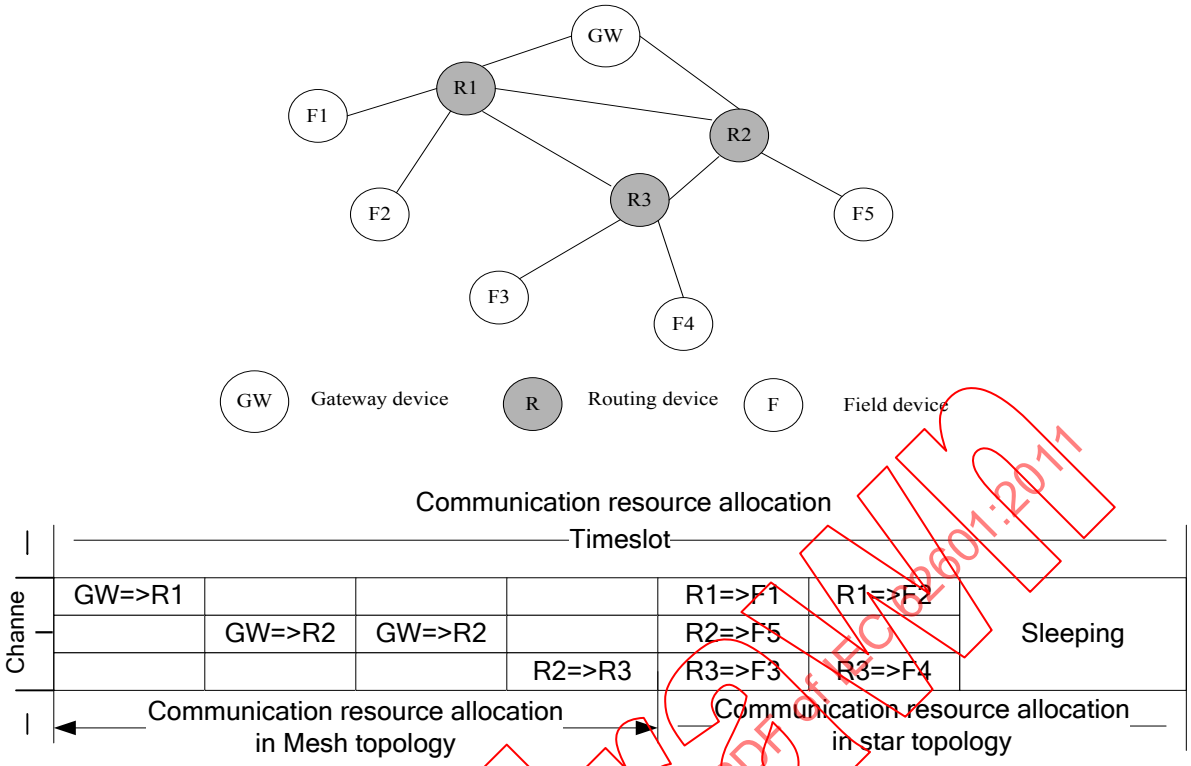

Figure 14 – An example of resource allocation

6.5.3 DLPDU priority and scheduling rules

Four priority levels of DLPDUs are defined:

a) Command (highest priority)

Any packet containing a payload with network-related diagnostics, configuration, control information, or emergent alarms should be classified with a priority of “Command”.

b) Process data (secondary priority)

Any packet containing process data should be classified as secondary priority level, “Process Data”.

c) Normal (third priority)

DLPDUs that do not meet the criteria of “Command”, “Process data”, or “Alarm” should be classified as “Normal” priority.

d) Alarm (lowest priority)

Packets containing only non-emergent alarm and event payload should assume a priority of “Alarm”. Devices should buffer no more than one DLPDU having “Alarm” priority.

The following scheduling rules are employed for allocating communication resources:

a) First allocating channels to the beacon frame and active period;

b) First allocating timeslots to the devices with the fastest update rate;

c) First allocating resources to the packet with the earliest generating time in multi-hop situations;

d) First allocating resources to the highest priority packet prior to other packets.

6.5.4 Communication resource allocation to routing device

The WIA-PA network applies the integrated management strategy of centralized and distributed schemes. After the routing device joins the network, it should scan its neighbor devices on each channel and report the neighbor information to the NM. After receiving the neighbor information, the NM should allocate paths, a superframe, and a block of links to the routing device by using the communication resource allocation services (see 9.5.8). The information of every routing device superframe is broadcast to its field devices in the beacon frame. The allocation process of routing device communication resources is illustrated in Figure 15.

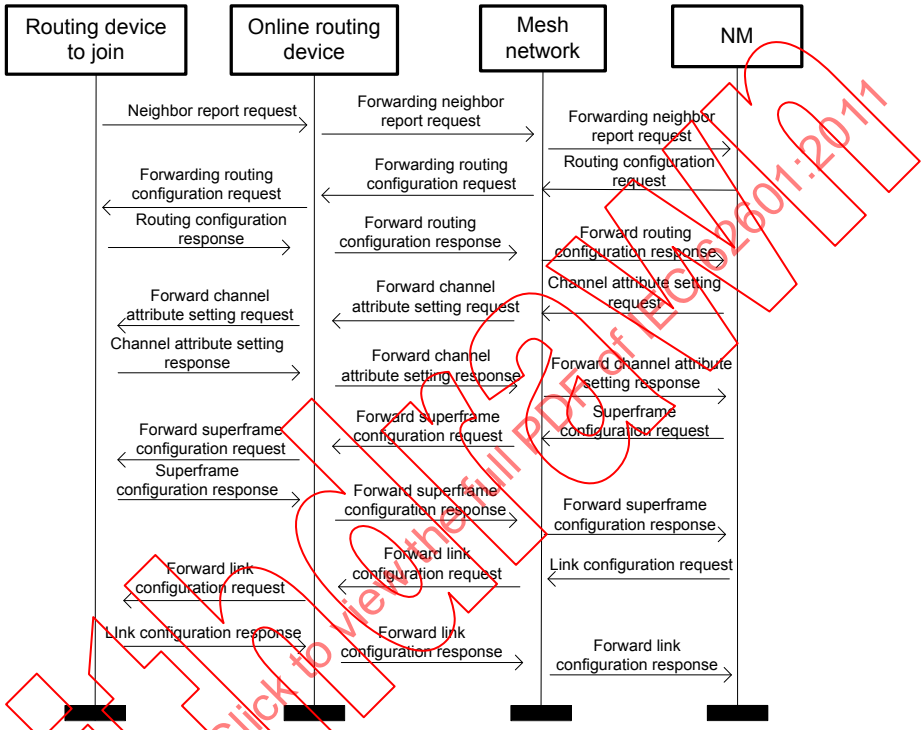

Figure 15 – Allocation process of routing device's communication resources

6.5.5 Communication resource allocation to field device

After VCRs of a field device are established successfully, either the routing device or the gateway device allocates timeslots to its field devices by using the communication resource allocation services (see 9.5.8), which are used in communication between the routing device and the field devices or between the gateway device and the field devices. If the superframes of the routing device and the gateway device are affected by the joining field devices, the routing device and the gateway device should update their RouteTable, Superframe and Link attributes (see 6.9.1.2.2 and 9.5.8).

When a field device joins the network through an online routing device, the communication resource allocation process of the field device is illustrated in Figure 16.

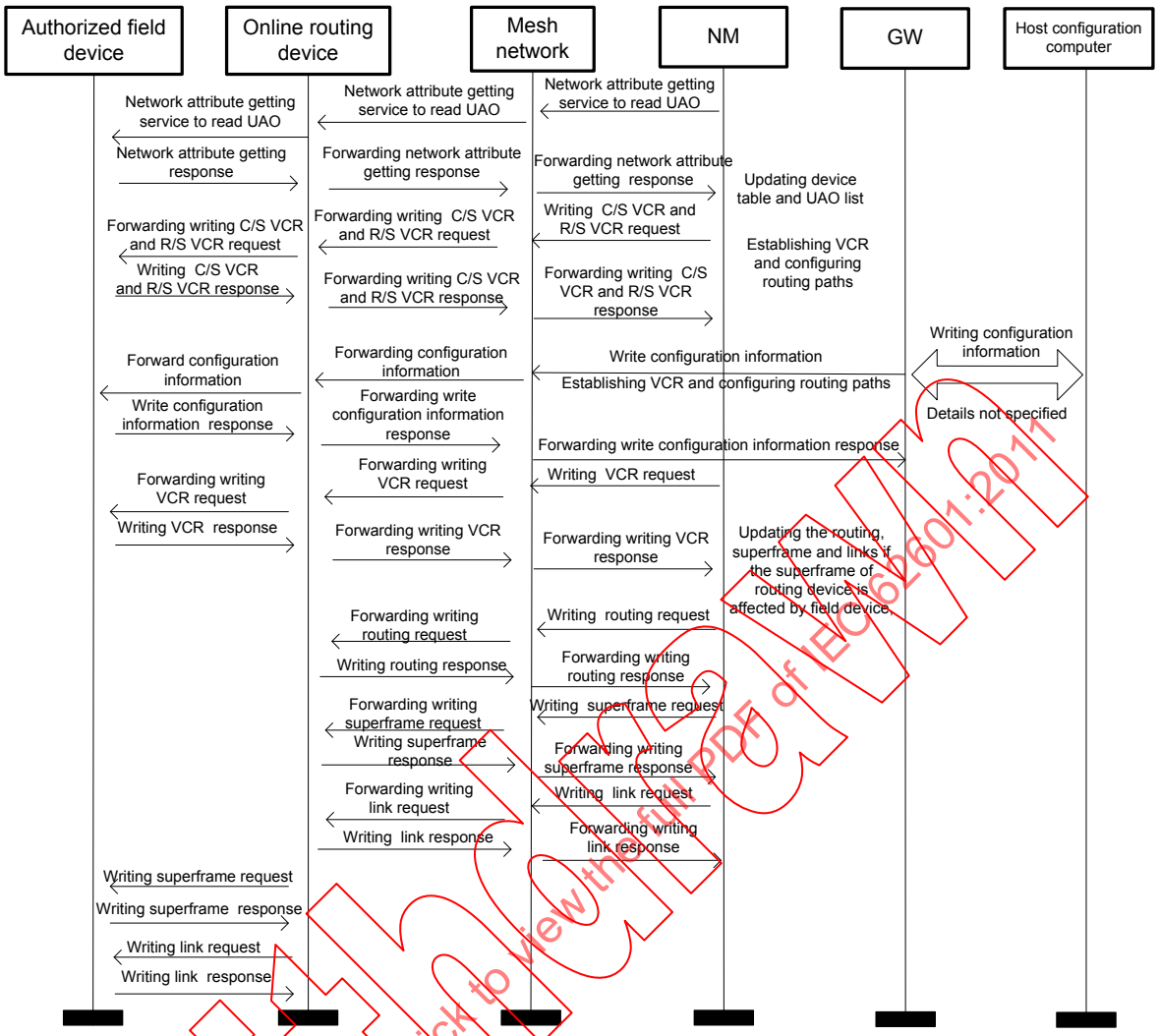

Figure 16 – Allocation process of field device's communication resources

6.6 Aggregation and disaggregation

6.6.1 Aggregation

The aggregation function is optional for field devices and routing devices and is indicated by the aggregation and disaggregation support flag *AGGSupportFlag* (see Table 19). If the field devices or the routing devices support the aggregation mechanism, the NM sets the aggregation and disaggregation enable flag *AGGEnableFlag* to 1 (see Table 19).

The WIA-PA network supports a two-level aggregation mechanism in order to reduce the number of the forwarding packets.

a) Data aggregation. If a field device has more than one User Application Object (UAO), it should decide whether or not to invoke the data aggregation mechanism according to the AGGEnableFlag. This mechanism should reduce the data communication frequency and enhance network efficiency.

b) Packet aggregation. If a routing device receives packets from more than one field device, it should decide whether invoking packet aggregation mechanism or not according to the AGGEnableFlag. This mechanism should reduce the number of packets from the routing device to the gateway device and enhance network efficiency.

The aggregation function is implemented by the DAGOs in field devices and PAGOs in routing devices. The operation parameters of DAGOs and PAGOs are configured by the NM.

The aggregation function supported by the WIA-PA network includes the following four situations:

a) A field device supports the aggregation function, while its routing device does not.

b) A routing device supports the aggregation function, while its field device does not.

c) Both, a field device and its routing device support the aggregation function.

d) Neither a field device nor its routing device supports the aggregation function.

Relations between the VCRs and the aggregation functions supported by field device, routing device and gateway device are shown in Table 2.

Table 2 – Relations between VCR and aggregation function

	Field device DAGO (Aggregation function)		**Routing device PAGO (Aggregation function)**		**Gateway device DGO (Disaggregation function)**	
	Disenable	Enable	Disenable	Enable	Disenable	Enable
Non-aggregation VCR	X		X		–	–
Data aggregation VCR		X	X[a]			X
		X		X[b]		X
Packet aggregation VCR	–[c]	–[c]		X		X

X indicates the aggregation/disaggregation function in the corresponding device is enabled or disabled.

– indicates the VCR is independent of the aggregation/disaggregation function.

[a] The data aggregation VCR starts with DAGO of a filed device and ends at DGO of the gateway device.

[b] The data aggregation VCR starts with DAGO of a field device and ends at PAGO of a routing device.

[c] Field devices have no packet aggregation function.

The aggregation configuration process is listed as follows:

a) NM should read the AGGSupportFlag to verify whether devices support the aggregation function.

b) NM should set the value of the AGGEnableFlag attribute to 1 in the MIB if the routing devices support the aggregation function. The NM should allocate packet aggregation VCRs to the routing devices. The aggregation duration of a routing device should be indicated by attribute AggPeriod (see 6.9.1.2.1) and set to the minimum Dataupdaterate attribute (see Table 20) of its field devices in the cluster. When aggregation duration expires, the routing device aggregates all its packets and sends the aggregated packet out.

c) NM should set the value of the AGGEnableFlag attribute to 1 in the MIB if the field devices support the aggregation function. The NM should allocate the data aggregation VCRs to the field devices. The aggregation duration of a field device should be indicated by attribute AggPeriod (see 6.9.1.2.1) and set to the minimum Dataupdaterate attribute (see Table 20) of its UAOs. When the aggregation duration expires, the field device aggregates all its packets and sends the aggregated packet to the routing device.

According to whether the aggregation is embedded or not, a field device or a routing device operates as follows:

a) If the AGGEnableFlag attribute of a field device is set to 0, this field device does not support the aggregation function. Field devices that have no aggregation function send their data to the routing device through the non-aggregation VCR and the related RouteID (see Table 20). If the aggregation flag AGGEnableFlag attribute of the routing device is also set to 0, the routing device should forward this aggregated packet to the DGO of the gateway device through the original RouteID.

b) If a field device has more than one UAO and its AGGEnableFlag attribute is set to 1, the DAGO of the field device calculates the length of the P/S data from the UAOs and aggregates these data. The format of the aggregated data is shown in Table 3. The aggregated data is sent to the ASL by the DAGO and is encapsulated with the ASL header. The field of the packet type in the ASL header is set to 0b11, which is used to indicate the aggregated packet (see 10.4.2). The aggregation packet in the ASL is sent to the routing device through the data aggregation VCR and the related RouteID; if the AGGEnableFlag attribute of the routing device is 0, the routing device should forward this aggregated packet to the DGO of the gateway device through the original RouteID.

c) If the aggregation function embedded in a routing device is enabled (AGGEnableFlag = 1) and receives packets from its field devices, it reads the NL packet headers. The routing device should aggregate the packet according to the P/S flags and the Flagment flat in the NL packet headers (see 9.6.3). The aggregation rules are as follows:

– If the received packet is P/S type and the Flagment flag is 0, the packet should be aggregated.

– Other types should not be aggregated.

If the packet can be aggregated, its source address and NL payload are sent to the PAGO of the DMAP. The PAGO uses the time when the first packet comes (required to be aggregated) as the start time of a superframe cycle. Once *AggPeriod* (see 6.9.1.2.1) expires, the PAGO aggregates the packets according to the format that is shown in Table 4, and searches the *RouteID* related to the packet aggregation VCR. The aggregated packet, packet aggregation VCR, and the related *RouteID* are then sent to the NL; the NL should send this aggregated packet to the gateway device through the related *RouteID*. The format of the aggregated packet in NL is shown in 9.6.3. If the length of the aggregated packet is longer than the maximum length allowed by the NL, the NL should fragment the packet and send the packet to the gateway device by using the packet aggregation VCR and the related *RouteID*.

Table 3 – Format of aggregated data followed by field device's DAGO

Length in octet(s)	1	1	Variable length	…	1	Variable length
Field name	Number of aggregated data	Data length	Data	…	Data length	Data

Table 4 – Format of aggregated packet followed by routing device's PAGO

Length in octet(s)	1	2	1	Variable length	…	2	1	Variable length
Field name	Number of aggregated packet	Source address	Data length	Data	…	Source address	Data length	Data

6.6.2 Disaggregation

The disaggregation function is optional for the gateway device and is indicated by the aggregation and disaggregation support flag *AGGSupportFlag* (see Table 19). If the gateway device supports the disaggregation mechanism, the NM sets the aggregation and disaggregation enable flag *AGGEnableFlag* to 1 (see Table 19).

The disaggregation function is implemented by the DGO in the gateway device. The gateway device decides whether to disaggregate the received packet according to the NL header and the ASL header. If the value of packet type in the NL header is 1, the gateway device disaggregates the NL aggregated packet; if the value of the packet type in the ASL header is ob11, the gateway device continues disaggregating the ASL aggregated packet.

The operation parameters of the DGO are configured by the NM.

The process of disaggregation is as follows:

a) After the gateway device receives the aggregated packets from field devices and routing devices, it will notify the DGO in its DMAP to disaggregate the packets.

b) The DMAP sends the disaggregated packets to UAOs.

6.6.3 An example of the two level aggregation process

The following example illustrates the aggregation and disaggregation processes. The example is based on a network that consists of two field devices, one routing device, the GW, and the host configuration computer. As shown in Figure 17, the routing device R acts as the cluster head of field device A and field device B; there are two UAOs named a1 and a2 residing in field device A and two UAOs named b1 and b2 residing in field device B.

This example supposes that field device A does not support the aggregation function, while field device B and routing device R do. The data update rates of a1 and a2 are configured to 1s, and the data update rates of b1 and b2 are configured to 4s. The DAGO of the field device B is responsible for aggregating the packets from b1 and b2, and generates packet p_b. The routing device R has one PAGO and the aggregation duration is the minimum data update rates of a1, a2, b1, and b2, whose value is 1s. The routing device R has a PAGO, which aggregates the packets from a1, a2 and packet p_b. The gateway device has a DGO, which disaggregates packets from field device A and B.

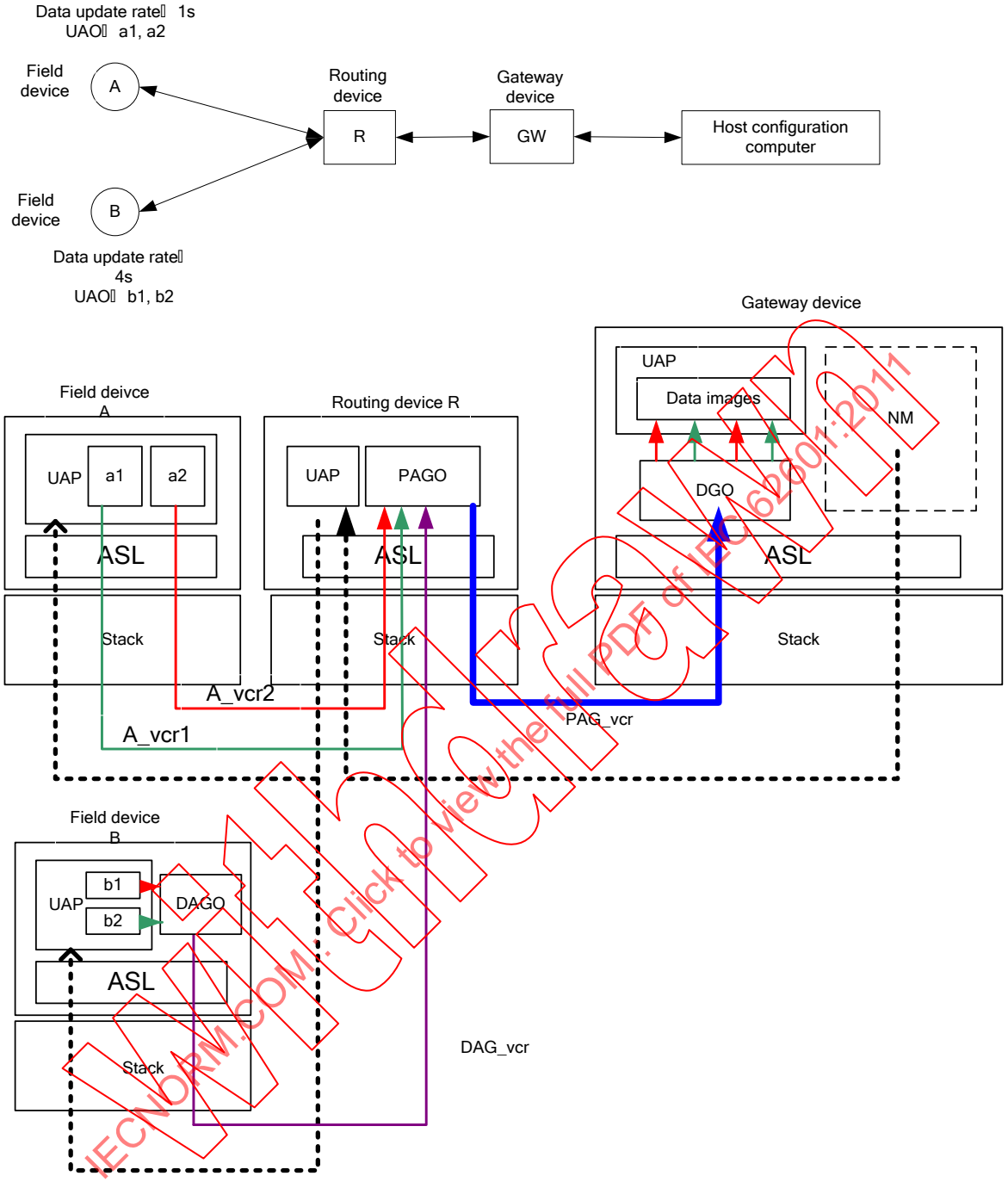

Figure 17 – Example of aggregation and disaggregation

The aggregation and disaggregation process is listed below:

a) After the network is established, the NM allocates the non-aggregation VCRs to a1, a2, b1, and b2 for the transmissions of non-aggregation packets. The non-aggregation VCRs of a1 and a2 in field device A are indicated by A_vcr1 and A_vcr2. The endpoints of A_vcr1 are a1 and the UAP of the GW, and the endpoints of A_vcr2 are a2 and the UAP of the GW. Because the data transmissions of field device B do not use the non-aggregation VCRs, the non-aggregation VCRs of b1 and b2 in field device B are not indicated in this example.

b) After constructing the non-aggregation VCRs, the NM configures the aggregation duration and data aggregation VCR for field device B and configures the aggregation duration and packet aggregation VCR for routing device R. The aggregation durations are the minimum

data update rate among all UAOs in a field device or the minimum data update rate among all field devices for a routing device. Because the data update rates of b1 and b2 are 4s, the aggregation duration of DAGO in field device B is four seconds; the data aggregation VCR of field device B is indicated by DAG_vcr and its endpoints are DAGO in field device B and the DGO in the GW. The aggregation duration of routing device R is set to the minimum data update rate in a cluster. Because the minimum data update rate among a1, a2, b1, and b2 is one second, the aggregation duration of routing device R is set to 1s. The packet aggregation VCR of routing device R is indicated by PAG_vcr and its endpoints are PAGO in the routing device R and DGO in the GW.

c) After A_vcr1, A_vcr2, DAG_vcr, and PAG_vcr are established, the DAGO, PAGO and DGO begin to work. The DAGO in field device B uses the time when the first packet comes (required to be aggregated) as the start time, and aggregates the packets and sends them to routing device R through DAG_vcr at the configured aggregation timeslot. The PAGO in routing device R uses the time when the first packet comes (required to be aggregated) as the start time, and aggregates the packets in DMAP and sends them to the DGO through PAG_vcr in the GW at the configured aggregation timeslot.

d) After the DGO of the GW receives the aggregated packets, it disaggregates the packets and finds the corresponding UAOs according to the packet source address and UAO identifiers.

e) The DMAP in GW sends the disaggregated packets to the corresponding UAOs.

6.6.4 Management of aggregation and disaggregation objects

The attributes of the DAGO class are shown in Table 5.

Table 5 – DAGO class attributes

Attribute ID	Name	Data type	Default value	Usage	Description	Supporting method
0	DagoID	Unsigned8	0	Mandatory	The identifier of the DAGO	READ
1	DagoRevision	Unsigned8	0	Optional	The version defined by the DAGO	READ

The attributes of the DAGO instance are shown in Table 6.

Table 6 – DAGO instance attributes

Attribute ID	Name	Data type	Default value	Usage	Description	Supporting method
0	DagoInsID	Unsigned8	0	Mandatory	The identifier of the DAGO instance	READ
1	DagoPeriod	Unsigned8	1	Mandatory	The aggregation cycle (in seconds)	READ/WRITE
2	DagoMemNum	Unsigned8	1	Mandatory	Count of the aggregated UAOs	READ/WRITE
3	DagoMemLst	MEM_STRUCT structure (see Table 7)		Optional	List of the aggregated UAOs	READ/WRITE
4	DagoMemData Size	Unsigned8	0	Optional	Length of the aggregated data DagoMemData (in octets)	READ/WRITE
5	DagoMemData	Octetstring		Mandatory	Data of the aggregated UAOs	READ/WRITE

The methods supported by the DAGOs are shown in 10.2.3 Table 165.

The definition of the MEM_STRUCT structure is shown in Table 7.

Table 7 – MEM_STRUCT structure

Attribute ID	Name	Data type	Description
0	AppObjID	Unsigned8	The identifier of the UAO
1	AppObjInsID	Unsigned8	The identifier of the UAO instance
2	AppObjAttrID	Unsigned8	The attribute identifier of the UAO instance

The attributes of the PAGO class are shown in Table 8.

Table 8 – PAGO class attributes

Attribute ID	Name	Data type	Default value	Usage	Description	Supporting method
0	PagoID	Unsigned8	1	Mandatory	The identifier of the PAGO	READ
1	PagoRevision	Unsigned8	0	Optional	The version defined by the PAGO	READ

The attributes of the PAGO instance are shown in Table 9.

Table 9 – PAGO instance attributes

Attribute ID	Name	Data type	Default value	Usage	Description	Supporting method
0	PagoInsID	Unsigned8	0	Mandatory	The identifier of the aggregation object instance	READ
1	PagoPeriod	Unsigned8	1	Mandatory	The aggregation cycle (in seconds)	READ/WRITE
2	PagoMemNum	Unsigned8	1	Mandatory	Count of the aggregated cluster members	READ/WRITE
3	PagoMemLst	Octetstring		Optional	List of the aggregated cluster members	READ/WRITE
4	PagoDataSize	Unsigned8	0	Optional	Length of the aggregated packet (in octets)	READ/WRITE
5	PagoMemData	Octetstring		Mandatory	Data of the aggregated cluster members	READ/WRITE

The methods supported by the PAGOs are shown in 10.2.3 Table 165.

The attributes of the DGO class are shown in Table 10.

Table 10 – DGO class attributes

Attribute ID	Name	Data type	Default value	Usage	Description	Supporting method
0	DgoID	Unsigned8	2	Mandatory	The identifier of the DGO	READ
1	DgoRevision	Unsigned8	0	Optional	The revision defined by the DGO	READ

The attributes of the DGO instance are shown in Table 11.

Table 11 – DGO instance attributes

Attribute ID	Name	Data type	Default value	Usage	Description	Supporting method
0	DgoInsID	Unsigned8	0	Mandatory	The identifier of the DGO instance	READ

The methods supported by the DGOs are shown in 10.2.3 Table 165.

6.7 Performance monitoring

6.7.1 Path failure report

The path failure report is used by routing devices to report path failure events to the NM through the redundant paths. The process of path failure report is shown in Figure 18. See 9.5.12 for more details.

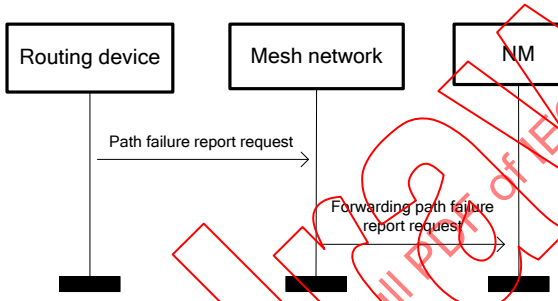

Figure 18 – Process of path failure report

6.7.2 Device status report

After receiving device status reports from intra-cluster field devices, a routing device reports its own health and the health of its field devices to the NM periodically. The NM appraises and diagnoses the network performance according to health information, and replies to the change of network environment timely. The NM should detect abnormal conditions in the WIA-PA device, such as low level of battery power and disconnection from neighboring devices. This is realized by setting alarm levels to the WIA-PA devices. The device status report processes of the field device and the routing device are shown in Figure 19 and Figure 20.

See 9.5.10 for more details.

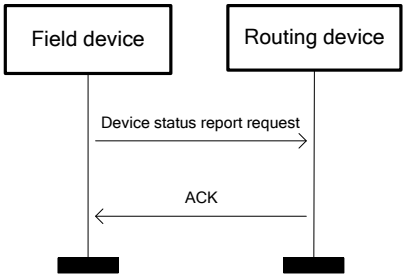

Figure 19 – Device status report process of field device

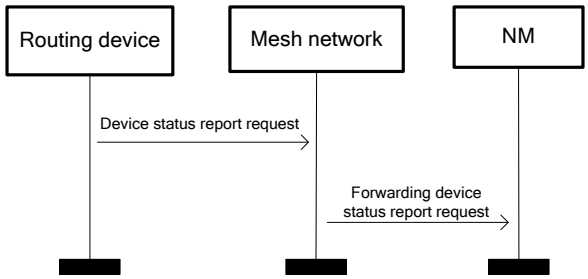

Figure 20 – Device status report process of routing device

6.7.3 Channel condition report

The channel condition report is used for the WIA-PA field devices or routing devices to report the channel condition remotely to the NM. The process of channel condition report is shown in Figure 21. See 9.5.11 for more details.

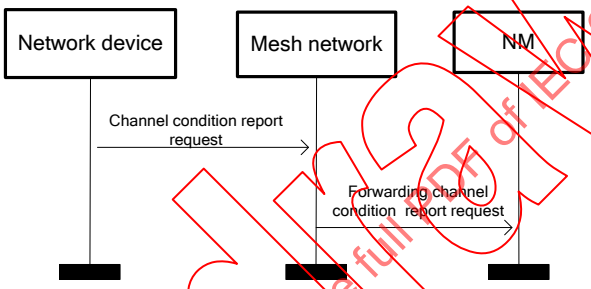

Figure 21 – Process of channel condition report

6.8 Leaving process

6.8.1 General

The leaving process of WIA-PA devices includes the abnormal leaving process, the active leaving process and the passive leaving process. The abnormal leaving process is caused by device failure, device invalidation, and energy depletion, which should be judged by the Keep-alive command frame of DLSL. The active leaving is requested by field devices from their routing devices or by routing devices from the gateway device. The passive leaving is requested by the gateway device from routing devices or by routing devices from their field devices.

6.8.2 Leaving process of routing device

The WIA-PA network specifies two leaving processes for routing devices: active leaving and passive leaving. The active leaving process of a routing device is shown in Figure 22.

a) A routing device sends the leaving request to the NM.

b) The NM gives a leaving response to the routing device.

c) After receiving a positive response, the routing device notifies the leaving to its field devices. After passive leaving of field devices, the routing device leaves the network.

d) The NM releases the network address, VCR and the communication resources of the departing routing device, and updates the network topology.

e) The NM notifies the routing devices that have communication relationships with the departed routing device to release the related communication resources.

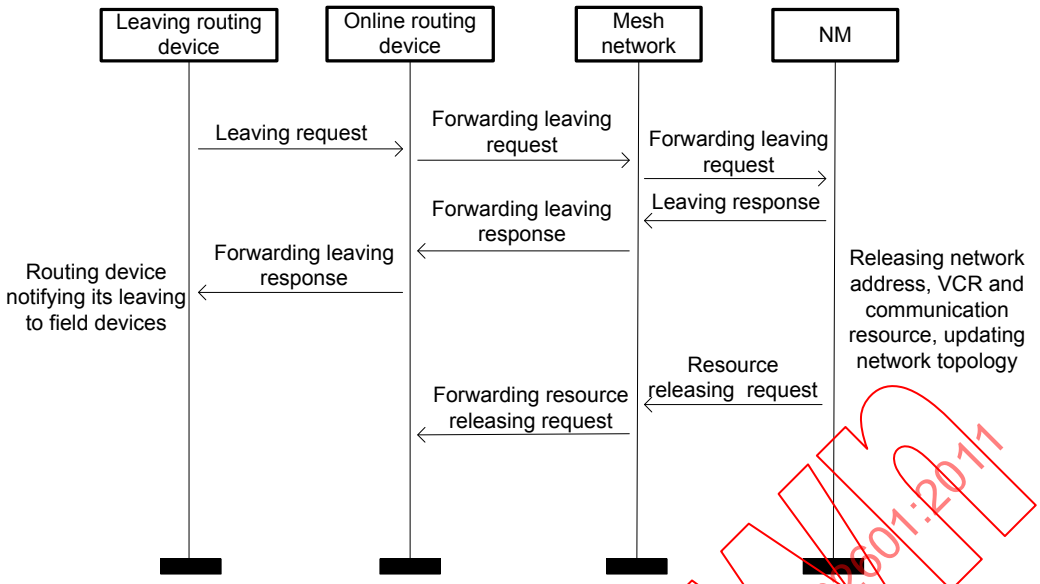

Figure 22 – Active leaving process of routing device

The passive leaving process of a routing device is shown in Figure 23.

a) The NM sends the leaving request to a routing device.

b) The routing device sends a leaving response to the NM after receiving the leaving request.

c) The routing device notifies its leaving to the field devices. After passive leaving of field devices, the routing device leaves the network.

d) The NM releases the network address, VCR and communication resources, and updates the network topology after receiving the leaving response from the routing device.

e) The NM notifies the routing devices that have communication relationships with the departed routing device to release the related communication resources.

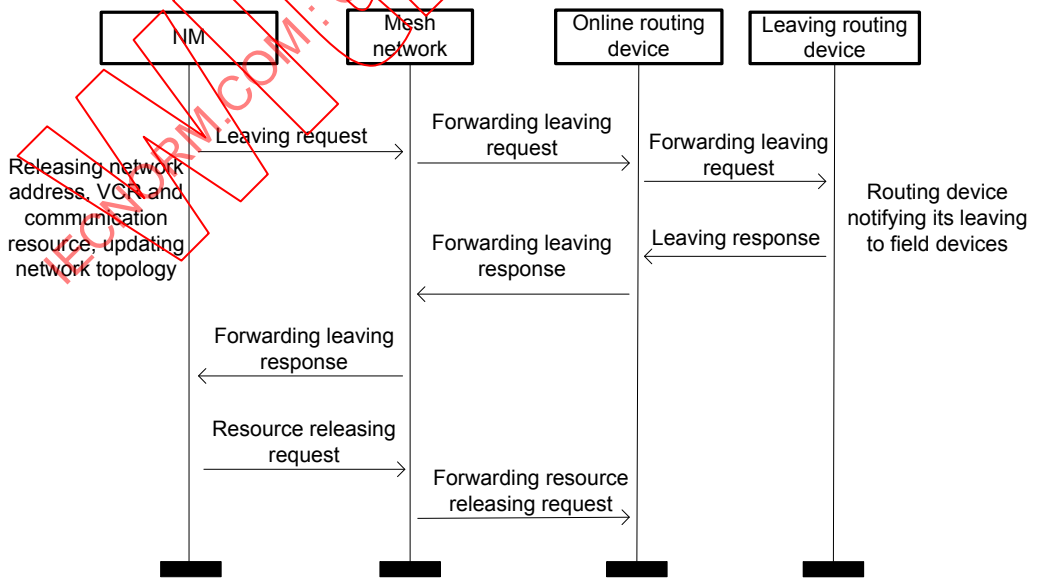

Figure 23 – Passive leaving process of routing device

6.8.3 Leaving process of field device

The WIA-PA network specifies two leaving process for field devices: active leaving and passive leaving.

The active leaving process of a field device is shown in Figure 24 andFigure 25.

a) A field device sends the leaving request to the routing device or the gateway device in a cluster.

b) The routing device or the gateway device returns a leaving response to the leaving field device.

c) The routing device or the gateway device releases the intra-cluster network address, VCR, and communication resources and updates the cluster member list and UAO list of the field device that has left the network.

d) After receiving the response, the field device leaves the network.

e) If the field device leaves the routing device, the routing device reports the updated cluster member list and UAO list to the NM.

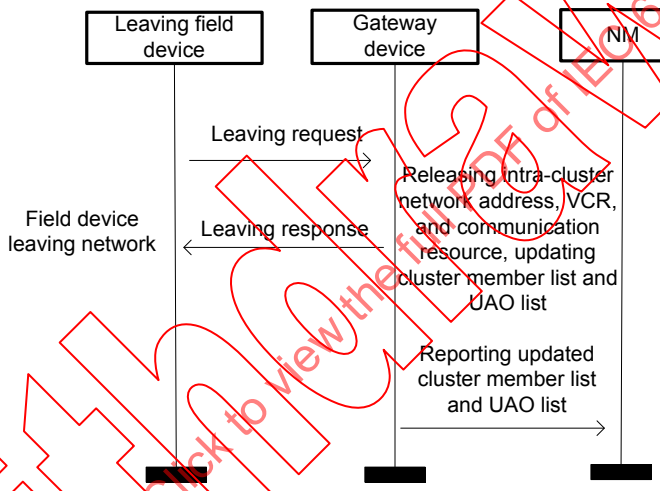

Figure 24 – Active leaving process of field device (leaving from gateway device)

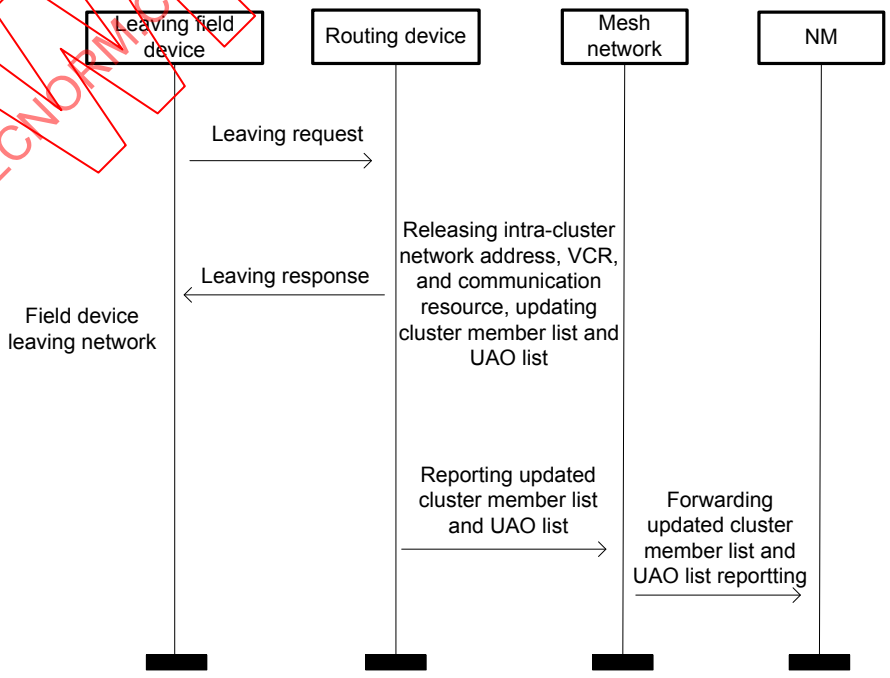

Figure 25 – Active leaving process of field device (leaving from routing device)

The passive leaving process of a field device is shown in Figure 26 and Figure 27.

a) The routing device or the gateway device sends the leaving request to a field device.

b) The field device returns the leaving response to the routing device or the gateway device.

c) The field device leaves the cluster.

d) The routing device or the gateway device releases the intra-cluster network address, VCR and communication resources of the departing field device and updates the cluster member list and UAO list.

e) If the field device leaves the routing device, the routing device reports the cluster member and its UAO list to the NM.

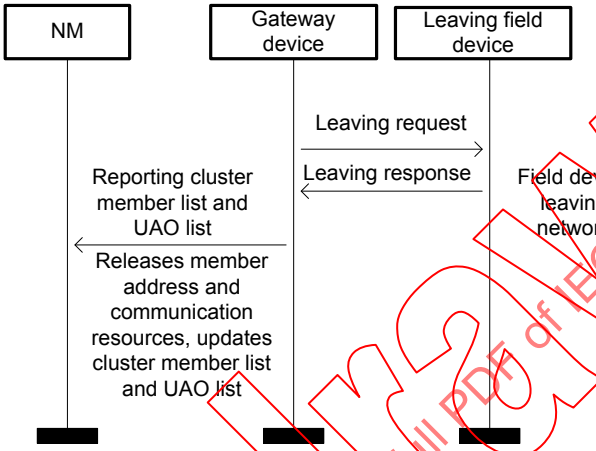

Figure 26 – Passive leaving process of field device (leaving from gateway device)

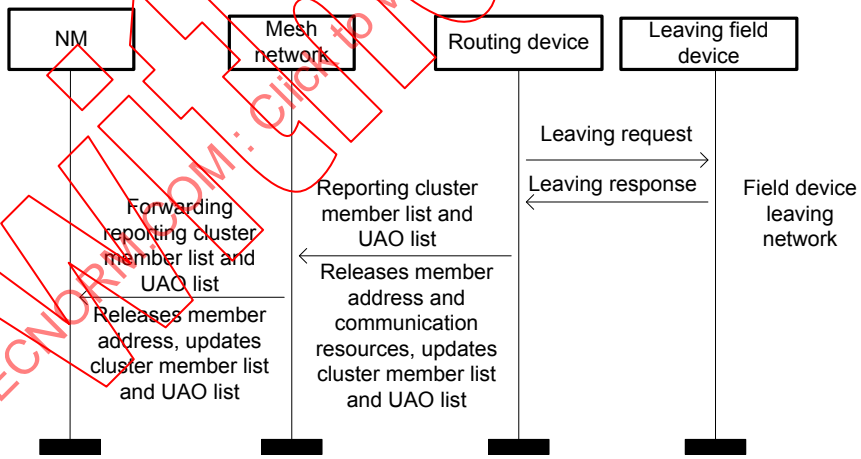

Figure 27 – Passive leaving process of field device (leaving from routing device)

6.9 Management information base and services

6.9.1 Management information base

6.9.1.1 General

Items stored in the MIB are called attributes and are used for monitoring and configuring the WIA-PA network parameters. These attributes can be accessed and updated by the NM.

According to the storage types, the attributes in the MIB are classified into three categories:

a) Constant attribute,

b) Static attribute, and

c) Dynamic attribute.

A constant attribute, such as the serial number of a wireless device, is unchangeable with time. The constant attribute is set when devices leave manufacturers and should not be modified.

A static attribute, such as an alarm limit, changes its value infrequently. The value of the static attribute should be preserved after the warm restart/reset/ power-failure.

A dynamic attribute changes its value frequently without any external command. The value of the dynamic attribute will be lost after the warm restart/reset/power-failure. Warm restart is a sequence of operations that is performed to reset a previously running device after an unintentional shutdown. When a device is reset, it enters into the idle state. Power-faliure indicates that the available power that is used by devices is becoming critically low.

According to the attribute data types, the attributes in the MIB are divided into unstructured attributes and structured attributes.

According to the implementation requirements, the attributes in the MIB are divided into mandatory attributes and optional attributes.

There are two types of the access rights to the the attributes in the MIB:

a) R (Read), which means the values of the attribute may be read by other devices in WIA-PA network; and

b) W (Write), which means the values of the attribute may be set by other devices in WIA-PA network.

6.9.1.2 MIB attributes

6.9.1.2.1 Unstructured attributes

The unstructured attributes are listed in Table 12. The values of attributes numbered 0 to 20 are identical in the entire network; and the values of attributes numbered 21 to 28 are device specific.

Table 12 – Unstructured attributes

ID	Name	Data type	Valid range	Access type	Storage type	Default value	Description
0	NetworkID	Unsigned8	0 to 255	R/W	Static	0	Network identifier, used for multiple networks coexisting
1	StatisticsDuration	Unsigned16	0 to 65 535	R/ W	Static	0	Configuring the cycle of the statistic collection (in seconds). After this duration, WIA-PA devices update the statistic data in the neighbor tables.
2	MaxNSDUSize	Unsigned8	0 to 255	R	Static	N/A	Maximum size of service data unit supported by the NL
3	BitMap[a]	Unsigned32	32 bits	R/W	Dynamic	0	Indicating if a channel can be used: 0 = Not used; 1 = Used. The value of bit *i* indicates the usage status of channel *i*. *i* is related to the IEEE 802.15.4 channel number.
4	KeepAliveDuration	Unsigned24	0 to (2^{24}-1)	R/W	Static	0	Duration between two keep-alive frames (in milliseconds)

Table 12 *(Continued)*

ID	Name	Data type	Valid range	Access type	Storage type	Default value	Description
5	TimeSynDuration	Unsigned24	0 to (224-1)	R/W	Static	0	Duration between two time synchronization frames (in milliseconds)
6	ChannelThreshold	Unsigned8	0 to (*macMaxFrame-Retries*)–*2*	R/W	Static	1	The channel switch threshold in adaptive frequency hopping, indicated as retry count; See IEEE STD 802.15.4, 7.4.2 for *macMaxFrameRetries*
7	SecEnableFlag	Boolean	0, 1	R/W	Static	0	Security enable flag: 0 = the whole network needs authentication; 1 = the whole network does not need authentication.
8	KeyupdataDur[b]	Unsigned32	0 to (2^{32}-1)	R/W	Dynamic	0	The updating cycle of key (in milliseconds)
9	MaxAttackedCnt[b]	Unsigned16	0 to 65 535	R/W	Static	0	The maximum count of attacks
10	EtoEACKTimeOut	Unsigned24	0 to (2^{24}-1)	R/W	Static	1	The upper limit of waiting time for end to end acknowledge (in milliseconds)
11	TimeSlotDuratioin	Unsigned16	0 to 65 535	R/W	Static	1	Indicating the timeslot duration. The timeslot duration is calculated by: (aBaseSlotDuration × $2^{\text{TimeSlotDuratioin}}$). See IEEE STD 802.15.4, 7.4.2 for *aBaseSlotDuration*.
12	PLRThreshold	Unsigned8	0 to 100	R/W	Static	50	The threshold of packet loss rate, which is *PLRThreshold* divided by 100 and is used for adaptive frequency switch. See 8.4.5 for adaptive frequency switch.
13	CmemRptCycle[c]	Unsigned16	0 to 65 535	R/W	Static	0	Configuring the reporting cycle of cluster member information (in seconds)
14	NeiInforRptCycle[c]	Unsigned16	0 to 65 535	R/W	Static	0	Configuring the reporting cycle of neighbor information (in seconds)
15	ChaStaRptCycle	Unsigned16	0 to 65 535	R/W	Static	0	Configuring the reporting cycle of channel condition information (in seconds)
16	DevStaRptCycle	Unsigned16	0 to 65 535	R/W	Static	0	Configuring the reporting cycle of device status information (in seconds)
17	PathFailRptCycle[c]	Unsigned16	0 to 65 535	R/W	Static	0	Configuring the reporting cycle of failure path information (in seconds)
18	MaxEtoERetry	Unsigned8	0 to 255	R/W	Static	1	The maximum number of end-to-end retry
19	NetworkTopology	Boolean	0, 1	R/W	Static	1	Indicating the network topology: 0 = hierarchical network that is the combination of star and mesh; 1 = star-only network topology.

Table 12 *(Continued)*

ID	Name	Data type	Valid range	Access type	Storage type	Default value	Description
20	UTCTime	Unsigned32	0 to (2^{32}-1)	R/W	Dynamic	0	Universal Time Coordinated (UTC). The octets from 0 to 2 are used to indicate the data, and the octet 3 is used to indicate the time.
21	SecMode[b]	Unsigned8	0 to 3	R/W	Static	0	Bit 0 represents whether the DLSL uses security service: 0 = Not used; 1 = Used. Bit 1 represents whether the AL uses security service: 0 = Not used; 1 = Used.
22	SecLevel[b]	Unsigned8	0 to 63	R/W	Static	0	Bit 0, 1 and 2 represent the security levels of the DLSL frames: 000 = None; 001 = MIC-32; 010 = MIC-64; 011 = MIC-128; 100 = Encryption; 101 = Encription-MIC-32; 110 = Encription-MIC-64; 111 = Encription-MIC-128; Bit 3, 4 and 5 represent the security levels of the AL packets. 000 = None; 001 = Encryption; 010 = MIC-32; 011 = Encription-MIC-32; Others are reserved.
23	AuthenState[b]	Boolean	0, 1	R/W	Dynamic	0	A flag that marks if a WIA-PA device has been authenticated to be a legal device: 0 = Not authenticated; 1 = Authenticated.
24	AuthenTime[b]	Unsigned32	0 to (2^{32}-1)	R/W	Dynamic	0	Time of a device being authenticated (in UTC time)
25	AttackedCount[b]	Unsigned16	0 to 65 535	R/W	Dynamic	0	The count of attacks, see Annex A for the detailed attacks.
26	AggPeriod[b]	Unsigned8	0 to 255	R/W	Static	1	The aggregation duration (in seconds)
27	ObjectNumber	Unsigned8	0 to 255	R/W	Static	N/A	The number of UAOs.

[a] Attributes are optional for the field device/handheld device.

[b] Attributes are optional for all devices in the WIA-PA networks.

[c] Attributes are not chosen by the field device/handheld device.

6.9.1.2.2 Structured attributes

The structured attributes are listed in Table 13.

Table 13 – Structured attributes

ID	Name	Data type	Access type	Storage type	Description
100	RouteTable	NLRoute_Struct structure (see Table 14)	R/ W	Dynamic	Including the route ID, destination address, next-hop address, VCR_ID, and retry counter Each array member is identified by storage index.
101	Superframe	Superframe_Struct structure (see Table 15)	R/ W	Dynamic	Describing the superframe information Each array member is identified by storage index.
102	Link	Link_Struct structure (see Table 16)	R/W	Dynamic	Describing the link information Each array member is identified by storage index.
103	Neighbor[a]	Neighbor_Struct structure (see Table 17)	R/W	Dynamic	Describing the information of neighbor devices Each array member is identified by storage index.
104	ChannelCondition	ChanCon_Struct structure (see Table 18)	R/ W	Dynamic	Recording the statistic information of channel condition Each array member is identified by storage index.
105	DeviceStruct	Device_Struct structure (see Table 19)	R/W	Dynamic/ constant/ static	Describing the information of WIA-PA devices Each array member is identified by storage index.
106	VCRList	VCR_Struct structure (see Table 20)	R/ W	Dynamic	Recording the VCR information Each array member is identified by storage index.
107	DevConRep	DevConRep_Struct structure (see Table 21)	R/W	Dynamic	Recording the information of device condition Each array member is identified by storage index.
108	KeyTable[b]	Key_Struct structure (see Table 22)	R/W	Dynamic	Including the key type, key length, key update time, key using time, key value, and count of key attack Each array member is identified by storage index.
109	ObjList	ObjList_Struct structure (see Table 23)	R/W	Static	The list of UAO identifiers Each array member is identified by storage index.

[a] Attributes are not chosen by the field device/handheld device.

[b] Attributes are optional for all devices in the WIA-PA networks.

NOTE The sizes of all the arrays listed in Table 13 are implementation dependent.

The data type of routing attributes RouteTable is shown in Table 14.

Table 14 – NLRoute_Struct structure

Attribute member identifier	Name	Data type	Valid range	Description
0	RouteID	Unsigned16	0 to 65 535	A unique routing identifier
1	SourceAddress	Unsigned16	0 to 65 535	The short address of source device
2	DestinationAddress	Unsigned16	0 to 65 535	The short address of destination device
3	NextHop	Unsigned16	0 to 65 535	The short address of next-hop device
4	RetryCounter	Unsigned8	0 to 255	A counter to record end-to-end retries

The data type of superframe attributes Superframe is shown in Table 15.

Table 15 – Superframe_Struct structure

Attribute member identifier	Name	Data type	Valid range	Description
0	SuperframeID	Unsigned16	0 to 65 535	Unique identifier of the superframe, supplied by the NM
1	SuperframeMultiple	Unsigned8	0 to 255	SuperframeMultiple = maximum data update rates / minimum data update rates. It is used for restricting the WIA-PA superframe length and is also used for processing the long cycle data transmission. See 8.4.6 for details.
2	NumberSlots	Unsigned16	0 to 65 535	Superframe size (counts of timeslots)
3	ActiveFlag	Boolean	0, 1	Superframe active flag: 0 = Inactive; 1 = Active.
4	ActiveSlot	Unsigned48	0 to $(2^{48}-1)$	Absolute timeslot number (ASN) when a superframe begins active

The data type of link attributes Link is shown in Table 16.

Table 16 – Link_Struct structure

Attribute member identifier	Name	Data type	Valid range	Description
0	LinkID	Unsigned16	0 to 65 535	Unique identifier of the link
1	NeighborID	Unsigned16	0 to 65 535	Reference to a neighbor table entry, which is empty when the NM allocates links to a routing device for intra-cluster communication.
2	LinkType	Unsigned8	0 to 31	Bit 0 represents the link type: 0 = Unicast; 1 = Broadcast; Bit 1 and bit 2 represent the character of a link: 00 = Transmitting; 01 = Transmit-shared; 10 = Receiving; Bit 3 represents the type of a timeslot: 0 = Data timeslot; 1 = Management timeslot. Bit 4 represents the aggregation character: 0 = Timeslot for non-aggregated packet; 1 = Timeslot for aggregated packet.
3	RelativeSlotNumber	Unsigned16	0 to 65 535	Relative timeslot number
4	LinkSuperframeNum	Unsigned8	0 to 255	LinkSuperframeNum = data update rate of this device/ the minimum data update rate. It is used for processing the long cycle data transmission.
5	ActiveFlag	Boolean	0, 1	Indicating if a link is being used: 0 = Not used; 1 = Being used.
6	ChannelIndex	Unsigned8	0 to 31	The channel sequence numbers for this link, namely, the sequence numbers of the main channels.
7	SuperframeID	Unsigned16	0 to 65 535	Reference to an superframe in the superframe table

The data type of neighbor attributes Neighbor is shown in Table 17.

Table 17 – Neighbor_Struct structure

Attribute member identifier	Name	Data type	Valid range	Description
0	NeighborAddr	Unsigned16	0 to 65 535	The short address of neighbor device
1	NeighborStatus	Unsigned8	0 to 3	Bit 0 represents whether this neighbor device is a main time source: 0 = No; 1 = Yes; Bit 1 represents the status of neighbor device: 0 = Normal; 1 = Abnormal.
2	BackoffCounter	Unsigned8	0 to 31	Backoff counter
3	BackoffExponent	Unsigned8	0 to 15	Backoff exponent
4	LastTimeCommunicated	Unsigned64	0 to (2^{64}-1)	Time when last communicated with the neighbor device
5	AveRSL	Unsigned8	0 to 255	The average level of signals received from the neighbor device in *StatisticsDuration*
6	PacketsTransmitted	Unsigned16	0 to 65 535	The number of un-broadcast frames sends to the neighbor device in *StatisticsDuration*
7	AckPackets	Unsigned16	0 to 65 535	The number of expected ACK/NACK packets received in *StatisticsDuration*
8	PacketsReceived	Unsigned16	0 to 65 535	The number of good packets received from the neighbor node *StatisticsDuration*
9	BroadcastPackets	Unsigned16	0 to 65 535	The number of good broadcast packets received from the neighbor node in *StatisticsDuration*

The data type of channel condition attributes ChannelCondition is shown in Table 18.

Table 18 – ChanCon_Struct structure

Attribute member identifier	Name	Data type	Valid range	Description
0	DeviceShortAddress	Unsigned16	0 to 65 535	16-bit address of device
1	ChannelID	Unsigned8	0 to 255	The sequence number of channel
2	NeighborAddr	Unsigned16	0 to 65 535	16-bit address of neighbor device
3	LinkQuality	Unsigned8	0 to 255	Link Quality Indication (LQI) value of every channel
4	PacketLossRate	Unsigned16	0 to 65 535	Packet loss rate of every channel, which is percentage and calculated through received ACKs and total sent packets
5	RetryNum	Unsigned8	0 to MaxFrameRetries	The count of retransmission of every channel

The data type of device attributes DeviceStruct is shown in Table 19.

Table 19 – Device_struct structure

Attribute member identifier	Name	Data type	Valid range	Storage type	Access type	Description
0	LongAddress	Unsigned64	0 to (2^{64}-1)	Constant	R	64-bit global unique address As for the *DeviceType* field as defined in 6.3.3: 0 = Gateway device; 1 = Routing device; 2 = Field device; 3 = Handheld device; Others are reserved.
1	RedundantDevFlag	Boolean	0, 1	Static	R/W	Flag that indicates whether this device is a redundant device: 0 = Irredundant device; 1 = Redundant device.
2	PrimaryDevAddr	Unsigned16	0 to 65 535	Static	R/W	The 16-bit short address of related primary device
3	NetAddressAssign	Boolean	0, 1	Static	R/W	A flag indicating whether the network address (see 6.3.3) has been assigned: 0 = No; 1 = Yes.
4	DeviceShortAddress	Unsigned16	0 to 65 535	Static	R/W	Short address of WIA-PA device The most significant 8 bits are cluster address and the least significant 8 bits are intra-cluster address
5	ManufacturerID	Unsigned24	0 to (2^{24}-1)	Constant	R	Manufacturer's identifier
6	DeviceSerialNum	Unsigned64	0 to (2^{64}-1)	Constant	R	Device serial number
7	PowerSupplyStatus	Unsigned8	0 to 10	Static	R/W	Power condition: 0 = Fix power supply; 1 = Highest power; 2 = Second highest power; …… 10= lowest power
8	RouterCapable	Boolean	0, 1	Static	R/W	A flag to indicate if the device has routing function: 0 = No; 1 = Yes.
9	Devicestate	Unsigned8	0 to 16	Static	R/W	Bit 0-1 represent device joining state: 00 = Not joined; 01 = Joining; 10 = Security authentication; 11 = Joined. Bit 2-3 represent device running state: 00 = Inactive; 01 = Active; 10 = Failed. Others are reserved.

Table 19 *(Continued)*

Attribute member identifier	Name	Data type	Valid range	Storage type	Access type	Description
10	DeviceMemorytotal	Unsigned32	0 to (2^{32}-1)	Consta-nt	R	Total memory in a device (in octets), including RAM, ROM, Flash…
11	DeviceUsedMemory	Unsigned32	0 to (2^{32}-1)	Dynamic	R	Memory used by device (in octets)
12	ClockMasterRole	Boolean	0, 1	Static	R/W	The device is a time source or not (see 8.3.2.1): 0 = No; 1 = Yes.
13	ClockUpdate	Unsigned32	0 to (2^{32}-1)	Dynamic	R	The last adjustment of clock (in seconds)
14	PacketsMACToDLSL	Unsigned32	0 to (2^{32}-1)	Dynamic	R	Count of the packets from MAC to DLSL
15	PacketsFromDLSL-Rejected	Unsigned32	0 to (2^{32}-1)	Dynamic	R	Count of the packets from the DLSL that are rejected by the NL
16	PacketsFromDLSL-Accepted	Unsigned32	0 to (2^{32}-1)	Dynamic	R	Count of the packets from the DLSL that are accepted by the NL
17	PacketsFromASL	Unsigned32	0 to (2^{32}-1)	Dynamic	R	Count of the packets from the ASL
18	PacketsFromASL-Rejected	Unsigned32	0 to (2^{32}-1)	Dynamic	R	Count of the packets from the ASL that are dropped by the NL
19	PacketsOutToDLSL	Unsigned32	0 to (2^{32}-1)	Dynamic	R	Count of the packets from the ASL that are forwarded by the NL to the DLSL
20	AGGSupportFlag	Boolean	0, 1	Static	R/W	Aggregation and disaggregation support flag (whether a routing device supports aggregation mechanism): 0 = Not support; 1 = Support.
21	AGGEnableFlag	Boolean	0, 1	Static	R/W	Aggregation and disaggregation enable flag (whether a routing device enables aggregation mechanism): 0 = Disenable; 1 = Enable.
22	IntraChannelNum	Unsigned8	0 to 31	Static	R/W	The number of usable channels for intra-cluster communication
23	IntraChanel	Unsigned8	0 to 31	Static	R/W	An array to store all communication channels, which is allocated by the GW to routing devices and field devices. The size of the array is *IntraChannelNum*. See 8.4.5 for details.
24	SuperframeID	Unsigned16	0 to 65 535	Static	R/W	Unique identifier of the superframe, supplied by the NM

The data type of VCR attributes VCR_Struct is shown in Table 20.

Table 20 – VCR_Struct structure

Attribute member identifier	Name	Data type	Valid range	Description
0	VcrID	Unsigned16	0 to 65 535	VCR identifier The VCR is invalid if its identifier is 0xFFFF.
1	VcrType	Unsigned8	0 to 16	Bit 0-1 represent VCR type: 00 = P/S VCR; 01 = R/S VCR; 10 = C/S VCR; Bit 2-3 represent aggregation VCR or not: 00 = Non-aggregation VCR; 01 = Data aggregation VCR; 10 = Packet aggregation VCR; Others are reserved.
2	SrcObjID	Unsigned8	0 to 255	Source object ID
3	SrcObjInsID	Unsigned8	0 to 255	ID of source object instance
4	DesObjID	Unsigned8	0 to 255	Destination object ID
5	DesObjInsID	Unsigned8	0 to 255	ID of destination object instance
6	DataUpdateRate	Unsigned32	0 to (2^{32}-1)	Data update rate (in milliseconds) The data has no cycle if this value is 0xFFFFFFFF.
7	VcrStatus	Boolean	0, 1	VCR status: 0 = Inactive; 1 = Active.
8	VcrActivationTime	Unsigned32	0 to (2^{32}-1)	The activation time of VCR (in milliseconds)
9	ServiceTime	Unsigned32	0 to (2^{32}-1)	The valid service duration of VCR (in milliseconds)
10	SourceChAddress	Unsigned16	0 to 65 535	The cluster head address of the source device
11	SourceAddress	Unsigned16	0 to 65 535	The network address of the source device
12	DestinationAddress	Unsigned16	0 to 65 535	The network address of the destination device
13	SecurityPolicy	Unsigned8	0 to 255	Security level of packets (see Annex A)
14	RouteID	Unsigned16	0 to 65 535	Route identifier
NOTE The time unit used in this table is the timeslot used by the superframe.				

The data type of device condition attributes DevConRep is shown in Table 21.

Table 21 – DevConRep_Struct structure

Attribute member identifier	Name	Data type	Valid range	Description
0	DevShortAddr	Unsigned16	0 to 65 535	16-bit short address of device
1	NumPktSent	Unsigned16	0 to 65 535	The total number of packet sent after the last report
2	NumPktRcvd	Unsigned16	0 to 65 535	The total number of packet terminated in the device after the last report
3	NumMacMicFailure	Unsigned16	0 to 65 535	The total number of Message Integrity Code (MIC) failure after the last report
4	BatLevel	Unsigned8	1 to 10	Residual power level
5	RestartCount	Unsigned8	0 to 255	Restart count of device
6	Uptime	Unsigned32	0 to (2^{32}-1)	Time from the last restart (in seconds)

The data type of key attributes KeyTable is shown in Table 22.

Table 22 – Key_Struct structure

Attribute member identifier	Name	Data type	Valid range	Description
0	KeyID	Unsigned16	0 to 65 535	Key identifier
1	KeyType	Unsigned8	0 to 3	Key type: 0 = Joining key; 1 = Key encryption key; 2 = DLSL encryption key; 3 = AL encryption key.
2	KeyLength	Unsigned8	0 to 255	(KeyLength+1) is the valid length of the key (in bits)
3	KeyActiveTime	Unsigned32	0 to (2^{32}-1)	Active time of key updating (in milliseconds)
4	KeyData	Octetstring		Key value.
5	KeyAttackTimes	Unsigned8	0 to 255	The total number of key attack
6	KeyState	Unsigned8	0 to 7	The using state of a key: 0 = Backup; 1 = Using; 2 = Invalid; Others are reserved.

The data type of object list attributes ObjList is shown in Table 23.

Table 23 – ObjList_Struct structure

Attribute member identifier	Name	Data type	Valid range	Description
0	ObjectID	Unsigned8	0 to 255	The unique identifier of the UAO
1	InstanceID	Unsigned8	0 to 255	The unique identifier of an object instance
2	ProfileID	Unsigned16	0 to 65 535	The profile ID for the UAO
3	ParameterNumber	Unsigned8	0 to 255	Number of parameters of the UAO
NOTE *ObjectID* and *InstanceID* in ObjList_Struct are identical with *Object Identifier* and *Instance ID* (see Annex C). DAGO, PAGO, and DGO are three special UAOs. Therefore, the attributes of DAGO, PAGO, or DGO are included in the ObjList_if such object is implemented in the device. In addition, *DagoID* and *DagoInsID* of DAGO, *PagoID* and *PagoInsID* of PAGO as well as *DgoID* and *DgoInsID* of DGO are respectively corresponding to the *ObjectID* and *InstanceID* in ObjList_Struct.				

6.9.2 MIB services

6.9.2.1 Remote MIB services

The attributes in the MIB can be read and written remotely through the attribute-getting and attribute-setting services provided by the NL.

6.9.2.2 Local MIB services

6.9.2.2.1 General

The attributes in the MIB can be read and written locally through the attribute-getting and attribute-setting services provided by the local DMAP.

6.9.2.2.2 DMAP attribute getting services

DMAP-MIB-GET.request is used by protocol layers to request attributes in the MIB.

The semantics of DMAP-MIB-GET request are as follows:

DMAP-MIB-GET.request (
AttributeID,
AttributeMemID,
FirstValueStorIndex,
Count
)

Table 24 specifies the parameters for DMAP-MIB-GET.request.

Table 24 – DMAP-MIB-GET.request parameters

Name	Data type	Valid range	Description
AttributeID	Unsigned8	0 to 255	Attribute ID in the MIB
AttributeMemID	Unsigned8	0 to 255	The identifier of attribute member, which is used to get the structured MIB attributes The value 255 means that all attributes should be read.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to get the structured MIB attributes
Count	Unsigned8	0 to 255	Number of attribute values or attributes member values, which is used to get the structured MIB attributes; Getting all attribute values from *FirstValueStorIndex* if *Count* = 0

DMAP-MIB-GET.confirm is used to return the result of DMAP-MIB-GET.request.

The semantics of DMAP-MIB-GET.confirm are as follows:

```
DMAP-MIB-GET.confirm (
                        Status,
                        AttributeID,
                        AttributeMemID,
                        FirstValueStorIndex,
                        Count,
                        AttributeValue
                      )
```

Table 25 specifies the parameters for DMAP-MIB-GET.confirm.

Table 25 – DMAP-MIB-GET.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Attribute getting results: 0 = SUCCESS; 1 = UNSUPPORTED_ATTRIBUTE; Others are reserved.
AttibuteID	Unsigned8	0 to 255	The requested attribute ID
AttributeMemID	Unsigned8	0 to 255	The identifier of attribute member The value 255 means that all attributes should be read.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple read attribute values
Count	Unsigned8	0 to 255	Number of attribute values or attributes member values
AttributeValue	Octetstring		The value of the attribute

If the operation of getting attributes is successful, the "*Status*" should be "*SUCCESS*" and the "*AttributeValue*" is valid; otherwise, if the MIB does not have the needed attributes, the "*Status*" should returns "*UNSUPPORTED_ATTRIBUTE*" and the "*AttributeValue*" is invalid.

6.9.2.2.3 DMAP attribute setting services

DMAP-MIB-SET.request should be used by the protocol layers to write attributes to the MIB.

The semantics of DMAP-MIB-SET.request are as follows:

```
DMAP-MIB-SET.request (
                        AttributeID,
                        AttributeMemID,
                        FirstValueStorIndex,
                        Count,
                        AttributeValue
                      )
```

Table 26 specifies the parameters for DMAP-MIB-SET.request.

Table 26 – DMAP-MIB-SET.request parameters

Name	Data type	Valid range	Description
AttributeID	Unsigned8	0 to 255	Attribute ID in the MIB
AttributeMemID	Unsigned8	0 to 255	Identifier of attribute member The value 255 means that all attributes should be read.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to get the structured MIB attributes
Count	Unsigned8	0 to 255	Number of attributes
AttributeValue	Octetstring		Value of the attribute

DMAP-MIB-SET.confirm is used to return the result of DMAP-MIB-SET.request.

The semantics of DMAP-MIB-SET.confirm are as follows:

DMAP-MIB-SET.confirm (
Status,
AttributeID,
AttributeMemID,
FirstValueStorIndex
)

Table 27 specifies the parameters for DMAP-MIB-SET.confirm.

Table 27 – DMAP-MIB-SET.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Attribute setting result: 0 = SUCCESS; 1 = UNSUPPORTED_ATTRIBUTE; 2 = INVALID_PARAMETER; Others are reserved.
AttributeID	Unsigned8	0 to 255	Attribute ID in the MIB
AttributeMemID	Unsigned8	0 to 255	Identifier of attribute member The value 255 means that all attributes should be read.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple written attribute values

If the operation of setting attributes is successful, the "*Status*" should be "*SUCCESS*"; if the MIB does not have the needed attributes, the "*Status*" should be "*UNSUPPORTED_ATTRIBUTE*"; otherwise, if the set attributes are not conformable to the specified attributes, the "*Status*" should be "*INVALID_PARAMETER*".

7 Physical Layer

The PHY is responsible for activation and deactivation of the radio transceiver, energy detection, link quality indicator, channel selection, clear channel assessment, transmitting and receiving packets across the physical medium.

The PHY specification of WIA-PA is based on the IEEE STD 802.15.4-2006 compliant radio. The PHY should support a minimum of sixteen channels, i.e. channels 11-26 specified in IEEE STD 802.15.4-2006. The PHY should support a basic air data rate of 250 kbit/s.

WIA-PA supports all mandatory items in IEEE 802.15.4-2006 PHY.

8 Data link layer

8.1 General

The WIA-PA Data Link Layer (DLL) is designed to guarantee communication among WIA-PA devices in a reliable and secure way in real-time. The DLL of WIA-PA extends the IEEE STD 802.15.4-2006 superframe structure. The WIA-PA DLL supports certain key functions, including frequency hopping, retransmission, and Time Division Multiple Access (TDMA) and Carrier Sense Multiple Access (CSMA) hybrid channel access mechanisms. These mechanisms are used to guarantee reliability and real-time transmission in communication. The WIA-PA DLL is designed to use MIC mechanism and encryption technology to guarantee the integrity and confidentiality of the communication process.

8.2 Protocol stack

The WIA-PA DLL is designed to leverage the IEEE STD 802.15.4-2006 to meet the requirements of process automation. The DLL protocol stack is shown in Figure 28.

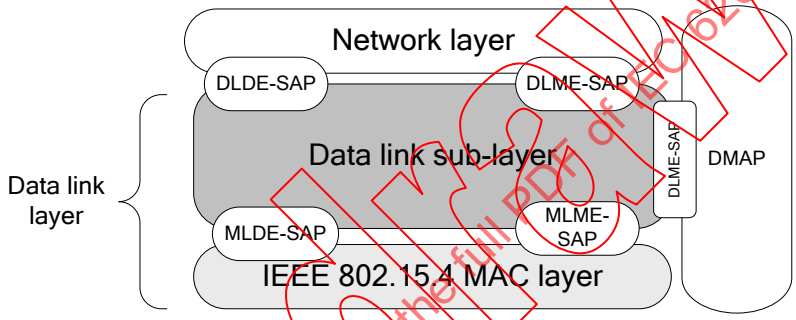

Figure 28 – WIA-PA DLL protocol stack

The WIA-PA DLL includes the following parts:

a) The IEEE STD 802.15.4-2006 MAC, which handles the mechanisms of sending and receiving individual data frames;

b) The DLSL, which handles the aspects of communication resource.

8.3 MAC overview and function extension

8.3.1 MAC overview

The following section specifies the MAC of the WIA-PA standard and the content of IEEE STD 802.15.4-2006 MAC is not stated in this specification. This specification is based on the IEEE STD 802.15.4-2006 beacon-enabled MAC, which handles all access to the physical radio channel and is responsible for the following tasks:

a) generating network beacons if the device is a gateway device or a routing device,

b) synchronizing to network beacons,

c) supporting network one-hop association and disassociation,

d) supporting device security,

e) employing the CSMA-CA mechanism for device joining,

f) handling and maintaining the GTS mechanism, and

g) providing a reliable link between two peer MAC entities.

Field devices and handheld devices correspond to the Reduced-Function Device (RFD), and routing devices and gateway devices correspond to the Full-Function Device (FFD). None of the optional functions of RFD and FFD in IEEE STD 802.15.4-2006 are required in this standard.

8.3.2 MAC function extension

8.3.2.1 General

This standard extends one MAC PIB attribute, primitives and command frames of Keep-alive and time synchronization. The extended PIB attribute is listed in Table 28. The identifiers, users and descriptions of the keep-alive command frame and the time synchronization command frame are listed in Table 28.

Table 28 – MAC extended PIB attributes

Attribute	Identifier	Type	Range	Description	Default
macBeaconNum	0x5e	Integer	1-15	The number of beacon transmissions in one timeslot	1

macBeaconNum is used to indicate the number of beacon transmissions in one timeslot.

Table 29 – MAC extended command frame

Command frame identifier	Command name	User	Description
10	Keep-alive command frame	Gateway device/Routing device/Field device	Indicating an existing device
11	Time synchronization command frame	Gateway device/ Routing device	Realizing time synchronization of the whole network

The Keep-alive command frame facilitates connection maintenance between neighbor devices. After devices join the network, their DMAPs send keep-alive command frame with the keep-alive request primitive. The gateway device sends the keep-alive command frame during the inter-cluster communication period in order to indicate that it is alive. Routing devices send the keep-alive command frames during the intra-cluster communication period in order to indicate that they are alive to their cluster members; in addition, routing devices send keep-alive command frames during the inter-cluster communication period in order to indicate that they are alive to other routing devices and to the gateway device. Field devices send the keep-alive command frames during the intra-cluster communication period in order to indicate that they are alive to their cluster heads.

In order to guarantee the reliability of the TDMA communication mode, devices in a network should synchronize with the time source. Time errors among devices are inevitable in spite of any hardware time sources. In order to overcome time clock drifting, the WIA-PA network performs two kinds of time synchronization: IEEE STD 802.15.4-2006 beacon and specifically designed time synchronization frame. In the mesh network, the gateway device is the UTC time source. All routing devices synchronize with the gateway device. In the star network, each routing device is the time source for its field devices, which synchronize with their routing devices.

NOTE This document specifies that the maximum synchronization error is less than 10 % of the basic timeslot length of the maximum superframe duration.

8.3.2.2 Extended primitives

8.3.2.2.1 MLME-KEEP-LIVE.request

MLME-KEEP-LIVE.request is used to send the Keep-alive command frame requested by the DLSL.

The semantics of MLME-KEEP-LIVE.request are as follows:

MLME-KEEP-LIVE.request (
)

8.3.2.2.2 MLME-KEEP-LIVE.confirm

MLME-KEEP-LIVE.confirm is used to respond to MLME-KEEP-LIVE.request.

The semantics of MLME-KEEP-LIVE.confirm are as follows:

```
MLME-KEEP-LIVE.confirm (
                          Status
                        )
```

Table 30 specifies the parameters for MLME-KEEP-LIVE.confirm.

Table 30 – MLME-KEEP-LIVE.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	The result of the keep alive: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

8.3.2.2.3 MLME-KEEP-LIVE.indication

MLME-KEEP-LIVE.indication is used to inform the DLSL that the keep-alive command frame has been successfully received.

The semantics of MLME-KEEP-LIVE.indication are as follows:

```
MLME-KEEP-LIVE.indication (
                             SrcAddr
                           )
```

Table 31 specifies the parameters for MLME-KEEP-LIVE.indication.

Table 31 – MLME-KEEP-LIVE.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	The source address

8.3.2.2.4 MLME-TIME-SYN.request

MLME-TIME-SYN.request is used to send the time synchronization command frame requested by the DLSL.

The semantics of MLME-TIME-SYN.request are as follows:

```
MLME-TIME-SYN.request (
                         TimeValue
                       )
```

Table 32 specifies the parameters for MLME-TIME-SYN.request.

Table 32 – MLME-TIME-SYN.request parameters

Name	Data type	Valid range	Description
TimeValue	Unsigned32	0 to (2^{32}-1)	Current time of device (in microsecond)

8.3.2.2.5 MLME-TIME-SYN.confirm

MLME-TIME-SYN.confirm is used to respond to MLME-TIME-SYN.request.

The semantics of MLME-TIME-SYN.confirm are as follows:

```
MLME-TIME-SYN.confirm (
                        Status
                      )
```

Table 33 specifies the parameters for MLME-TIME-SYN.confirm.

Table 33 – MLME-TIME-SYN.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	The result of time synchronization: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

8.3.2.2.6 MLME-TIME-SYN.indication

MLME-TIME-SYN.indication is used to inform the DLSL that the time synchronization command frame has been successfully received.

The semantics of MLME-TIME-SYN.indication are as follows:

```
MLME-TIME-SYN.indication (
                           SrcAddr,
                           TimeValue
                         )
```

Table 34 specifies the parameters for MLME-TIME-SYN.indication.

Table 34 – MLME-TIME-SYN.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	The source address
TimeValue	Unsigned32	0 to $(2^{32}-1)$	Current time of device (in microsecond)

8.3.2.3 Extended command frames

8.3.2.3.1 Beacon payload

The WIA-PA network uses the IEEE STD 802.15.4-2006 MAC beacon payload to distribute superframe information. See 7.2.2.1 in IEEE STD 802.15.4-2006 for the frame format of the MAC beacon.

The beacon payload is shown in Table 35.

Table 35 – Beacon payload

Length in octet(s)	1	6	4	1
Field name	ClusterID	ASN	Timevalue	NextBcnChannel

The subfields in Table 35 are defined as follows:

a) ClusterID indicates the identifier of the cluster;

b) ASN indicates the absolute timeslot number;

c) Timevalue indicates the time reference; and

d) NextBcnChannel indicates the channel used to transmit the next beacon.

8.3.2.3.2 Keep-alive command frame

The format of the keep-alive command frame is shown in Table 36.

Table 36– Format of keep-alive command frame

Length in octet(s)	See IEEE STD 802.15.4-2006, 7.2.2.4	1
Field name	MHR (IEEE STD 802.15.4-2006 MAC header)	Command frame identifier

The subfields in Table 36 are listed as follows:

a) See IEEE STD 802.15.4-2006 7.2.1 for MHR;

b) Command frame identifier is 10.

8.3.2.3.3 Time synchronization command frame

The time synchronization command frame is used to synchronize the entire network. The gateway device and the routing devices send the time synchronization command frames periodically. The gateway device sends the time synchronization command frame during the inter-cluster communication period. Routing devices send the time synchronization command frames during the intra-cluster communication period to synchronize their star networks and send them during the inter-cluster communication period to synchronize the mesh network.

The format of the time synchronization command frame is shown in Table 37.

Table 37– Format of time synchronization command frame

Length in octet(s)	See IEEE STD 802.15.4-2006, 7.2.2.4	1	4
Field name	MHR (IEEE STD 802.15.4-2006 MAC header)	Command frame identifier	Calibrated value of TimeValue

The subfields in Table 37 are defined as follows:

a) See IEEE STD 802.15.4-2006, 7.2.1 for MHR;

b) Command frame identifier is 11; and

c) See Table 32 for time calibration value.

8.4 DLSL function description

8.4.1 General

The DLSL provides a service interface between the NL and the MAC. The DLSL conceptually includes a Data Link Sub-Layer Data Entity (DLDE) and a Data Link Sub-Layer Management Entity (DLME). The DLDE provides data service interfaces. The DLME provides layer management services such as the configuration of the parameters of DLSL and the monitoring of the operation status of DLSL.

Figure 29 depicts the components and interfaces of the DLSL.

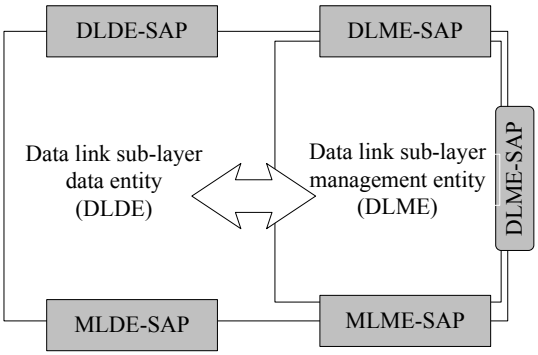

Figure 29 – WIA-PA DLSL reference model

The DLSL provides two services, which are accessed through the following two Service Access Points (SAPs):

a) The DLSL data service, which is accessed through the DLDE Service Access Point (DLDE-SAP);

b) The DLSL management service, which is accessed through the DLME-SAP.

In this document, the main function of the DLSL is to allocate communication resources among competitive users in order to avoid collisions, improve throughput, and increase bandwidth utilization. The main concepts of the DLSL include timeslot, superframe and link, which are defined as follows:

a) Timeslot is the basic time unit in packet exchange. The WIA-PA timeslot duration is configurable.

b) Superframe is a collection of timeslots repeating on a cyclic schedule. The number of timeslots in a given superframe determines the communication cycle for the WIA-PA devices that use the timeslots.

c) Link includes time and frequency. A link assignment specifies how the device uses a set of superframe timeslots. The link types include transmitting, receiving and transmit-shared. The sharing link allows more than one device to contend this link for packet exchange at the same time. The transmitting and receiving links should only allow designated devices to exchange packets.

8.4.2 Coexistence

The WIA-PA network should consider the following coexistence strategies:

a) The WIA-PA network extends the IEEE STD 802.15.4-2006 superframe structure;

b) The WIA-PA DLSL together with the NM achieves coexistence with other wireless networks. The WIA-PA DLSL incorporates several strategies to optimize coexistence:

- timeslot communication,
- low duty-cycle,
- multi-channel,
- frequency Hopping (FH), and
- collision avoidance.

8.4.3 Timeslot communication

The key requirement of timeslot communication is to guarantee that all transactions occur in a timeslot according to specific timing requirements. That is to say, all packets should be exchanged in a prescriptive timeslot and not be delayed. The timeslot length of WIA-PA DLSL is fully compatible with the timeslot length of IEEE STD 802.15.4-2006.

The timeslot duration is configured by the NM after devices join the network.

8.4.4 WIA-PA superframe

In order to guarantee real-time and reliable communication, this standard only takes account of the beacon-enabled IEEE STD 802.15.4-2006 superframe structure.

The WIA-PA superframe structure is shown in Figure 30.

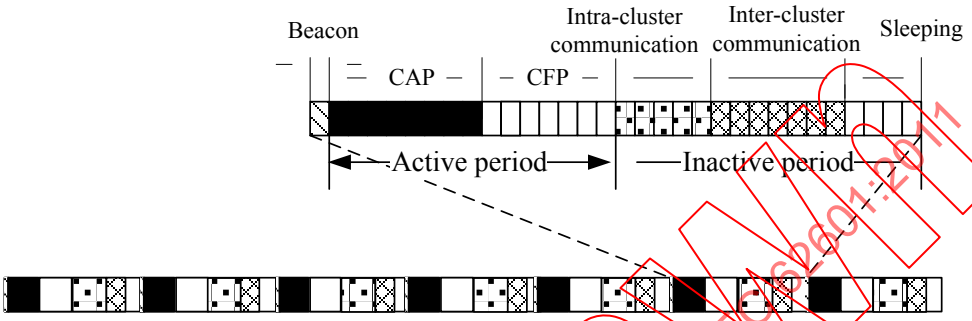

Figure 30 – WIA-PA superframe

The periods of the WIA-PA superframe are defined as follows:

a) The CAP period defined in the IEEE STD 802.15.4-2006 superframe is used for device joining, intra-cluster management and retry in the WIA-PA superframe;

b) The CFP period defined in the IEEE STD 802.15.4-2006 superframe is used for communication between handheld devices and their cluster head in the WIA-PA superframe; and

c) The inactive period defined in the IEEE STD 802.15.4-2006 superframe is used for intra-cluster communication, inter-cluster communication, and sleeping in the WIA-PA superframe.

The NM should generate WIA-PA superframes. The superframe lengths of routing devices are different, and are set as the lowest data update rate of all the members in a cluster. The timeslot types of WIA-PA superframe includes shared timeslots and dedicated timeslots. Shared timeslots are used for transmission of aperiodic data, and dedicated timeslots are used for intra- and inter-cluster transmission of periodic data.

Because the inactive period defined in the IEEE STD 802.15.4-2006 superframe is used for intra-cluster communication, inter-cluster communication, and sleeping in the WIA-PA superframe, the WIA-PA basic superframe duration is defined as thirty-two timeslots. The duration of the WIA-PA superframe is defined as 2^N (N is a natural integer) times the WIA-PA basic superframe duration.

8.4.5 Frequency hopping

The WIA-PA network supports frequency hopping, and the hopping sequence is designated by the NM. Frequency hopping in the WIA-PA network includes three mechanisms: AFS, AFH, and TH.

Adaptive Frequency Switch (AFS): in the WIA superframe, the beacon, CAP and CFP use the same channel in the same superframe cycle, and change the channel according to the channel conditions in different superframe cycles. That is to say, bad channel condition, which means that the packet drop rate is above "*PLRThreshold*", triggers the operation of changing channels. See 6.9.1.2.1 for "*PLRThreshold*".

Adaptive Frequency Hopping (AFH): irregularly changes communication channels per timeslot of the WIA superframe according to actual channel condition. The channel conditions are measured in retry times. If the channel condition is bad and the retry times of the sender reaches the value of "*ChannelThreshold*", the sender chooses the next channel in sequence from "*IntraChanel[ ]*" and notifies the receiver during the next retry timeslot by using the main channel (see Table 62). If the receiver does not receive the notification, it counts its retry times continuously. When the retry times of the receiver reach the value of "*ChannelThreshold*", the receiver chooses the next channel from "*IntraChanel[]*" during the (*ChannelThreshold+2*)th timeslot. If the receiver receives the notification of a channel switch, it changes the communication channel and returns ACK; otherwise, it does not change the communication channel and retry data by using the main channel. If the retry times of the sender reach "*macMaxFrameRetries*", the sender discards the current packet and transmits the next packet by using the main channel. If the communication between the sender and the receiver is successful before the retry times of the sender reaches "*macMaxFrameRetries*", the sender transmits the next packet by using the standby channel. The Intra-cluster period adopts the AFH mechanism. See 6.9.1.2.1 for "*ChannelThreshold*" and "*IntraChanel[]*"; See IEEE STD 802.15.4-2006 for the information of "*macMaxFrameRetries*".

NOTE The current channel in "*IntraChanel[]*" used by sender and receiver is marked as main channel, while other channels in "*IntraChanel[]*" are marked as standby channel.

Timeslot Hopping (TH): regularly changes communication channels per timeslot of the WIA superframe to combat interference and fading. The Inter-cluster period adopts the TH mechanism. The hopping structure is: <timeslot 1, channel 1> <timeslot 2, channel 2>… <timeslot i, channel i>.

The specific hopping mechanisms of DLSL in the WIA-PA network are shown in Table 38.

Table 38 – Hopping mechanisms

<table>
<tr><th>IEEE STD 802.15.4-2006</th><th>WIA-PA</th><th colspan="2">Basic MAC mechanism</th><th>DLSL Hopping mechanism</th></tr>
<tr><td>Beacon</td><td>Beacon</td><td>TDMA</td><td rowspan="3">Frequency Division Multiple Access (FDMA)</td><td rowspan="3">AFS</td></tr>
<tr><td>CAP</td><td>CAP</td><td>CSMA</td></tr>
<tr><td>CFP</td><td>CFP</td><td>TDMA</td></tr>
<tr><td rowspan="3">Inactive</td><td>Intra-cluster period</td><td colspan="2">TDMA</td><td>AFH</td></tr>
<tr><td>Inter-cluster period</td><td colspan="2">TDMA</td><td>TH</td></tr>
<tr><td>Sleeping</td><td colspan="2">-</td><td>-</td></tr>
</table>

Different routing devices use different channels in the active period. If the number of channels is not enough, the WIA-PA network uses the TDMA mechanism to enhance the system capacity. The start time of a superframe is configured by the NM. For example, there are three routing devices R1, R2, and R3 in the WIA-PA network. The superframe lengths of R1, R2 and R3 are respectively one, two and four WIA-PA basic superframe duration(s), as shown in Figure 31. According to the superframe definition, the active period of R1 cannot be multiplexed with the active periods of R2 and R3. However, the active periods of R2 and R3 may be multiplexed with each other. The active periods of R2 and R3 may use the same channel, while the active period of R1 should use a different channel from that of the active periods of R2 and R3.

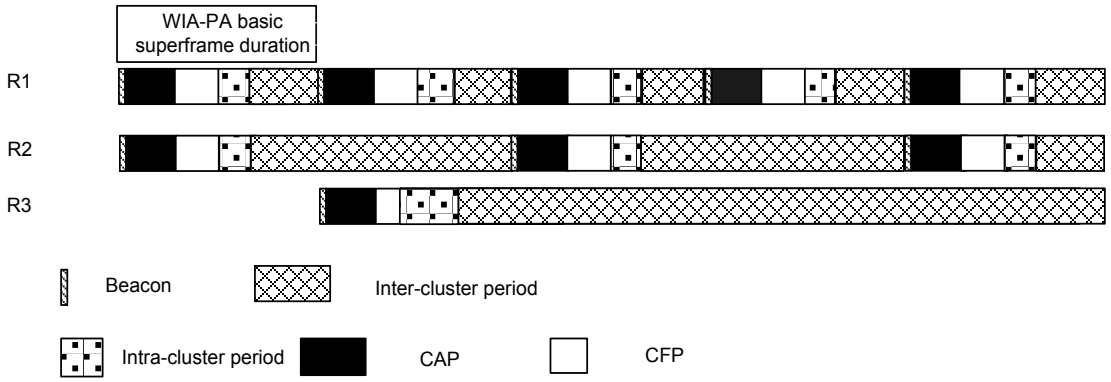

Figure 31 – R1, R2 and R3 superframe structures

8.4.6 Transmission of long cycle data

The long cycle data is defined as the data update rate of a device that is either greater than the maximum superframe length of IEEE STD 802.15.4-2006 or is greater than the data update rate of the routing device in a cluster.

To indicate the transmission of the long cycle data, this standard defines a parameter termed *TransmitFlag*. *TransmitFlag* is defined as follows:

$$TransmitFlag = \lceil (AbsoluteSlotNumber - ActiveSlot + 1) / NumberSlots \rceil \% SuperframeMultiple$$

$$AbsoluteSlotNumber = \lfloor (UTCTime - ActiveSlot \times TimeSlotDuratioin) / TimeSlotDuratioin \rfloor + ActiveSlot$$

See 6.9.1.2 for *UTCTime*, *ActiveSlot*, *NumberSlots*, and *SuperframeMultiple*.

In every superframe cycle, field devices receive the beacons and decide whether to send data in this superframe cycle. The process of long cycle data transmission is described as follows:

- If 0 < *TransmitFlag* < *SuperframeMultiple* and *TransmitFlag* = *LinkSuperframeNum*, then the field device transmits data in this superframe cycle.
- If *TransmitFlag* = 0 and *LinkSuperframeNum* = *SuperframeMultiple*, then the field device transmits data in this superframe cycle.

See Table 16 for *LinkSuperframeNum*.

Figure 32 is an example of long cycle data transmission.

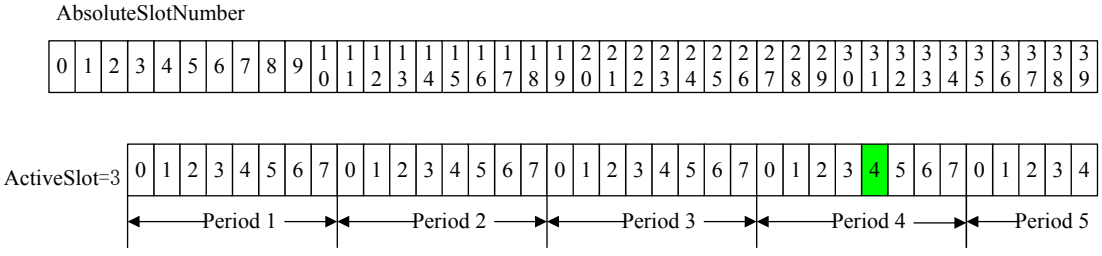

Figure 32 – An example of long cycle data transmission

In Figure 32, *ActiveSlot*=3, *NumberSlots* = 8, *SuperframeMultiple* = 4, and *LinkSuperframeNum* = 4.

If the current absolute slot of a beacon is 27 (*AbsoluteSlotNumber* = 27), *TransmitFlag* is calculated as follows:

$$TransmitFlag = \lceil (27 - 3 + 1)/8 \rceil \% 4 = 0$$

According to the calculation result, it can be concluded that *TransmitFlag* = 0 and that *SuperframeMultiple* = *LinkSuperframeNum*, which indicates that the packet should be sent during the current superframe cycle.

8.4.7 Retry strategy

The NM in the mesh network and the routing devices in the star network should allocate some timeslots for retries. The retry strategy supports the frequency hopping mechanisms (see 8.4.5).

NOTE The number of retry timeslots is bound by the constant "*macMaxFrameRetries*" in the IEEE STD 802.15.4-2006 MAC PAN Information Base (PIB).

8.4.8 Management service

The WIA-PA DLSL provides management services to the upper layer and the DMAP with DLME-SAP. The services include network discovery, device joining and leaving, resource allocation and operations of management database attributes (see Clause 6).

8.4.9 Radio link quality and channel condition measurement

Radio link quality and channel condition measurements include the link quality measurement of each pair of neighbors and the channel condition measurement of each channel. The collected information may be accumulated at the DLSL and reported through the DMAP. Link quality and channel condition are used to dynamically allocate communication resources.

Generally, the radio link quality measurement is performed according to the following performance indices:

a) Average signal level received from the neighbor devices in the statistics duration;

b) Non-broadcast packets sent to the neighbor devices in the statistics duration;

c) Count of ACK that is not received in the statistics duration; and

d) Count of the non-broadcast packets received from the neighbor devices in the statistics duration.

The channel condition measurement is performed according to the following performance data:

a) LQI per link;

b) Packet loss rate per link, which is determined by the count of received ACK and packets that are sent; and

c) Average count of retries per link.

8.4.10 Security

The WIA-PA DLSL applies the MIC mechanism and encryption technology to guarantee the integrity and confidentiality of the communication. See Clause 11 for details.

8.4.11 DLSL state machine

The WIA-PA DLSL state machine is shown inFigure 33.

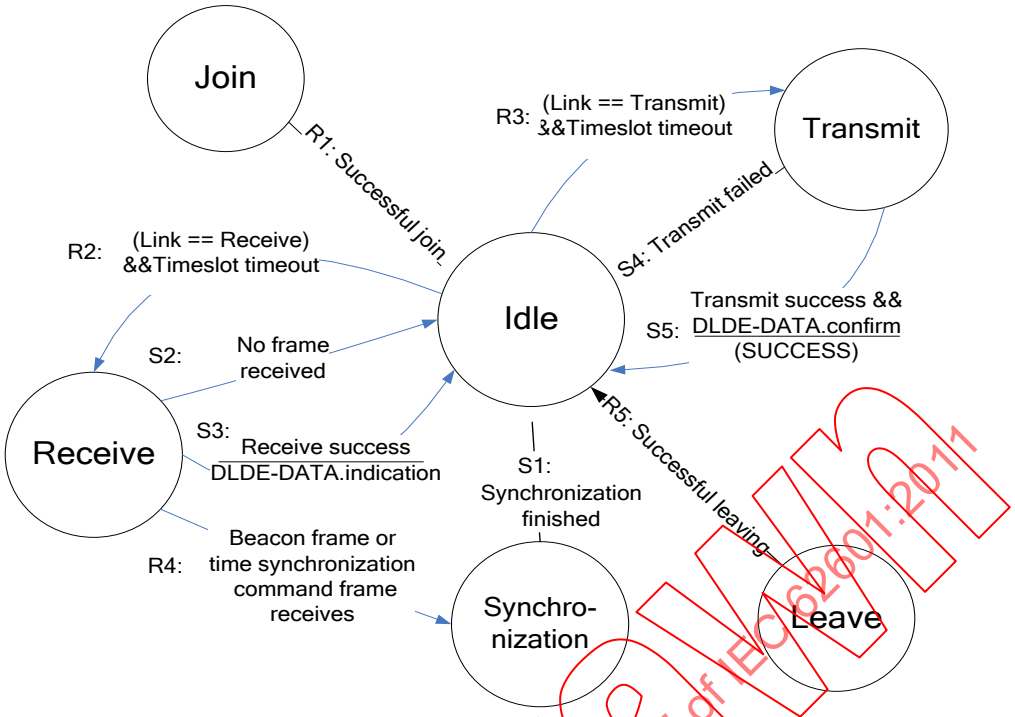

Figure 33 – DLSL state machine

The DLSL state transitions are shown in Table 39.

Table 39 – DLSL state transitions

Sequence number #	Current state	Event or condition = > actions	Next state
R1	Join	Successfully join	Idle
R2	Idle	Timeslot timeout and Link = Receive	Receive
R3	Idle	Timeslot timeout and Link = Transmit	Transmit
R4	Receive	Beacon frame or time synchronization command frame receives	Synchronization
R5	Leave	Successfully leave	Idle
S1	Synchronization	Synchronization finished	Idle
S2	Receive	No frame received	Idle
S3	Receive	Receive success = > DLDE-DATA. indication()	Idle
S4	Transmit	Transmit failed	Idle
S5	Transmit	Transmit success = > DLDE-DATA. confirm()	Idle

The states in DLSL state machine are specified as follows:

a) Join

This state handles the joining procedure of a device. After a device joins the network, the DLSL state machine runs.

b) Leave

This state handles the leaving procedure of a device.

c) Idle state

After a device joins the network, the DLSL enters the “Idle” state. The following transitions may occur while the DLSL is in the “Idle” state:

- When a timeslot arrives, the DLSL enters either the “Transmit” state or the “Receive” state according to the link options (transmitting or receiving).

d) Transmit state

When the timeslot arrives and the link option is a transmit link, the DLSL enters the “Transmit” state. The following events will happen while the DLSL is in the “Transmit” state.

- Successful propagation of a frame with a broadcast/multicast destination address happens as soon as the frame is transmitted. The frame’s buffer is then released.
- Successful propagation of a frame with a unicast destination address occurs when a validated and successful confirmation is received from the local MAC layer. This indicates that message propagation has been completed successfully; the frame’s buffer is then released.
- If a failure confirmation is received from the local MAC layer, the frame should be retried.
- If a response with an error or no response is received, the frame should be re-scheduled or retried.

e) Receive state

The frames that a device can receive include frames whose final destination addresses are this device and frames whose final destination addresses are not this device. The functions of the “Receive” state are to receive, check and process the frame. The following transitions may occur while the DLSL is in the “Receive” state:

- If there is no frame received, the DLSL will evaluate the link and return to the “Idle” state.
- If a beacon frame or a time synchronization command frame is received, then the “Synchronization” state is entered.
- Upon successful frame reception, the DLSL will process the frame according to frame priority (receive or discard), and then the device returns directly to the the “Idle” state.

f) Synchronization

In this state, time synchronization is executed after a device receives a beacon frame or a time synchronization command frame, and then the DLSL enters “Idle” state.

8.5 Data link sub-layer data services

8.5.1 General

The DLDE-SAP supports the point-to-point transmission of DLSL Protocol Data Units (DLPDUs) between devices. The primitives supported by DLSL data services include DLDE-DATA.request, DLDE-DATA.confirm, and DLDE-DATA.indication.

8.5.2 DLDE-DATA.request

The DLDE receives the payload from the NL through a DLDE-DATA.request and adds it to the message queue of the DLSL.

The semantics of DLDE-DATA.request are as follows:

DLDE-DATA.request (
NetworkID,
SrcAddrMode,
SrcAddr,
DstAddrMode,
DstAddr,
Priority,
Type,
PayloadLength,
Payload,
PayloadHandle
)

Table 40 specifies the parameters for DLDE-DATA.request.

Table 40 – DLDE-DATA.request parameters

Name	Data type	Valid range	Description
NetworkID	Unsigned8	0 to 255	Network identifier
SrcAddrMode	Unsigned8	0 to 3	Mode of source address: 0 = No address; 1 = Reserved; 2 = 16-bit short address; 3 = 64-bit long address.
SrcAddr	Unsigned16/64	0 to 65 535 or (2^{64}-1)	Source address 64-bit long address is used only in device joining process, and 16-bit short address is used generally.
DstAddrMode	Unsigned8	0 to 3	Mode of destination address: 0 = No address; 1 = Reserved; 2 = 16-bit short address; 3 = 64-bit long address.
DstAddr	Unsigned16/64	0 to 65 535 or (2^{64}-1)	Destination address, 16- or 64-bit
Priority	Unsigned8	0 to 15	Priority of the payload
Data type	Unsigned8	0 to 1	0 = Intra-cluster transmission; 1 = Inter-cluster transmission.
PayloadLength	Unsigned8	≤MaxMACFrameSize	Length of payload
Payload	Octetstring		Payload
PayloadHandle	Unsigned8	0 to 255	Handle allocated when call DLDE-DATA.request

8.5.3 DLDE-DATA.confirm

The semantics of DLDE-DATA.confirm are as follows:

DLDE-DATA.confirm (
PayloadHandle,
Status
)

Table 41 specifies the parameters for DLDE-DATA.confirm.

Table 41 – DLDE-DATA.confirm parameters

Name	Data type	Valid range	Description
PayloadHandle	Unsigned8	0 to 255	Handle allocated when call DLDE-DATA.confirm
Status	Unsigned8	0 to 255	Result of the data transmission of DLSL: 0 = SUCCESS; 1 = TRANSACTION_OVERFLOW; 2 = TRANSACTION_EXPIRED; 3 = NO_ACK; 4 = CHANNEL_ACCESS_FAILURE; 5 = UNAVAILABLE_KEY; 6 = FAILED_SUCURITY_CHECK; 7 = INVALID_PARAMETER; Others are reserved. See Table 42 for more detail

Table 42 – Status table

ID	Value	Description
0	SUCCESS	The requested operation is completed successfully. For a transmission request, this value indicates a successful transmission.
1	TRANSACTION_OVERFLOW	Not enough space for storing transactions
2	TRANSACTION_EXPIRED	Transaction has expired and its information is discarded
3	NO_ACK	No acknowledgement message is received after *macMaxFrameRetries*
4	CHANNEL_ACCESS_FAILURE	Can not transmit due to channel access failure
5	UNAVAILABLE_KEY	No valid key in access control list
6	FAILED_SECURITY_CHECK	Received packet makes a security checking error in security mode
7	INVALID_PARAMETER	A parameter in the primitive is out of value range
8	NO_BEACON	A scan operation failed to find any network beacons
9 to 255	reserved	

8.5.4 DLDE-DATA.indication

The semantics of DLDE-DATA.indication are as follows:

```
DLDE-DATA.indication (
                        NetworkID,
                        ScrAddrMode,
                        SrcAddr,
                        Type,
                        Priority,
                        PayloadLength,
                        Payload,
                        PayloadLinkQuality,
                    )
```

Table 43 specifies the parameters for DLDE-DATA.indication.

Table 43 – DLDE-DATA.indication parameters

Name	Data type	Valid range	Description
NetworkID	Unsigned8	0 to 255	Network identifier
SrcAddrMode	Unsigned8	0 to 3	The source address mode. Four options are available: 0 = No address; 1 = Reserved; 2 = 16-bit short address; 3 = 64-bit long address.
SrcAddr	Unsigned16/64	0 to 65 535 or (2^{64}-1)	Source address, using 64-bit long address only in device join process, and using 16-bit short address generally
Type	Unsigned8	0 to 1	0 = Intra-cluster transmission; 1 = Inter-cluster transmission.
Priority	Unsigned8	0 to 15	Priority of payload
PayloadLength	Unsigned8	≤MaxMACFrameSize	Length of payload
Payload	Octetstring		Payload
PayloadLinkQuality	Unsigned8	0 to 255	LQI value measured during reception of the DLPDU Lower values represent lower link quality.

8.5.5 Time sequence of DLSL data service

Figure 34 shows the data transaction sequence of transporting one data packet from the source device to the destination device.

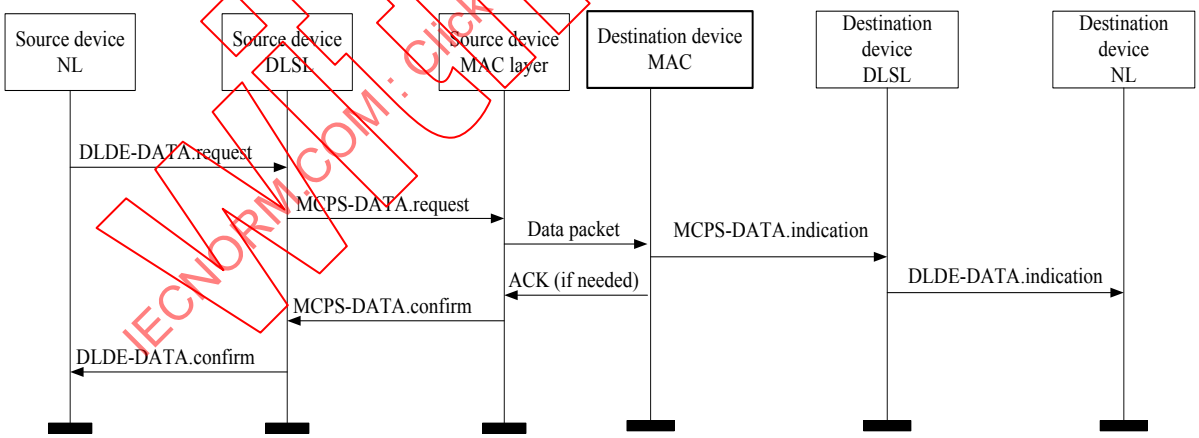

Figure 34 – Time sequence of data service

DLDE-DATA.request is generated by a local Network Layer Data Entity (NLDE) when a Network Protocol Data Unit (NPDU) is to be transferred to a peer NLDE.

On receipt of DLDE-DATA.request, DLDE inserts the message into the sending buffer and begins sending the supplied DLPDU.

DLDE-DATA.indication is generated by the DLDE of a destination device and issued to the NLDE on receipt of a data frame at the local DLDE that passes the appropriate message filtering operations.

DLDE-DATA.confirm is generated by the DLDE of a source device in response to DLDE-DATA.request. DLDE-DATA.confirm returns a status indicating the result of the transmission.

NOTE Refer to IEEE STD 802.15.4-2006 for the detailed message sequence chart between peer MAC entities. This note is also available for other time sequences.

8.6 Data link sub-layer management services

8.6.1 General

The upper layer uses the DLME-SAP to send management commands to the DLSL. The DLSL management services include subnet discovery, device joining and leaving, channel condition collection and report, neighbor information collection and report, and time synchronization.

8.6.2 Network discovery services

8.6.2.1 General

The DLSL network discovery services are used to scan a given list of communication channels. One device may use the network discovery services to search for cluster heads sending beacon frames within its communication scope. The primitives supported by the DLSL network discovery services include DLME-DISCOVERY.request and DLME-DISCOVERY.confirm.

8.6.2.2 DLME-DISCOVERY.request

DLME-DISCOVERY.request is used to request a device to scan channels.

The semantics of DLME-DISCOVERY.request are as follows:

```
DLME-DISCOVERY.request (
                        ScanChannels,
                        ScanDuration
                      )
```

Table 44 specifies the parameters for DLME-DISCOVERY.request.

Table 44 – DLME-DISCOVERY.request parameters

Name	Data type	Valid range	Description
ScanChannels	Unsigned32	32-bit Bits-Map	Higher 27 bits indicate IEEE STD 802.15.4-2006 channels of 2.4G frequency band; others are set up based on real field frequency channel (if less than 32-bit, the remaining bits are filled with 0). See IEEE 802.15.4-2006, 7.1.11.1.1 for the definition of higher 27 bits.
ScanDuration	Unsigned8	0 to 14	A value used to calculate the length of time to spend scanning each channel of ED, active, and passive scans This parameter is ignored for orphan scans. The time spent scanning each channel is: *aBaseSuperframeDuration* × (2^n + 1) symbols, where *n* is the value of *ScanDuration* parameter.
NOTE The definition and value set of *aBaseSuperframeDuration* refer to the parameters of PIB in the IEEE STD 802.15.4-2006 standard.			

8.6.2.3 DLME- DISCOVERY.confirm

DLME-DISCOVERY.confirm is used to respond to DLME-DISCOVERY.request.

The semantics of DLME-DISCOVERY.confirm are as follows:

```
DLME-DISCOVERY.confirm(
                        Status,
                        NetworkCount,
                        NetworkDescriptor
                      )
```

Table 45 specifies the parameters for DLME- DISCOVERY.confirm.

Table 45 – DLME- DISCOVERY.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Scan results: 0 = SUCCESS; 7 = INVALID_PARAMETER; 8 = NO_BEACON; Others are reserved. See Table 42 for more detail
NetworkCount	Unsigned8	0 to 255	The count of active network found during scan
NetworkDescriptor	Network descriptor list	0 to NetworkCount	Network Descriptor list of every network found, refer to Table 46

Table 46 – Network descriptor list

Name	Data type	Valid range	Description
LogicalChannel	Unsigned8	0 to 31	Logic channel used for joining, chosen from valid channels supported by PHY
BeaconOrder	Unsigned8	0 to 15	The frequency sending beacon frame
SuperframeOrder	Unsigned8	0 to 15	Active period length of the superframe
PermitJoining	Boolean	0, 1	Whether routing device permits field device to join: 0 = At least one device is permitted to join; 1 = No permit.

If the scan is successful, DLME-DISCOVERY.confirm returns "*SUCCESS*"; however if no beacons are found, DLME-DISCOVERY.confirm returns "*NO_BEACON*"; if there are some errors or invalid parameters in DLME-DISCOVERY.request, DLME-DISCOVERY.confirm returns "*INVALID_PARAMETER*".

8.6.2.4 Time sequence of subnet discovery

The time sequence for subnet discovery is shown in Figure 35. See IEEE STD 802.15.4-2006 for the primitives of MLME-SCAN.request and MLME-SCAN.confirm.

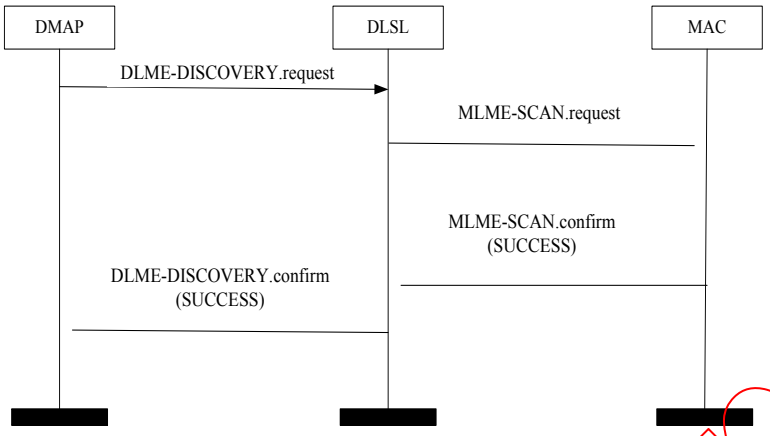

Figure 35 – Time sequence of network discovery

8.6.3 Device joining services

8.6.3.1 General

There are two instances requiring device joining services: (1) A new field device joining the star network or (2) a new routing device joining the mesh network. The primitives supported by DLSL device joining services include DLME-JOIN.request, DLME-JOIN.indication, DLME-JOIN.response, and DLME-JOIN.confirm.

8.6.3.2 DLME-JOIN.request

DLME-JOIN.request is used for a device to join the network (star or mesh).

The semantics of DLME-JOIN.request are as follows:

```
DLME-JOIN.request (
                    LogicalChannel,
                    JoinAddr,
                    PhyAddr,
                    DeviceType
                  )
```

Table 47 specifies the parameters for DLME-JOIN.request.

Table 47 – DLME-JOIN.request parameters

Name	Data type	Valid range	Description
LogicalChannel	Unsigned8	0 to 31	Logic channel used for joining, chosen from valid channels supported by PHY
JoinAddr	Unsigned16	0 to 65 535	The short address of routing device or gateway device that accepts joining request
PhyAddr	Unsgned64	0 to (2^{64}-1)	Physical address of the new device waiting to join
DeviceType	Unsigned8	0 to 255	Type of the new device waiting to join: 0 = Gateway device; 1 = Routing device; 2 = Field device; 3 = Handheld device; Others are reserved.

8.6.3.3 DLME- JOIN.indication

DLME-JOIN.indication is used to inform the NLME of a routing device or to inform the gateway device that the joining request from one device has been successfully received.

The semantics of DLME- JOIN.indication are as follows:

```
DLME-JOIN.indication (
                        PhyAddr,
                        DeviceType
                    )
```

Table 48 specifies the parameters for DLME-JOIN.indication.

Table 48 – DLME-JOIN.indication parameters

Name	Data type	Valid range	Decription
PhyAddr	Unsigned64	0 to (2^{64}-1)	Address of new device waiting to join
DeviceType	Unsigned8	0 to 255	Type of device waiting to join: 0 = Gateway device; 1 = Routing device; 2 = Field device; 3 = Handheld device; Others are reserved.

8.6.3.4 DLME-JOIN.response

DLME-JOIN.response is the response of DLME-JOIN.indication.

The semantics of DLME-JOIN.response are as follows:

```
DLME-JOIN.response (
                        PhyAddr,
                        ShortAddr,
                        TimeSource,
                        Status
                    )
```

Table 49 specifies the parameters for DLME-JOIN.response.

Table 49 – DLME-JOIN.response parameters

Name	Data type	Valid range	Description
PhyAddr	Unsigned64	0 to (2^{64}-1)	Address of device waiting to join
ShortAddr	Unsigned16	0 to 65 535	Short address allocated by the NM to device waiting to join
TimeSource	Unsigned8	0 to 1	Whether this device is set as time source: 0 = Not time source; 1 = Time source.
Status	Unsigned8	0 to 255	Result of joining request: 0 = SUCCESS; 1 = FAILURE_TOP_DISMATCH; 2 = FAILURE_ELSE; Others are reserved.

8.6.3.5 DLME-JOIN.confirm

DLME-JOIN.confirm reports the joining result to the DMAP.

The semantics of DLME-JOIN.confirm are as follows:

```
DLME-JOIN.confirm (
                    ShortAddr,
                    Status
                  )
```

Table 50 specifies the parameters for DLME-JOIN.confirm.

Table 50 – DLME-JOIN.confirm parameters

Name	Data type	Valid range	Description
ShortAddr	Unsigned16	0 to 65 535	Short address allocated by NM for device waiting to join
Status	Unsigned8	0 to 255	Result of joining request: 0 = SUCCESS; 1 = FAILURE_TOP_DISMATCH; 2 = FAILURE_ELSE; Others are reserved.

8.6.3.6 Time sequence for device joining in the network

See 9.5.3 for the detailed joining process

8.6.4 Device leaving services

8.6.4.1 General

There are two instances requiring device leaving services: (1) A field device leaving the star network or (2) a routing device leaving the mesh network. The primitives supported by DLSL network leaving services include DLME-LEAVE.request, DLME-LEAVE.indication, and DLME-LEAVE.confirm.

8.6.4.2 DLME-LEAVE.request

The semantics of DLME-LEAVE.request are as follows:

```
DLME-LEAVE.request (
                     ShortAddr
                   )
```

Table 51 specifies the parameters for DLME-LEAVE.request.

Table 51 – DLME-LEAVE.request parameters

Name	Data type	Valid range	Description
ShortAddr	Unsigned16	0 to 65 535	The short address of the device asking to leave

8.6.4.3 DLME-LEAVE.indication

DLME-LEAVE.indication is used to indicate to the upper layer that a device leaving request has been received.

The semantics of DLME-LEAVE.indication are as follows:

```
DLME-LEAVE.indication (
```

ShortAddr
)

Table 52 specifies the parameters for DLME-LEAVE.indication.

Table 52 – DLME-LEAVE.indication parameters

Name	Data type	Valid range	Description
ShortAddr	Unsigned16	0 to 65 535	The short address of the device asking to leave

8.6.4.4 DLME-LEAVE.confirm

DLME-LEAVE.confirm is used to report the result of DLME-LEAVE.request.

The semantics of DLME-LEAVE.confirm are as follows:

DLME-LEAVE.confirm (
Status
)

Table 53 specifies the parameters for DLME-LEAVE.confirm.

Table 53 – DLME-LEAVE.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Result of leaving request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

8.6.4.5 Time sequence for device leaving

See IEEE STD 802.15.4-2006 for the following primitives: MLME-DISASSOCIATE.request, MLME-DISASSOCIATE.confirm, and MLME-DISASSOCIATE.indication.

The time sequences for devices leaving the network are listed as follows:

a) The time sequence for routing devices leaving the network

Routing devices connect to both the mesh network and the star network. Therefore, the process of routing devices leaving the network includes routing devices leaving the mesh network and routing devices leaving the star network. According to the leaving originator, the process of routing devices leaving the network includes active leaving and passive leaving.

b) The time sequence for field devices leaving the network

According to the leaving source, the process of field devices leaving the network includes active leaving and passive leaving.

See 9.5.4 for detailed process of device leaving.

8.6.5 DLME-CHANNEL-CONDITION.indication

DLME-CHANNEL-CONDITION.indication is used by DLSL to report channel condition information to DMAP. The reported performance data includes LQI, packet loss rate, and count of retries.

The semantics of DLME-CHANNEL-CONDITION.indication are as follows:

DLME-CHANNEL-CONDITION.indication (
NeighborAddr,
ChannelID,
LinkQuality,
PacketLossRate,
RetryNum
)

Table 54 specifies the parameters for DLME-CHANNEL-CONDITION.indication.

Table 54 – DLME-CHANNEL-CONDITION.indication parameters

Name	Data type	Valid range	Description
NeighborAddr	Unsigned16	0 to 65 535	16-bit short address of neighbor device
ChannelID	Unsigned8	0 to 31	Channel identifier
LinkQuality	Unsigned8	0 to 255	LQI per channel
PacketLossRate	Unsigned8	0 to 100	Packet loss rate per channel
RetryNum	Unsigned8	0 to *macMaxFrameRetries*	Count of retry per channel

8.6.6 DLME-NEIGHBOR-INFO.indication

DLME-NEIGHBOR-INFO.indication is used to report the collected neighbor information to the DMAP and is used to update the neighbor attributes.

The semantics of DLME-NEIGHBOR-INFO.indication are as follows:

DLME-NEIGHBOR-INFO.indication (
NeighborCount,
NeighborStructure
)

Table 55 specifies the parameters for DLME-NEIGHBOR-INFO.indication.

Table 55 – DLME-NEIGHBOR-INFO.indication parameters

Name	Data type	Valid range	Description
NeighborCount	Unsigned16	0 to 65 535	Count of neighbor devices
NeighborStructure	Neighbor_Struct structure		Information of Neighbors

8.6.7 DLME-COMM-STATUS.indication

DLME-COMM-STATUS.indication is used to report the joining status to the DMAP.

The semantics of DLME-COMM-STATUS.indication are as follows:

DLME-COMM-STATUS.indication (
PhyAddr,
Status
)

Table 56 specifies the parameters for DLME-COMM-STATUS.indication.

Table 56 – DLME-COMM-STATUS.indication parameters

Name	Data type	Valid range	Description
PhyAddr	Unsigned64	0 to (2^{64}-1)	Address of new device that has just joined the network
Status	Unsigned8	0 to 255	The joining status of a device: 0 = SUCCESS; 1 = FAILURE_TOP_DISMATCH; 2 = FAILURE_ELSE; Others are reserved..

8.6.8 Keep-alive services

8.6.8.1 DLME-KEEP-LIVE.request

DLME-KEEP-LIVE.request is used to send Keep-alive command frames that are requested by the DMAP.

The semantics of DLME-KEEP-LIVE.request are as follows:

DLME-KEEP-LIVE.request (
)

8.6.8.2 DLME-KEEP-LIVE.confirm

DLME -KEEP-LIVE.confirm is used to respond to DLME -KEEP-LIVE.request.

The semantics of DLME -KEEP-LIVE.confirm are as follows:

DLME -KEEP-LIVE.confirm (
Status
)

Table 57 specifies the parameters for DLME -KEEP-LIVE.confirm.

Table 57 – DLME -KEEP-LIVE.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	The result of the keep alive: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

8.6.8.3 DLME-KEEP-LIVE.indication

DLME-KEEP-LIVE.indication is used to inform the DMAP that the keep-alive command frame has been successfully received.

The semantics of DLME -KEEP-LIVE.indication are as follows:

DLME -KEEP-LIVE.indication (
SrcAddr
)

Table 58 specifies the parameters for DLME -KEEP-LIVE.indication.

Table 58 – DLME -KEEP-LIVE.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	The source address

8.6.9 Time synchronization services

8.6.9.1 DLME-TIME-SYN.request

DLME-TIME-SYN.request is used to send time synchronization command frames that are requested by the DMAP.

The semantics of DLME-TIME-SYN.request are as follows:

```
DLME-TIME-SYN.request (
                        TimeValue
                      )
```

Table 59 specifies the parameters for DLME-TIME-SYN.request.

Table 59 – DLME-TIME-SYN.request parameters

Name	Data type	Valid range	Description
TimeValue	Unsigned32	0 to $(2^{32}-1)$	Current time of device (in microsecond)

8.6.9.2 DLME-TIME-SYN.confirm

DLME-TIME-SYN.confirm is used to respond to DLME-TIME-SYN.request.

The semantics of DLME-TIME-SYN.confirm are as follows:

```
DLME-TIME-SYN.confirm (
                        Status
                      )
```

Table 60 specifies the parameters for DLME-TIME-SYN.confirm.

Table 60 – DLME -TIME-SYN.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	The result of time synchronization: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

8.6.9.3 DLME-TIME-SYN.indication

DLME-TIME-SYN.indication is used to inform the DMAP that the time synchronization command frame has been successfully received.

The semantics of DLME-TIME-SYN.indication are as follows:

```
DLME-TIME-SYN.indication (
                           SrcAddr,
                           TimeValue
                         )
```

Table 61 specifies the parameters for DLME-TIME-SYN.indication.

Table 61 – DLME-TIME-SYN.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	The source address
TimeValue	Unsigned32	0 to (2^{32}-1)	Current time of device (in microsecond)

8.7 DLSL frame formats

8.7.1 General frame format

The DLSL general frame format is illustrated in Figure 36.

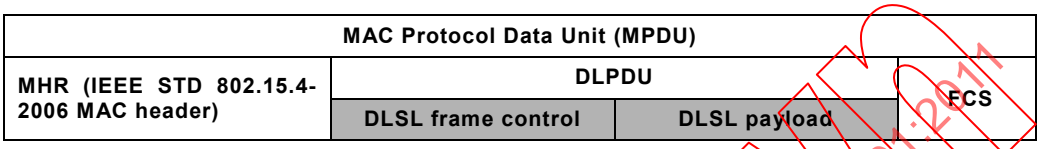

Figure 36 – General frame format

The DLSL frame is composed of:

a) IEEE STD 802.15.4-2006 MAC Header (MHR) (See IEEE STD 802.15.4-2006);
b) DLSL frame control (see Table 62);
c) DLSL payload; and
d) Frame Check Sequence (FCS).

NOTE See 11.3.3 for the content of security.

The format of the WIA-PA DLSL frame control is shown inTable 62.

Table 62 – DLSL frame control filed

Length in bit(s)	1	1	1	5
Subfield name	Frame type	Clock source	Security enable	Index of main channel

The subfields in Table 62 are listed as follows:

a) The frame type subfield has 1-bit length and is used to specify the frame type. If a frame is a data frame, this subfield is set to 0; otherwise, this subfield is set to 1.
b) The clock recipient subfield has 1-bit length and is used to indicate whether this device is a clock source. If the device is a clock source, this subfield is set to 0; otherwise, this subfield is set to1.
c) The security enable subfield has 1-bit length and is used to instruct DLSL whether or not to use the security mechanism. A value of 0 means that the security mechanism is disabled, and a value of 1 means that the security mechanism is enabled.
d) The index of the main channel is used to indicate the current communication channel.

NOTE All multiple octet fields should be transmitted or received with the least significant octet first and each octet should be transmitted or received with the Least Significant Bit (LSB) first. The same transmission order should apply to data fields transferred between layers.

8.7.2 Date frame format

The format of the DLSL data frame is shown in Table 63.

Table 63 – Date frame format

Length in octet(s)	1	Variable length
Field name	DLSL frame control	Data payload

The subfields in Table 63 are listed as follows:

a) The DLSL frame control is defined in Table 62; and

b) The data payload has variable length and is filled with the DLSL data.

8.7.3 Command frame format

8.7.3.1 General command frame format

The general command frame format is shown in Table 64.

Table 64 – General command frame format

Length in octet(s)	1	1	Variable length
Field name	DLSL frame control	Command frame identifier	Command payload

The subfields in Table 64 are listed as follows:

a) The DLSL frame control is defined in Table 62;

b) The command frame identifier is defined in Table 65; and

c) The command payload has variable length and is filled with the DLSL management data.

8.7.3.2 DLSL command frame

The DLSL command frames are to be extended and are shown in Table 65.

Table 65 – DLSL command frame

Command frame identifier	Command name	User	Description
1 to 255	Reserved	Reserved	To be extended

9 Network layer

9.1 General

The WIA-PA network layer (NL) receives and transports packets over networks, provides interfaces to the ASL, and carries out network layer management, configuration and control.

9.2 Protocol stack

The protocol stack of the WIA-PA network layer is shown in Figure 37.

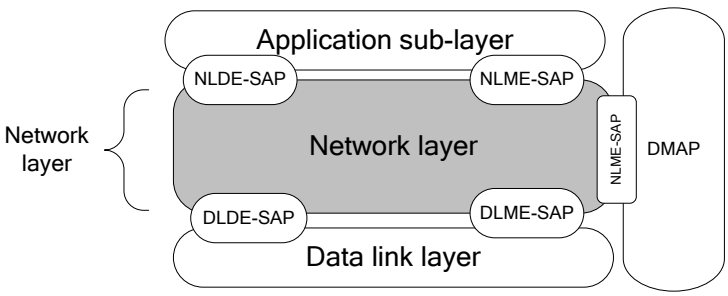

Figure 37 – WIA-PA network layer protocol stack

The layers and entities in Figure 37 are defined as follows:

a) The NL defines the NLDE and the Network Layer Management Entity (NLME);

b) The NLDE provides the service interface through which the ASL transmits and receives data; and

c) The NLME provides the service interfaces through which the layer management functions are invoked by the upper layer and DMAP.

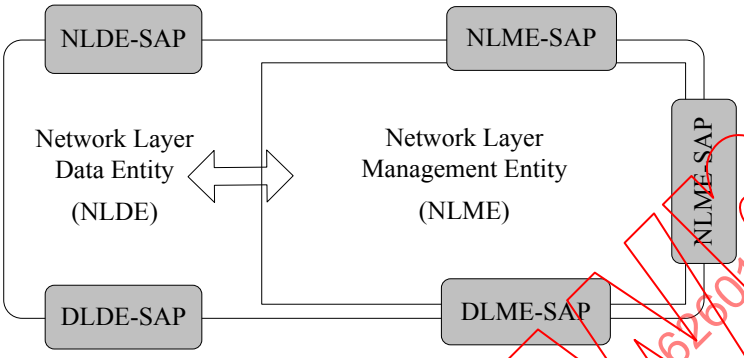

Figure 38 – WIA-PA Network layer reference model

The NL provides two kinds of services that are accessed through the following two SAPs:

a) The network layer data service, accessed through the NLDE SAP (NLDE-SAP);

b) The network layer management service, accessed through the NLME SAP (NLME-SAP).

9.3 Function description

9.3.1 General

The NL is designed to perform the following functions:

a) addressing,

b) routing,

c) communication resource allocation,

d) packet lifecycle management,

e) management for device joining and leaving network,

f) end-to-end network performance monitoring, and

g) fragmentation and reassembly.

9.3.2 Addressing

Each device has a global unique 64–bit “long address” and a 16-bit “short address”. Each device joins the network by using the long address and communicates with other devices by using the short address after joining the network. The 16-bit short address is indicated as x.y, where x and y are 8-bit integers. The most significant 8-bit in the 16-bit short address is the cluster address, and the least significant 8-bit is the intra-cluster address. The range of the cluster address is 0 to 255, and the range of the intra-cluster address is 0 to 255. The number 255 is used for the broadcast address. A routing device’s intra-cluster address is 0.

The classifications of the short address are as follows:

a) Unicast addresses: each device’s address is the unicast address, which includes the following types.

- The gateway device’s address is 0.y, where y is 0 to 254;
- The routing device’s address is x.0, where x is 1 to 254;

- The field device's address is x.y, where x is 1 to 254 and y is 1 to 254;

b) Broadcast address: according to the different broadcast domains, there are four types of broadcast addresses.

- The intra-cluster broadcast address is x.255, where x is 0 to 254;
- The global broadcast address is 255.255;
- The mesh network's broadcast address is 255.0;
- The gateway device's broadcast address is 0.255.

9.3.3 Routing

9.3.3.1 General

The WIA-PA network supports static routing algorithms configured by the NM. After getting the neighbor information from each routing device, the NM generates the connection relationship of all routing devices. On the basis of the connection relationship, the NM generates and writes the routing information to each routing device in the form of a routing table. Each route in the table is assigned a route ID. Multiple paths correspond to each VCR, including the main path and the redundant path. The main path and the redundant path use the same route ID.

9.3.3.2 Routing table

Each routing device maintains a routing table, which is generated by the NM. The routing table is used for the path selection in the mesh network. The table has five items. *RouteID* is the identifier of a route. *SourceAddress* is the address of the start point of a route, *DestinationAddress* is the end point address of a route, *NextHop* is the address of the next hop device in a route, and *RetryCounter* records the number of end-to-end retries in a route which reflects the status of the route. The *RouteID*s of paths with same source address and destination address are same.

An example of a routing table is shown in Table 66:

Table 66 – Example of a routing table

RouteID	SourceAddress	DestinationAddress	NextHop	RetryCounter
5	F1	N1	N3	0
8	F2	Ng	N2	0
…	…	…	…	…

NOTE Ng in the table indicates a gateway address; N1, N2 and N3 indicate routing devices' addresses; F1 and F2 indicate field devices' addresses.

9.3.4 Packet lifecycle management

Each packet has a lifecycle in the WIA-PA network. The lifecycle is expressed as a maximum surviving time. The NL records the generation time of packets by using timestamps. The surviving time is computed according to the generation time. When the surviving time of a packet exceeds its lifecycle, the packet should be discarded.

9.3.5 Joining and leaving network of device

The WIA-PA NL supports the joining and leaving processes of devices. The joining and leaving processes include the joining and leaving of field devices and of routing devices.

9.3.6 End-to-end network performance monitoring

The WIA-PA network layer monitors each path's health status. The NLME records the number of retries of each path to estimate the path failures. If there is a path failure, the NLME sends an indication to the DMAP (see 9.5.12).

9.3.7 Fragmentation and reassembly

Fragmentation and reassembly are handled at the NL. If the length of an NPDU is longer than the maximum DLSL payload, the NPDU should be fragmented at the NL of the sender. When the fragmented NPDUs reach the receiver, they are reassembled at the NL. See 9.6.2 for detailed packet format.

9.3.8 Network layer state machine

The NL defines the following states: Idle, Transmitting, and Receiving, as shown in Table 67.

Table 67 – Network layer states

State	Description
Idle	Do nothing, wait for events to occur
Transmitting	Pack the packets from ASL, and pass them to the DLSL
Receiving	Unpack the packets, and pass them to the ASL

State transitions are shown in Figure 39:

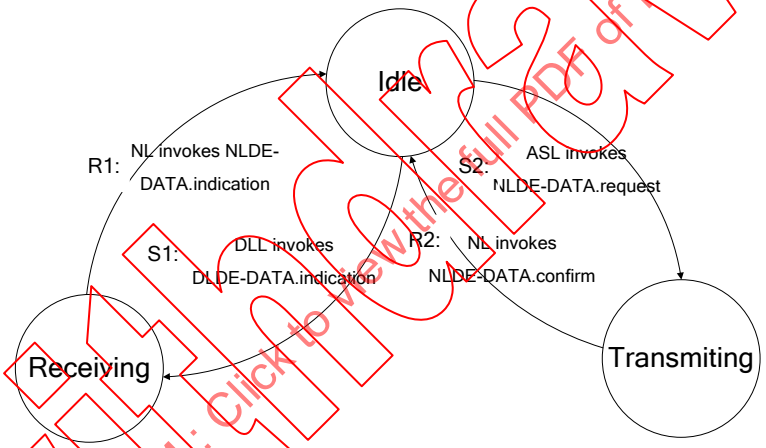

Figure 39 – Network layer state machine

Table 68– NL state transitions

Sequence number #	Current state	Event or conditions = > Actions	Next state
R1	Receive	NL invokes = > NLDE-DATA. indication()	Idle
R2	Transmit	NL invokes = > NLDE-DATA.confirm ()	Idle
S1	Idle	DLSL invokes = > DLDE-DATA. Indication()	Receive
S2	Idle	ASL invokes = > NLDE-DATA. Request()	Transmit

The states in the NL state machine are specified as follows:

a) Idle state

The following transitions may occur while in the “Idle” state:

- When the DLSL invokes DLDE-DATA.indication, the NL enters the “Receive” state.
- When the ASL invokes NLDE-DATA.request, the NL enters the “Transmit” state.

b) Transmit state

In the “Transmit” state, if the NL receives NLDE-DATA.confirm, it enters the “Idle” state.

c) Receive state

When in the “Receive” state, the NL enters the “Idle” state after the NL issues NLDE-DATA.indication to the upper layer.

9.4 Network layer data services

9.4.1 General

The NLDE-SAP is used by the ASL to receive and transmit data. The primitives supported by NL data services include NLDE-DATA.request, NLDE-DATA.confirm, and NLDE-DATA.indication.

9.4.2 NLDE-DATA.request

The NLDE receives the payload from the ASL through NLDE-DATA.request and adds it to the message queue of the NL.

The semantics of NLDE-DATA.request are described as follows:

```
NLDE-DATA.request (
                    VcrID,
                    SrcAddr,
                    Priority,
                    PayloadLength,
                    Payload,
                    PayloadHandle
                  )
```

Table 69 specifies the parameters for NLDE-DATA.request.

Table 69 – NLDE-DATA.request parameters

Name	Data type	Valid range	Description
VcrID	Unsigned16	0 to 65 535	The VCR identifier
SrcAddr	Unsigned16	0 to 65 535 (Unicast address)	The 16-bit short address of the NSDU’s source
Priority	Unsigned8	0 to 15	Priority of this NSDU
PayloadLength	Unsigned16	0 to 65 535	The length of NSDU to be transmitted
Payload	Octetstring		NSDU
PayloadHandle	Unsigned8	0 to 255	The handle associated with the NSDU to be transmitted

9.4.3 NLDE-DATA.confirm

NLDE-DATA.confirm reports the results of NLDE-DATA.request.

The semantics of NLDE-DATA.confirm are described as follows:

```
NLDE-DATA.confirm (
                    PayloadHandle,
                    Status
                  )
```

Table 70 specifies the parameters for NLDE-DATA.confirm.

Table 70 – NLDE-DATA.confirm parameters

Name	Data type	Valid range	Description
PayloadHandle	Unsigned8	0 to 255	Handle of the NSDU, which is used to indicate the NL payload.
Status	Unsigned8	0 to 255	Result of NLDE-DATA.request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.4.4 NLDE-DATA.indication

NLDE-DATA.indication informs the ASL when the NL receives a packet.

The semantics of NLDE-DATA.indication are described as follows:

```
NLDE-DATA.indication (
                        SrcAddr,
                        Priority,
                        NSDULength,
                        NSDU
                      )
```

Table 71 specifies the parameters of NLDE-DATA.indication.

Table 71 – NLDE-DATA.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535 (Unicast Address)	The 16-bit short address of the NSDU's source
Priority	Unsigned8	0 to 15	Priority of this NSDU
NSDULength	Unsigned16	0 to 65 535	Length of the NSDU
NSDU	octet		Data of the NSDU

9.4.5 Time sequence of NL data services

Figure 40 shows the basic procedures of packet sending and receiving. NLDE-DATA.request is generated by a local Application Sub-Layer Data Entity (ASLDE) when a data Application Sub-Layer Protocol Data Unit (ASLPDU) is to be transferred to a peer NLDE. On receipt of a NLDE-DATA.request, the NLDE begins transmitting of the NPDU.

NLDE-DATA.confirm is generated by the NLDE of a source device in response to NLDE-DATA.request. NLDE-DATA.confirm returns a status indicating the result of the transmission.

NLDE-DATA.indication is generated by the NLDE of a destination device and is issued to the ASLDE on receipt of a data packet at the local NLDE.

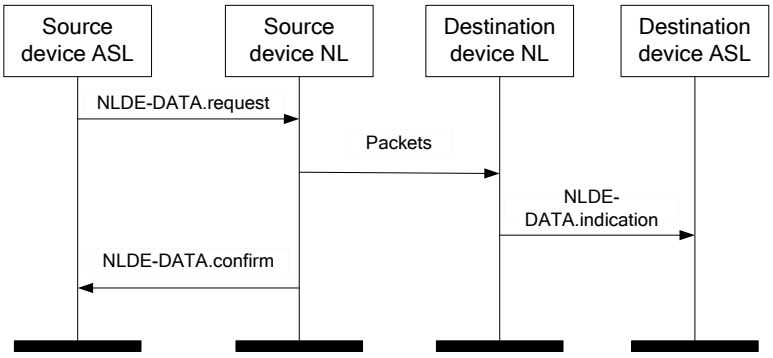

Figure 40 – Time sequence of NL data services

9.5 Network layer management services

9.5.1 General

The DMAP uses the interface supplied by NLME-SAP to configure and control the NL operation.

9.5.2 Network communication status report services

9.5.2.1 NLME-COMM-STATUS.request

The semantics of NLME-COMM-STATUS.request are described as follows:

```
NLME-COMM-STATUS.request (
                          ProxyAddr,
                          PhyAddr,
                          DeviceType,
                          Status
                          )
```

Table 72 specifies the parameters for NLME-COMM-STATUS.request.

Table 72 – NLME-COMM-STATUS.request parameters

Name	Data type	Valid range	Description
ProxyAddr	Unsigned16	0 to 65 535	The address of the routing device selected by the new device (unicast address)
PhyAddr	Unsigned64	0 to (2^{64}-1)	64-bit physical address of the new device
DeviceType	Unsigned8	0 to 255	The type of the new device: 0 = Gateway device; 1 = Routing device; 2 = Field device; 3 = Handheld device; Others are reserved.
Status	Unsigned8	0 to 255	Results of the joining process of a new device: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.5.2.2 NLME-COMM-STATUS.indication

The semantics of NLME-COMM-STATUS.indication are described as follows:

```
NLME-COMM-STATUS.indication (
                                    ProxyAddr,
                                    PhyAddr,
                                    DeviceType,
                                    Status
                                )
```

Table 73 specifies the parameters for NLME-COMM-STATUS.indication.

Table 73 – NLME-COMM-STATUS.indication parameters

Name	Data type	Valid range	Description
ProxyAddr	Unsigned16	0 to 65 535	The address of the routing device selected by the new device (unicast address)
PhyAddr	Unsigned64	0 to (2^{64}-1)	64-bit physical address of the new device
DeviceType	Unsigned8	0 to 255	The type of the new device: 0 = Gateway device; 1 = Routing device; 2 = Field device; 3 = Handheld device; Others are reserved.
Status	Unsigned8	0 to 255	Results of the joining process of a new device: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

On receipt of NL communication status request packet, the NL of the gateway device will invoke NLME-COMM-STATUS.indication and inform the DMAP.

9.5.2.3 NLME-COMM-STATUS.confirm

The semantics of NLME-COMM-STATUS.confirm are described as follows:

```
NLME-COMM-STATUS.confirm (
                                    PhyAddr,
                                    Status
                                )
```

Table 74 specifies the parameters for NLME-COMM-STATUS.confirm.

Table 74 – NLME-COMM-STATUS.confirm parameters

Name	Data type	Valid range	Description
PhyAddr	Unsigned64	0 to (2^{64}-1)	64-bit physical address of the new device
Status	Unsigned8	0 to 255	Execution result of the request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

NLME-COMM-STATUS.confirm is generated by the NL in response to NLME-COMM-STATUS.request. NLME-COMM-STATUS.confirm returns a status of either SUCCESS, indicating that the requested transmission has been successful, or FAILURE, indicating that the transmission has failed.

9.5.3 Network joining services

9.5.3.1 NLME-JOIN.request

The semantics of NLME-JOIN.request are described as follows:

```
NLME-JOIN.request (
                    ProxyAddr,
                    PhyAddr,
                    SecMaterial,
                    DeviceType
                  )
```

Table 75 specifies the parameters for NLME-JOIN.request.

Table 75 – NLME-JOIN.request parameters

Name	Data type	Valid range	Description
ProxyAddr	Unsigned16	0 to 65 535	The address of the routing device selected by the new device (Unicast address)
PhyAddr	Unsigned64	0 to (2^{64}-1)	64-bit physical address of the new device
SecMaterial	Unsigned32	0 to (2^{32}-1)	The identity information of a device
DeviceType	Unsigned8	0 to 255	The type of the new device: 0 = Gateway device; 1 = Routing device; 2 = Field device; 3 = Handheld device; Others are reserved.

9.5.3.2 NLME-JOIN.indication

The semantics of NLME-JOIN.indication are described as follows:

```
NLME-JOIN.indication (
                       ProxyAddr,
                       PhyAddr,
                       SecMaterial,
                       DeviceType
                     )
```

Table 76 specifies the parameters for NLME-JOIN.indication.

Table 76 – NLME-JOIN.indication parameters

Name	Data type	Valid range	Description
ProxyAddr	Unsigned16	0 to 65 535	Short address of the routing device that generates NL join request
PhyAddr	Unsigned64	0 to (2^{64}-1)	Device's physical address
SecMaterial	Unsigned32	0 to (2^{32}-1)	The identity information of a device
DeviceType	Unsigned8	0 to 255	The type of the new device: 0 = Gateway device; 1 = Routing device; 2 = Field device; 3 = Handheld device; Others are reserved.

On receipt of NL joining request packet, the NL of the gateway device will invoke NLME-JOIN.indication and inform the DMAP.

9.5.3.3 NLME-JOIN.response

The semantics of NLME-JOIN.response are described as follows:

```
NLME-JOIN.response (
                    ProxyAddr,
                    PhyAddr,
                    ShortAddr,
                    Status
                  )
```

Table 77 specifies the parameters for NLME-JOIN.response.

Table 77 – NLME-JOIN.response parameters

Name	Data type	Valid range	Description
ProxyAddr	Unsigned16	0 to 65 535	Short address of the routing device that generates NL join request
PhyAddr	Unsigned64	0 to (2^{64}-1)	Physical address of new device
ShortAddr	Unsigned16	0 to 65 535	Allocated network address (Unicast address) of new device
Status	Unsigned8	0 to 255	Execution result of the request 0 = SUCCESS; 1 = FAILURE; Others are reserved.

The DMAP invokes NLME-JOIN.response to assign a network address to the new device.

9.5.3.4 NLME-JOIN.confirm

The semantics of NLME-JOIN.confirm are described as follows:

```
NLME-JOIN.confirm (
                    ShortAddr,
                    Status
                  )
```

Table 78 specifies the parameters for NLME-JOIN.confirm.

Table 78 – NLME-JOIN.confirm parameters

Name	Data type	Valid range	Description
ShortAddr	Unsigned16	0 to 65 535	Network address (Unicast address)
Status	Unsigned8	0 to 255	Execution result of the request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

NLME-JOIN.confirm is generated by the NL in response to NLME-JOIN.request. NLME-JOIN.confirm returns a status of either SUCCESS, indicating that the requested transmission has been successful, or FAILURE, indicating that the transmission has failed.

9.5.3.5 Time sequence for device joining

9.5.3.5.1 The time sequence for field device joining

To join the star network, a field device should initiate a joining request in the MAC layer by DMAP to the routing device or the gateway device. A field device should join the network either through the gateway device or through a routing device.

When the gateway device receives a joining request from a field device, it should indicate the joining to the NM and then return a joining response. If the new field device receives the MAC associate response from the gateway device, it joins the network.

When the routing device receives a joining request from a field device, it should produce a joining request of the NL and send this request to the gateway device. If the new field device receives the joining response, the joining process is finished.

Figure 41 illustrates an example of the joining process of a field device in the hierarchical network that is the combination of star and mesh.

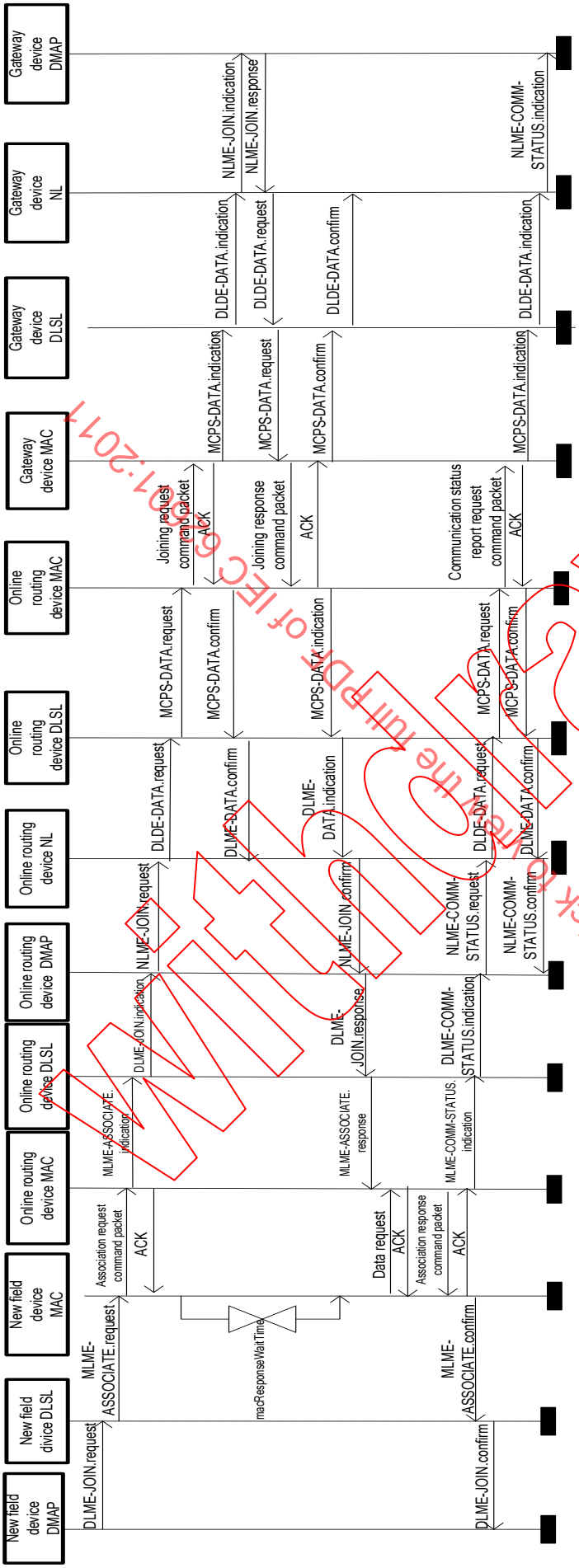

Figure 41 – Time sequence for field device joining through routing device

9.5.3.5.2 The time sequence for routing device joining

If a routing device wants to join the mesh network, it should send a joining request to the gateway device, and the request includes the long address of this routing device. There are two kinds of joining processes for routing devices: one-hop joining and multi-hop joining. If the new routing device sends a joining request to the gateway device, the one-hop joining process is used; if the new routing device joins the mesh network through another existing routing device, the multi-hop joining process is used.

The gateway device receives the joining request and informs the NM. The NM should return a response. If the new routing device receives the response, its joining process is completed.

Figure 42 and Figure 43 illustrate examples of the joining process of routing devices.

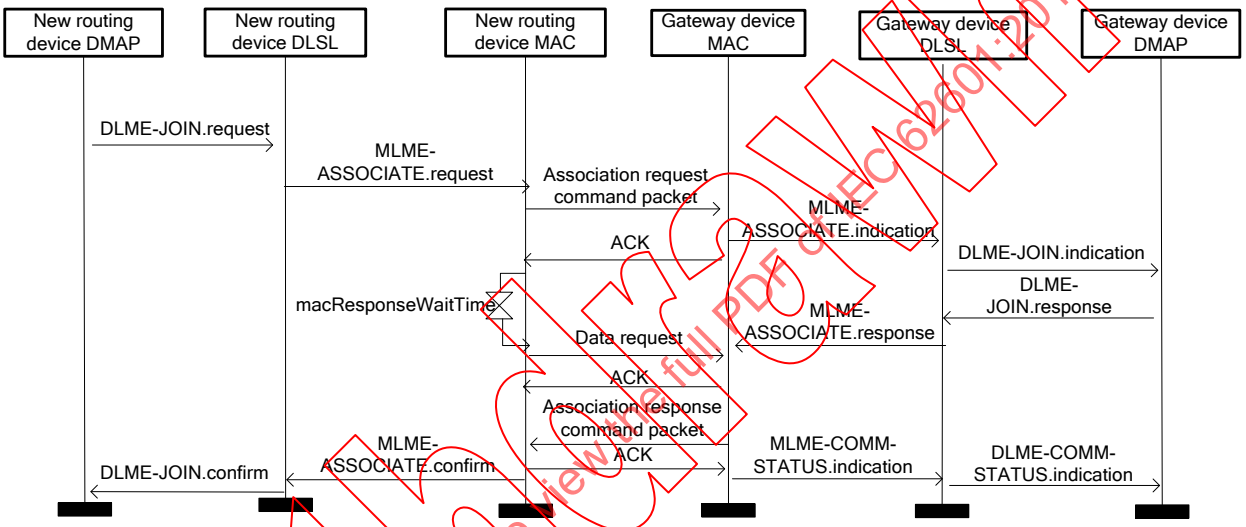

Figure 42 – One-hop joining process for routing device

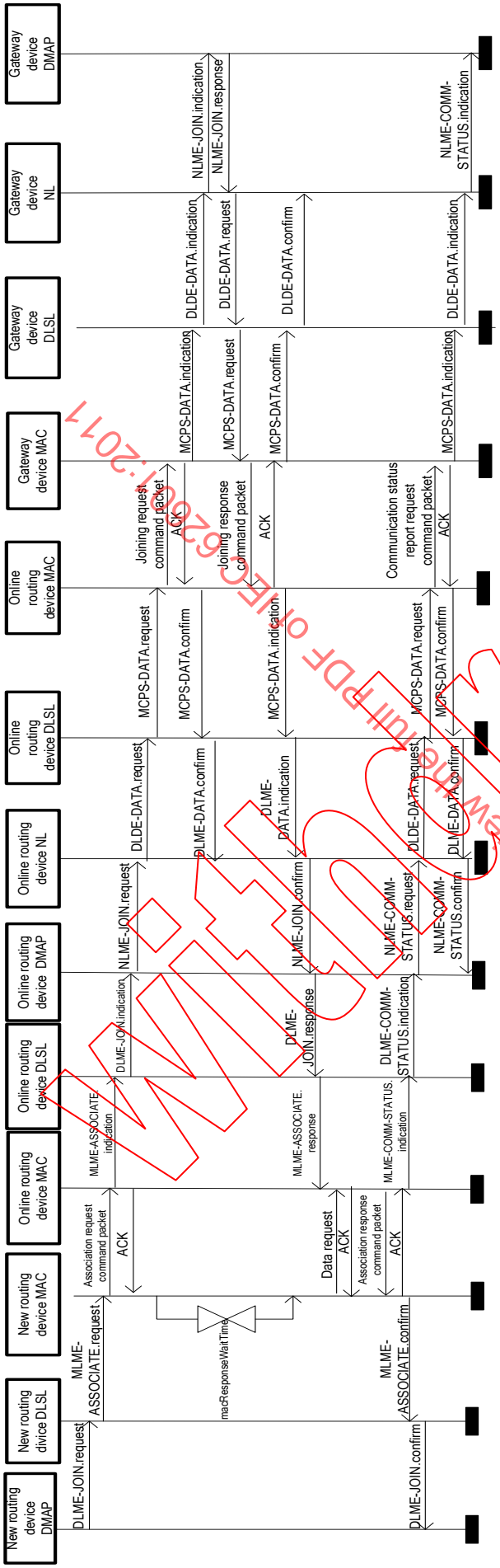

Figure 43 – Multi-hop join process of routing device

NOTE Before joining the network, the field devices and the routing devices set the "DeviceState" (see Table 19) to 0 to indicate that the devices have not joined the network. If the new devices have sent out the joining request and have not received the joining response, they set the "DeviceState" to 1 to indicate that the devices are joining the network. If the new devices have sent out the joining request and have received the joining response successfully, they set the "DeviceState" to 3 to indicate that the devices have joined the network. If the joining process needs security authentication, the devices set the "DeviceState" to 3 during the authentication process. See Table 19 for "DeviceState".

9.5.4 Network leaving services

9.5.4.1 NLME-LEAVE.request

The semantics of NLME-LEAVE.request are described as follows:

```
NLME-LEAVE.request (
                        SrcAddr,
                        ShortAddr
                    )
```

Table 79 specifies the parameters for NLME-LEAVE.request.

Table 79 – NLME-LEAVE.request parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	Source device's address
ShortAddr	Unsigned16	0 to 65 535	Network address (unicast address)

When a device wants to leave the network, the DMAP invokes NLME-LEAVE.request and requests its NL to start the leaving process.

9.5.4.2 NLME-LEAVE.indication

The semantics of NLME-LEAVE.indication are described as follows:

```
NLME-LEAVE.indication (
                        ShortAddr
                    )
```

Table 80 specifies the parameters for NLME-LEAVE.indication.

Table 80 – NLME-LEAVE.indication parameters

Name	Data type	Valid range	Description
ShortAddr	Unsigned16	0 to 65 535	Network address (unicast address)

On receipt of a leaving request command packet, the NL will invoke NLME-LEAVE.indication and inform the DMAP.

9.5.4.3 NLME-LEAVE.response

The semantics of NLME- LEAVE.response are described as follows:

```
NLME-LEAVE.response (
                        ShortAddr,
                        Status
                    )
```

Table 81 specifies the parameters for NLME-LEAVE.response.

Table 81 – NLME-LEAVE.response parameters

Name	Data type	Valid range	Description
ShortAddr	Unsigned16	0 to 65 535	Network address of the device
Status	Unsigned8	0 to 255	Execution result of the request 0 = SUCCESS; 1 = FAILURE; Others are reserved.

NLME-LEAVE.response is used by DMAP or NM to inform the leaving device whether or not the leaving request has been accepted.

9.5.4.4 NLME-LEAVE.confirm

The semantics of NLME-LEAVE.confirm are described as follows:

```
NLME-LEAVE.confirm (
                        ShortAddr,
                        Status
                    )
```

Table 82 specifies the parameters for NLME-LEAVE.confirm.

Table 82 – NLME-LEAVE.confirm parameters

Name	Data type	Valid range	Description
ShortAddr	Unsigned16	0 to 65 535	The network address(Unicast address)
Status	Unsigned8	0 to 255	Execution result of the request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

NLME-LEAVE.confirm is generated by the NL in response to NLME-LEAVE.request. NLME-LEAVE.confirm returns a status of either "*SUCCESS*", indicating that the request to transmit is successful, or "*FAILURE*", indicating that the request to transmit is failed.

9.5.4.5 Time sequence for device leaving

9.5.4.5.1 The time sequence for field device leaving

The leaving processes of field devices include the following two types.

a) Active leaving. For a field device, the DMAP invokes DLME-LEAVE.request to send a leaving request to the routing device or the gateway device that is its cluster head. After receiving the leaving request command packet from the field device, the NL of the routing device or the gateway device informs the DMAP with NLME-LEAVE.indication. If the field device leaves the routing device, the routing device sends NLME-RPT-CLRMEM.request to the gateway device. Figure 44 illustrates the time sequence of a field device's active leaving.

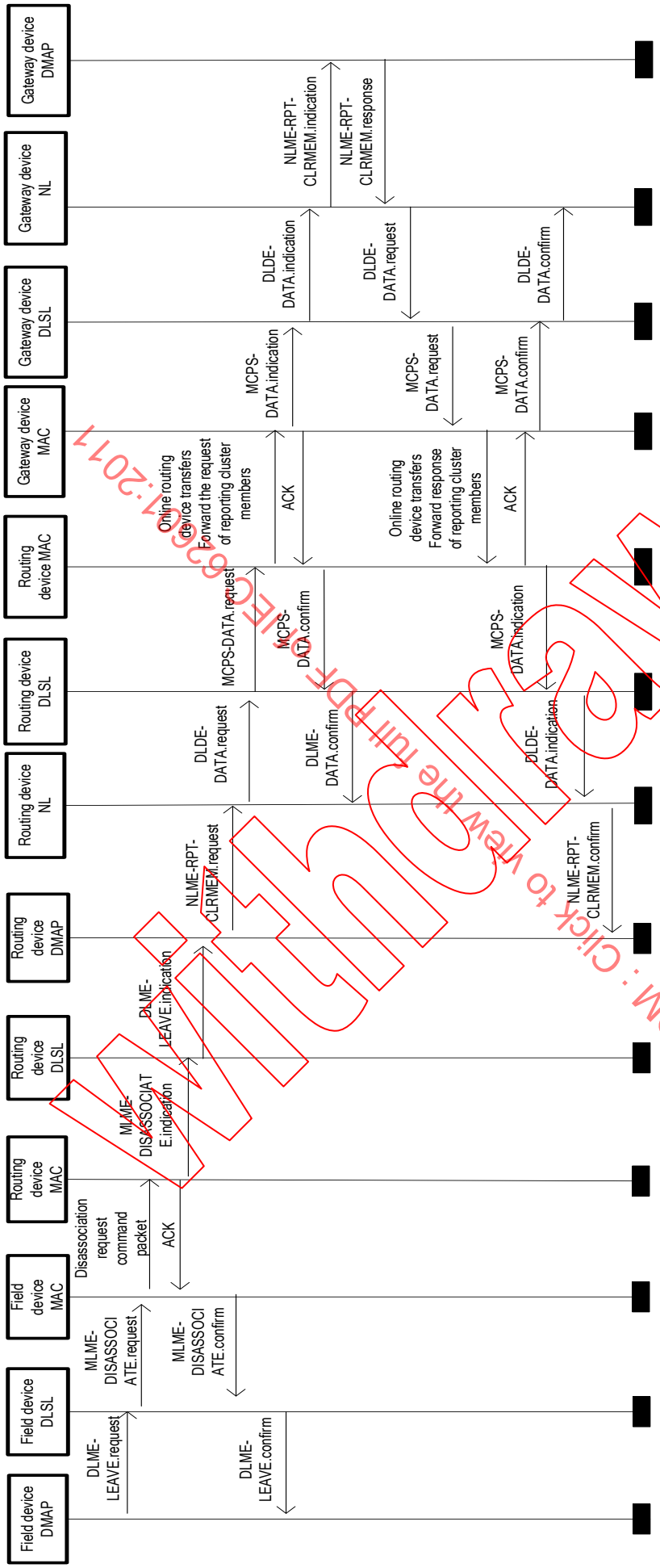

Figure 44 – Active leaving process of field device (leaving routing device)

b) Passive leaving. The DMAP of a routing device requests for its field device to leave the network by DLME-LEAVE.request. After receiving this request, the field device returns Acknowledge (ACK). If the routing device receives DLME-LEAVE.confirm, it sends a cluster member report request to the gateway device. Figure 45 illustrates the time sequence of a field device passive leaving.

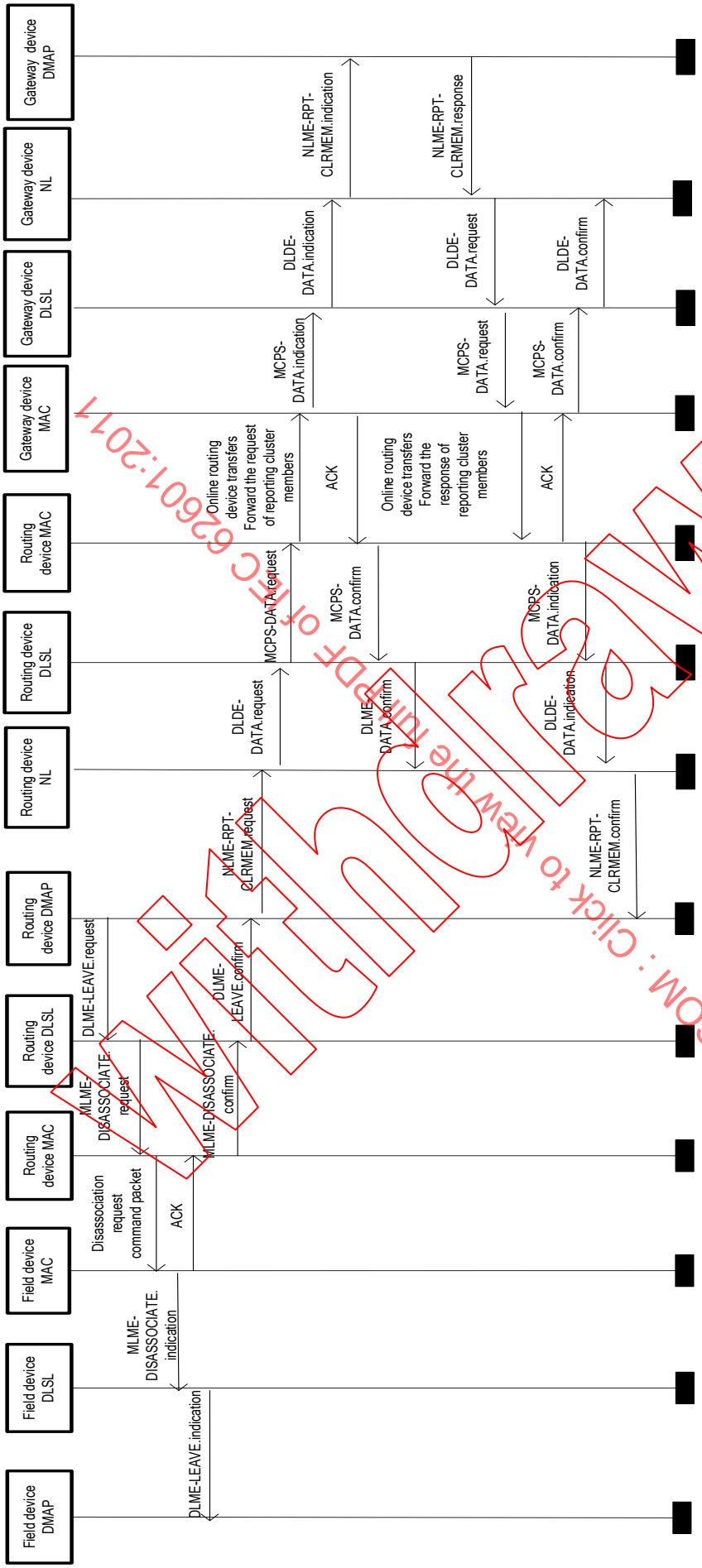

Figure 45 – Passive leaving of field device

Routing devices connect to both the mesh network and the star network. Therefore, the routing device should inform both the gateway device and its field devices of its leaving.

9.5.4.5.2 The time sequence for routing device leaving

The leaving processes of routing devices include the following two types.

a) Active leaving. The DMAP of a routing device sends its leaving request to the gateway device by NLME-LEAVE.request. After receiving the leaving request, the gateway device returns a leaving response. When the DMAP of the leaving routing device receives NLME-LEAVE.confirm, it informs its cluster members. Figure 46 illustrates the time sequence of a routing device's active leaving.

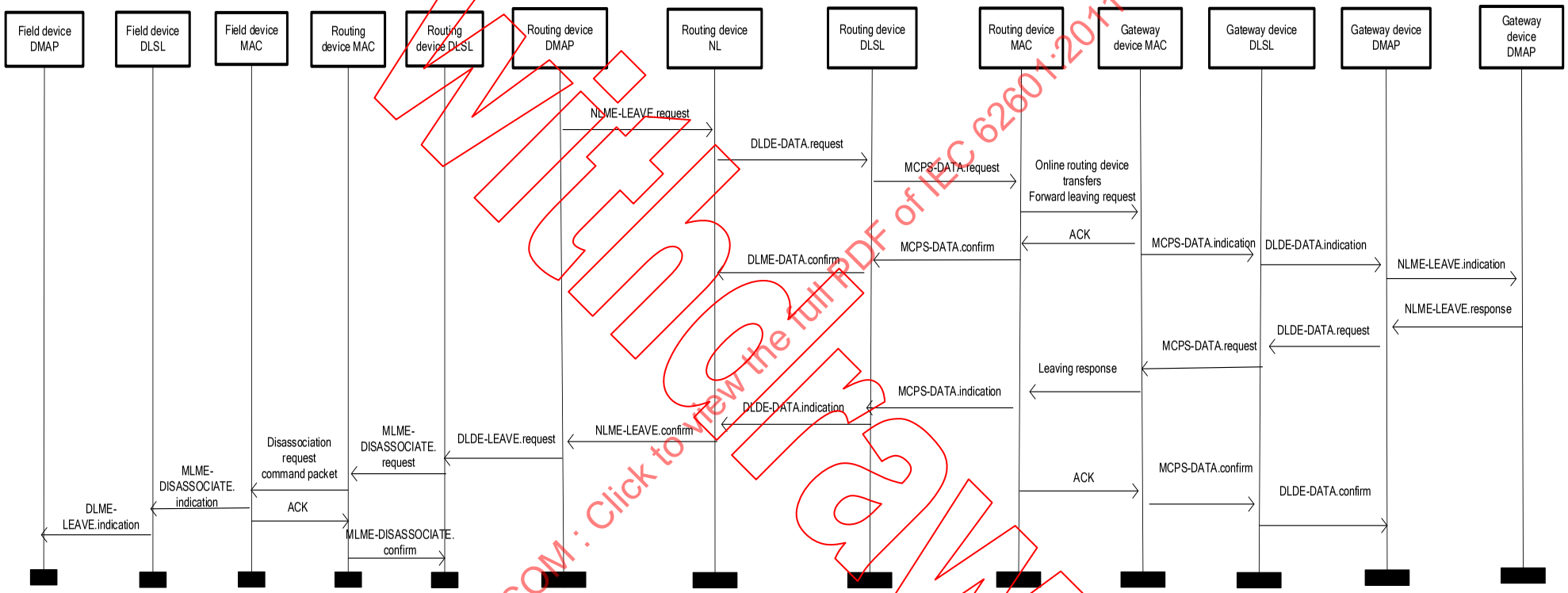

Figure 46 – Active leaving process of routing device

b) Passive leaving. If the gateway device wants a routing device to leave the network, it produces an NL leaving request packet, and sends it to the designated routing device. After receiving the leaving request packet from the gateway device, the routing device informs all its cluster members of its leaving. Figure 47 illustrates the time sequence of a routing device's passive leaving.

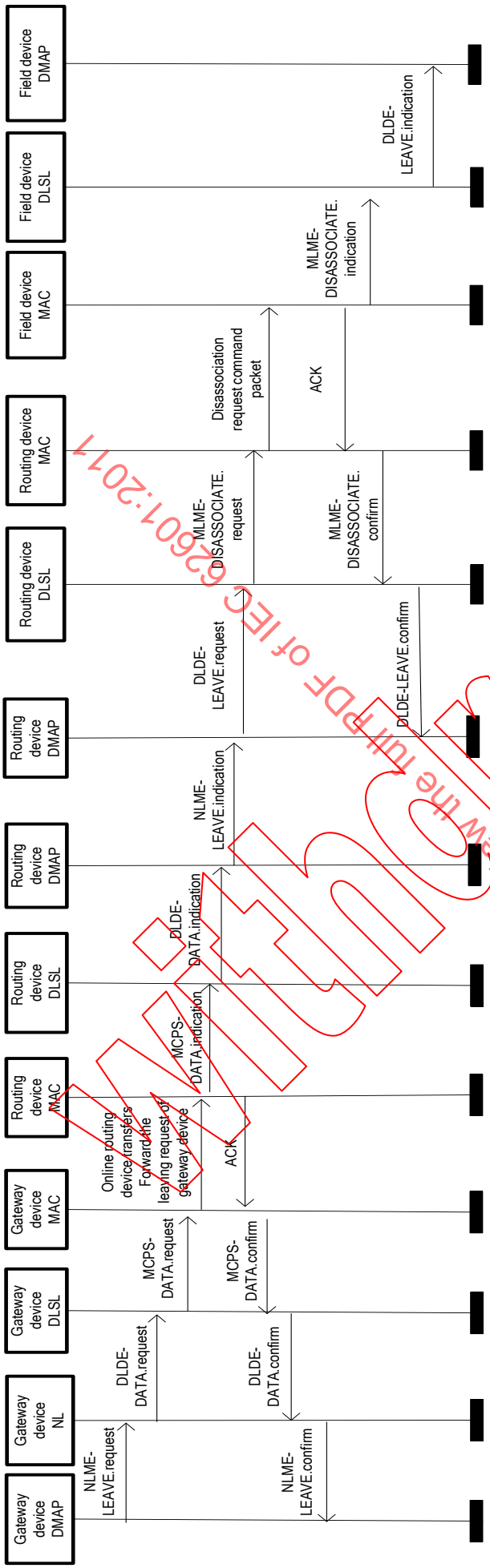

Figure 47 – Passive leaving process of routing device

9.5.5 Cluster member report services

9.5.5.1 NLME-RPT-CLRMEM.request

NLME-RPT-CLRMEM.request is used by routing devices to report the information of the cluster members to the gateway device.

The semantics of NLME-RPT-CLRMEM.request are described as follows:

```
NLME-RPT-CLRMEM.request (
                          SrcAddr,
                          DstAddr,
                          ClrMemFlag,
                          ClrMemAddr
                        )
```

Table 83 specifies the parameters for NLME-RPT-CLRMEM.request.

Table 83 – NLME-RPT-CLRMEM.request parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	The source address
DstAddr	Unsigned16	0 to 65 535	The destination address
ClrMemFlag	Unsigned8	0 to 255	The flag of the cluster member modification: 0 = Add; 1 = Delete; Others are reserved.
ClrMemAddr	Unsigned16	0 to 65 535	The network address of the modified cluster member

9.5.5.2 NLME-RPT-CLRMEM.confirm

NLME-RPT-CLRMEM.confirm is used by the NL to return the result of NLME-RPT-CLRMEM.request to the DMAP..

The semantics of NLME-RPT-CLRMEM.confirm are described as follows:

```
NLME-RPT-CLRMEM.confirm (
                          Status
                        )
```

Table 84 specifies the parameters for NLME-RPT-CLRMEM.confirm.

Table 84 – NLME-RPT-CLRMEM.confirm parameter

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	The result of the cluster member report: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.5.5.3 NLME-RPT-CLRMEM.indication

NLME-RPT-CLRMEM.indication is used by the NL to report the successful receipt of cluster member report request packets.

The semantics of NLME-RPT-CLRMEM.indication are described as follows:

NLME-RPT-CLRMEM.indication (
SrcAddr,
ClrMemFlag,
ClrMemAddr
)

Table 83 specifies the parameters for NLME-RPT-CLRMEM.indication.

9.5.5.4 NLME-RPT-CLRMEM.response

NLME-RPT-CLRMEM.response is used to respond to NLME-RPT-CLRMEM.indication.

The semantics of NLME-RPT-CLRMEM.response are described as follows:

NLME-RPT-CLRMEM.response (
DstAddr,
Status
)

Table 85 specifies the parameters for NLME-RPT-CLRMEM.response.

Table 85 – NLME-RPT-CLRMEM.response parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	The destination address
Status	Unsigned8	0 to 255	The result returned from cluster member report: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.5.5.5 Time sequence for cluster member reporting

The time sequence diagram for reporting cluster member is shown in Figure 48.

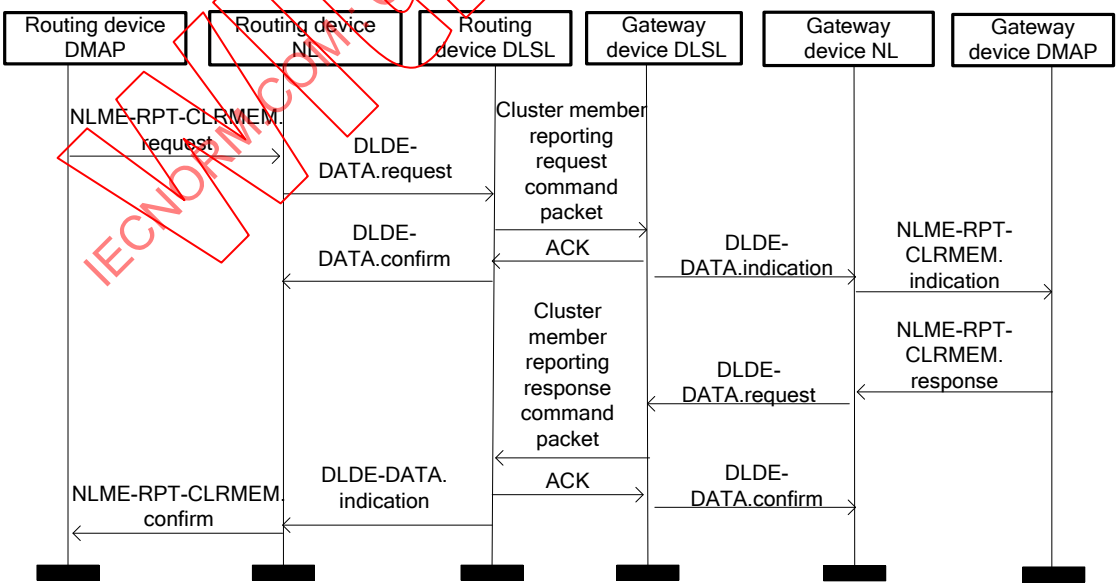

Figure 48 – Cluster member reporting process

9.5.6 Neighbor information report services

9.5.6.1 NLME-NEIGHBOR-INFO.request

NLME-NEIGHBOR-INFO.request is used for a routing device to report its one-hop neighbors' information to the gateway device.

The semantics of NLME-NEIGHBOR-INFO.request are described as follows:

```
NLME-NEIGHBOR-INFO.request (
                                    SrcAddr,
                                    DstAddr,
                                    NeighborCount,
                                    NeighborStructure
                                )
```

Table 86 specifies the parameters for NLME-NEIGHBOR-INFO.request.

Table 86 – NLME-NEIGHBOR-INFO.request parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	16-bit source address
DstAddr	Unsigned16	0 to 65 535	16-bit destination address
NeighborCount	Unsigned16	0 to 65 535	Number of the neighbors
NeighborStructure	Neighbor_Struct structure (see Table 17)		Information of the neighbors

9.5.6.2 NLME-NEIGHBOR-INFO.indication

NLME-NEIGHBOR-INFO.indication is used for the DLSL to report the received NLME-NEIGHBOR-INFO.request packet to the DMAP.

The semantics of NLME-NEIGHBOR-INFO.indication are described as follows:

```
NLME-NEIGHBOR-INFO.indication (
                                    SrcAddr,
                                    NeighborCount,
                                    NeighborStructure
                                )
```

Table 86 specifies the parameters for NLME-NEIGHBOR-INFO.indication.

9.5.6.3 NLME-NEIGHBOR-INFO.confirm

NLME-NEIGHBOR-INFO.confirm is used to report the successful sending of neighbor information report request packets.

The semantics of NLME-NEIGHBOR-INFO.confirm are described as follows:

```
NLME-NEIGHBOR-INFO.confirm (
                                    Status
                                )
```

Table 87 specifies the parameters for NLME-NEIGHBOR-INFO.confirm.

Table 87 – NLME-NEIGHBOR-INFO.confirm parameter

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Reporting the result of the neighbor information: 0 = SUCCESS; 1 = .FAILURE; Others are reserved.

9.5.6.4 Time sequence for neighbor information reporting

The time sequence diagram for reporting neighbor information is shown in Figure 49.

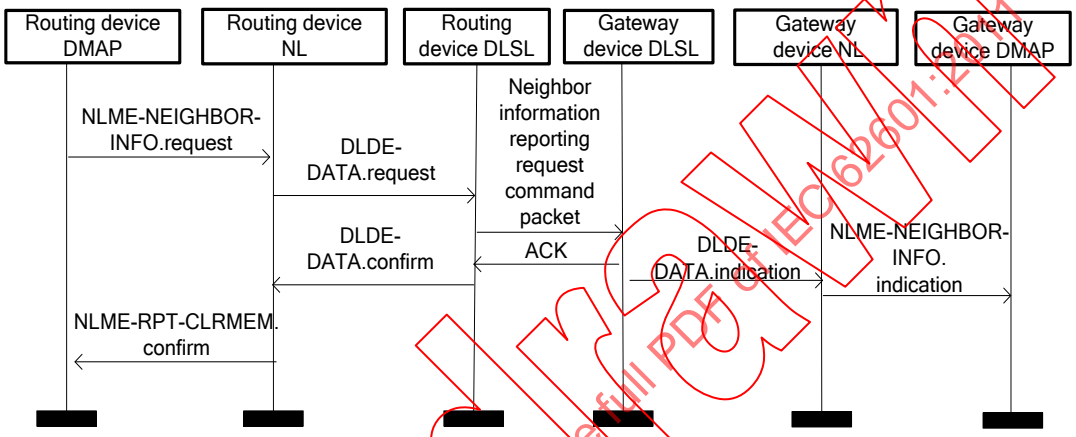

Figure 49 – Neighbor information reporting process

9.5.7 Route allocation services

9.5.7.1 Route adding services

9.5.7.1.1 NLME-ADD_ROUTE.request

NLME-ADD_ROUTE.request is used to add a record to the routing table of a routing device.

The semantics of NLME-ADD_ROUTE.request are described as follows:

```
NLME-ADD_ROUTE.request (
                        DstAddr,
                        RoutingTableRecord
                       )
```

Table 88 specifies the parameters for NLME-ADD_ROUTE.request.

Table 88 – NLME-ADD_ROUTE.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	The 16-bit short address of the routing device
RoutingTableRecord	NLRoute_Struct structure (see Table 14)		A routing table item

9.5.7.1.2 NLME-ADD_ROUTE.confirm

NLME-ADD_ROUTE.confirm reports the result of a NLME-ADD_ROUTE.request.

The semantics of NLME_ADD-ROUTE.confirm are described as follows:

NLME-ADD_ROUTE.confirm (
Status
)

Table 89 specifies the parameters for NLME-ADD_ROUTE.confirm.

Table 89 – NLME-ADD_ROUTE.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	The result of a NLME-ADD_ROUTE.request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

On receipt of NLME-ADD_ROUTE.request, the NL should transmit an adding-route-request packet to the routing device and should return an NLME-ADD_ROUTE.confirm to report the result.

9.5.7.1.3 NLME-ADD_ROUTE.indication

NLME-ADD_ROUTE.indication is used to report to the DMAP that a device has successfully received a route adding request packet.

The semantics of NLME_ADD-ROUTE.confirm are described as follows:

NLME-ADD_ROUTE.indication (
RoutingTableRecord
)

Table 88 specifies the parameters for NLME-ADD_ROUTE.indication.

9.5.7.1.4 NLME-ADD_ROUTE.response

NLME-ADD_ROUTE.response is the response of NLME-ADD_ROUTE.indication.

The semantics of NLME-ADD_ROUTE.response are described as follows:

NLME-ADD_ROUTE.response (
Status
)

Table 89 specifies the parameters for NLME-ADD_ROUTE.response.

9.5.7.1.5 Time sequence for route adding

The time sequence diagram for adding a record to the routing table of a routing device is shown in Figure 50.

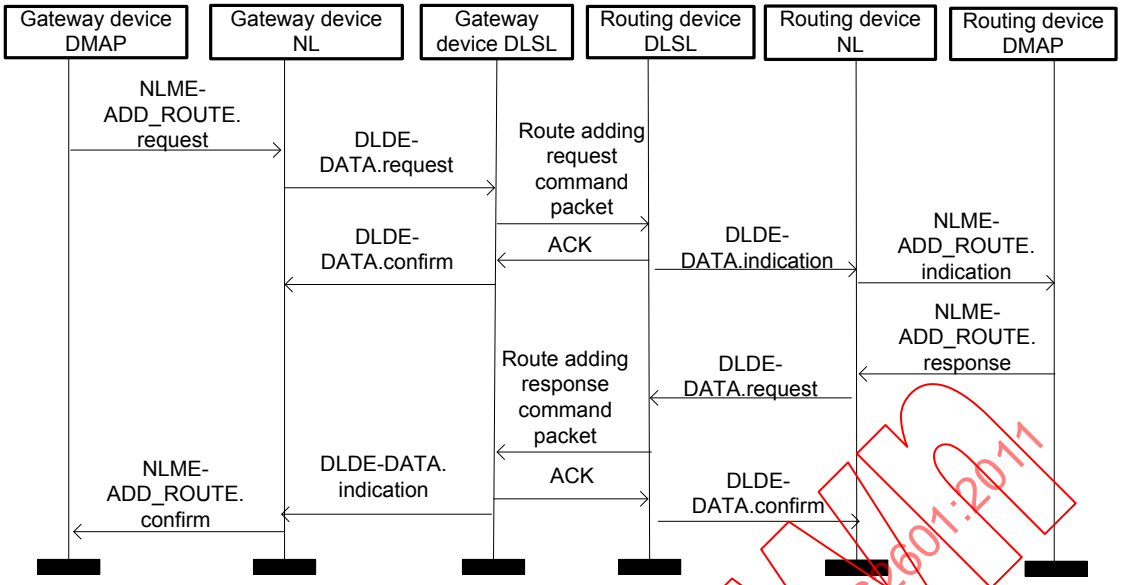

Figure 50 – Time sequence for route adding

9.5.7.2 Route update services

9.5.7.2.1 NLME-UPDATE_ROUTE.request

NLME-UPDATE_ROUTE.request updates a record in the routing table of a routing device.

The semantics of NLME-UPDATE_ROUTE.request are described as follows:

```
NLME-UPDATE_ROUTE.request (
                    DstAddr,
                    RoutingTableRecord
                    )
```

Table 90 specifies the parameters for NLME-UPDATE_ROUTE.request.

Table 90 – NLME-UPDATE_ROUTE.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	16-bit short address of routing device
RoutingTableRecord	NLRoute_Struct structure (see Table 14)		A routing table item

The NM invokes NLME-UPDATE_ROUTE.request to update a record in the routing table of routing devices.

9.5.7.2.2 NLME-UPDATE_ROUTE.confirm

The NLME_UPDATE-ROUTE.confirm reports the execution result of an NLME-UPDATE_ROUTE.request.

The semantics of NLME_UPDATE-ROUTE.confirm are described as follows:

```
NLME-UPDATE_ROUTE.confirm (
                    Status
                    )
```

Table 91 specifies the parameters for NLME-UPDATE-ROUTE.confirm.

Table 91 – NLME-UPDATE_ROUTE.confirm parameter

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	The result of NLME-UPDATE_ROUTE.request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

On receipt of NLME-UPDATE_ROUTE.request, the NL will transmit a route update request packet to the routing device and return NLME-UPDATE_ROUTE.confirm to report the result.

9.5.7.2.3 NLME-UPDATE_ROUTE.indication

NLME-UPDATE_ROUTE.indication is used to report to the DMAP that a device has successfully received a route update request packet.

The semantics of NLME-UPDATE_ROUTE.confirm are described as follows:

NLME- UPDATE_ROUTE.indication (
RoutingTableRecord
)

Table 90 specifies the parameters for NLME- UPDATE _ROUTE.indication.

9.5.7.2.4 NLME-UPDATE _ROUTE.response

NLME-UPDATE_ROUTE.response is the response of NLME- UPDATE_ROUTE.indication.

The semantics of NLME-UPDATE_ROUTE.response are described as follows:

NLME-UPDATE_ROUTE.response (
Status
)

Table 91 specifies the parameters for NLME-UPDATE_ROUTE.response.

9.5.7.2.5 Time sequence for route updating

The time sequence diagram for updating a record in the routing table of a routing device is shown in Figure 51.

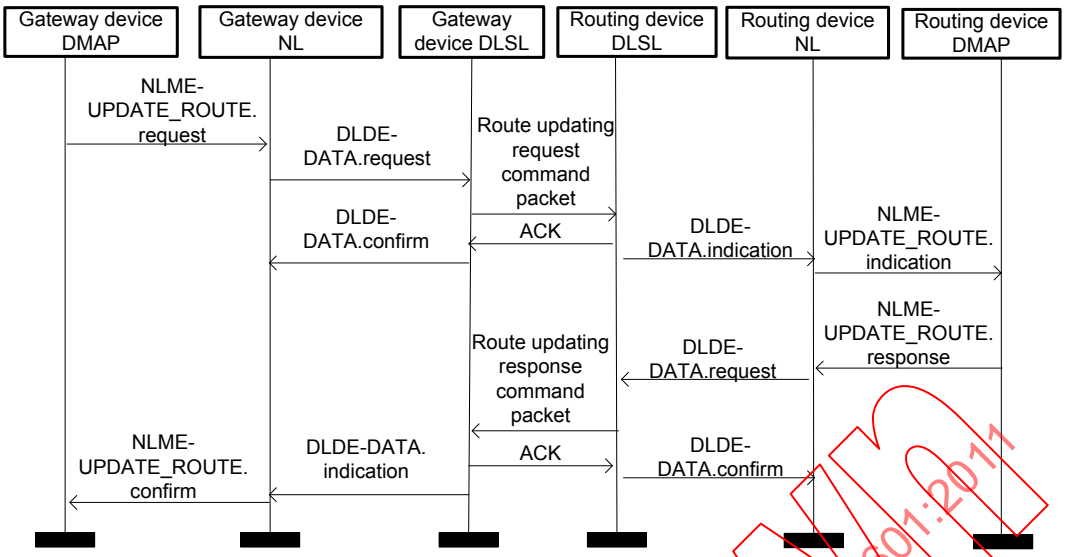

Figure 51 – Time sequence for route updating

9.5.7.3 Route deleting services

9.5.7.3.1 NLME-DELETE_ROUTE.request

The NM invokes NLME-DELETE_ROUTE.request to delete records in the routing table of a routing device.

The semantics of NLME_DELETE-ROUTE.confirm are described as follows:

```
NLME-DELETE_ROUTE.request (
                            DstAddr,
                            RouteID
                          )
```

Table 92 specifies the parameters for NLME-DELETE-ROUTE.request.

Table 92 – NLME-UPDATE_ROUTE.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	The 16-bit short address of the destination routing device
RouteID	Unsigned16	0 to 65 535	Route ID

9.5.7.3.2 NLME-DELETE_ROUTE.confirm

The NLME_DELETE-ROUTE.confirm reports the result of executing an NLME-DELETE_ROUTE.request.

The semantics of NLME_DELETE-ROUTE.confirm are described as follows:

```
NLME-DELETE_ROUTE.confirm (
                            Status
                          )
```

Table 93 specifies the parameters for NLME-DELETE_ROUTE.confirm.

Table 93 – NLME-DELETE_ROUTE.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	The result of a NLME-DELETE_ROUTE.request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

On receipt of NLME-DELETE_ROUTE.request, the NL should transmit a route deleting request packet to the routing device and return a NLME-DELETE_ROUTE.confirm to report the result.

9.5.7.3.3 NLME-DELETE_ROUTE.indication

NLME-DELETE_ROUTE.indication is used to report to the DMAP that the device has successfully received a route deleting request packet.

The semantics of NLME- DELETE_ROUTE.confirm are described as follows:

NLME-DELETE_ROUTE.indication (
RouteID
)

Table 92 specifies the parameters for NLME-DELETE_ROUTE.indication.

9.5.7.3.4 NLME-DELETE_ROUTE.response

The NLME-NLME-DELETE_ROUTE.response is the response of NLME-DELETE _ROUTE.indication.

The semantics of NLME-DELETE_ROUTE.response are described as follows:

NLME-DELETE_ROUTE response (
Status
)

Table 93 specifies the parameters for NLME-DELETE_ROUTE.response.

9.5.7.3.5 Time sequence for route deleting

The time sequence diagram for deleting records in the routing tables of routing devices is shown in Figure 52.

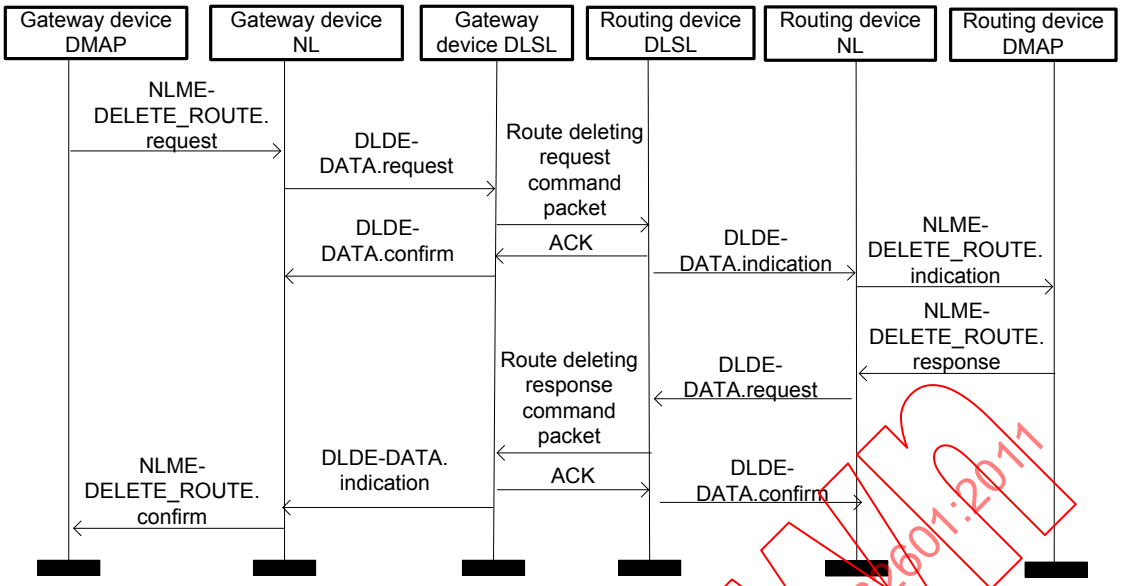

Figure 52 – Time sequence for route deleting

9.5.8 Communication resource allocation services

9.5.8.1 General

Communication resource allocation includes the allocation of link and superframe. These services provide the following primitives:

- Link adding services:
  - NLME-ADD-LINK.request,
  - NLME-ADD-LINK.confirm,
  - NLME-ADD-LINK.response, and
  - NLME-ADD-LINK.indication;
- Link update services:
  - NLME-UPDATE-LINK.request,
  - NLME-UPDATE-LINK.confirm,
  - NLME-UPDATE-LINK.response, and
  - NLME-UPDATE-LINK.indication;
- Link release services:
  - NLME-RELEASE-LINK.request,
  - NLME-RELEASE-LINK.confirm,
  - NLME-RELEASE-LINK.response, and
  - NLME-RELEASE-LINK.indication;
- Superframe adding services:
  - NLME-ADD-SFR.request,
  - NLME-ADD-SFR.confirm,
  - NLME-ADD-SFR.response, and
  - NLME-ADD-SFR.indication;
- Superframe update services:
  - NLME-UPDATE-SFR.request,

  - NLME-UPDATE-SFR.confirm,
  - NLME-UPDATE-SFR.response, and
  - NLME-UPDATE-SFR.indication;
- Superframe release services:
  - NLME-RELEASE-SFR.request,
  - NLME-RELEASE-SFR.confirm,
  - NLME-RELEASE-SFR.response, and
  - NLME-RELEASE-SFR.indication.

9.5.8.2 Link adding services

9.5.8.2.1 NLME-ADD-LINK.request

NLME-ADD-LINK.request is used to add one or more record(s) of new link(s), which is originated either from the gateway device to a routing device or from a routing device to a field device.

The semantics of NLME-ADD-LINK.request are described as follows:

```
NLME-ADD-LINK.request (
                        DstAddr,
                        LinkCount,
                        LinkStructure
                      )
```

Table 94 specifies the parameters for NLME-ADD-LINK.request.

Table 94 – NLME-ADD-LINK.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	16-bit destination address
LinkCount	Unsigned16	0 to 65 535	Count of added links to support adding multiple links each time
LinkStructure	Link_Struct structure (See 0)		Information of the links

9.5.8.2.2 NLME-ADD-LINK.confirm

NLME-ADD-LINK.confirm reports the results of NLME-ADD-LINK.request.

The semantics of NLME-ADD-LINK.confirm are described as follows:

```
NLME-ADD-LINK.confirm (
                        Status
                      )
```

Table 95 specifies the parameters for NLME-ADD-LINK.confirm.

Table 95 – NLME-ADD-LINK.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Results of link adding request: 0 = SUCCESS; 1 = TRANSACTION_OVERFLOW; 2 = TRANSACTION_EXPIRED; 3 = NO_ACK; 4 = CHANNEL_ACCESS_FAILURE; 5 = UNAVAILABLE_KEY; 6 = FAILED_SUCURITY_CHECK; 7 = INVALID_PARAMETER; Others are reserved.

The details of the results are shown in the Table 42.

9.5.8.2.3 NLME-ADD-LINK.indication

NLME-ADD-LINK.indication is used to report to the DMAP that the device has successfully received a link adding request packet.

The semantics of NLME-ADD-LINK.indication are described as follows:

```
NLME-ADD-LINK.indication (
                    LinkCount,
                    LinkStructure
                )
```

Table 94 specifies the parameters for NLME-ADD-LINK.indication.

9.5.8.2.4 NLME-ADD-LINK.response

NLME-ADD-LINK.response is the response of NLME-ADD-LINK.indication.

The semantics of NLME-ADD-LINK.response are described as follows:

```
NLME-ADD-LINK.response (
                    Status
                )
```

Table 95 specifies the parameters for NLME-ADD-LINK.response.

9.5.8.2.5 Time sequence for link adding

The time sequence for adding a link originated from the gateway device to a routing device is shown in Figure 53.

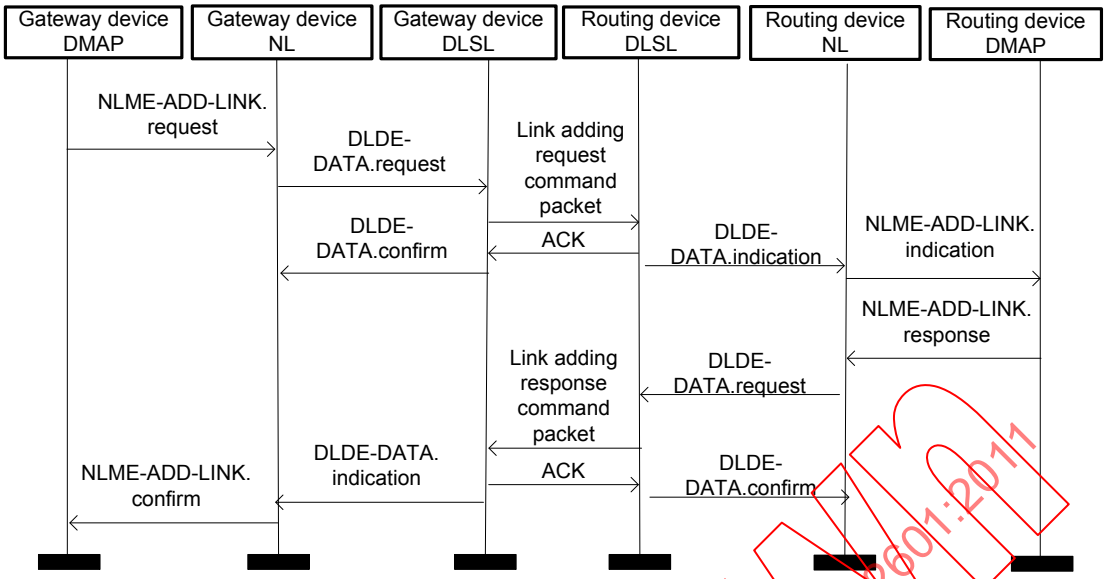

Figure 53 – Adding a link originated from gateway device to routing device

The time sequence for adding a link originated from a routing device to a field device is shown in Figure 54.

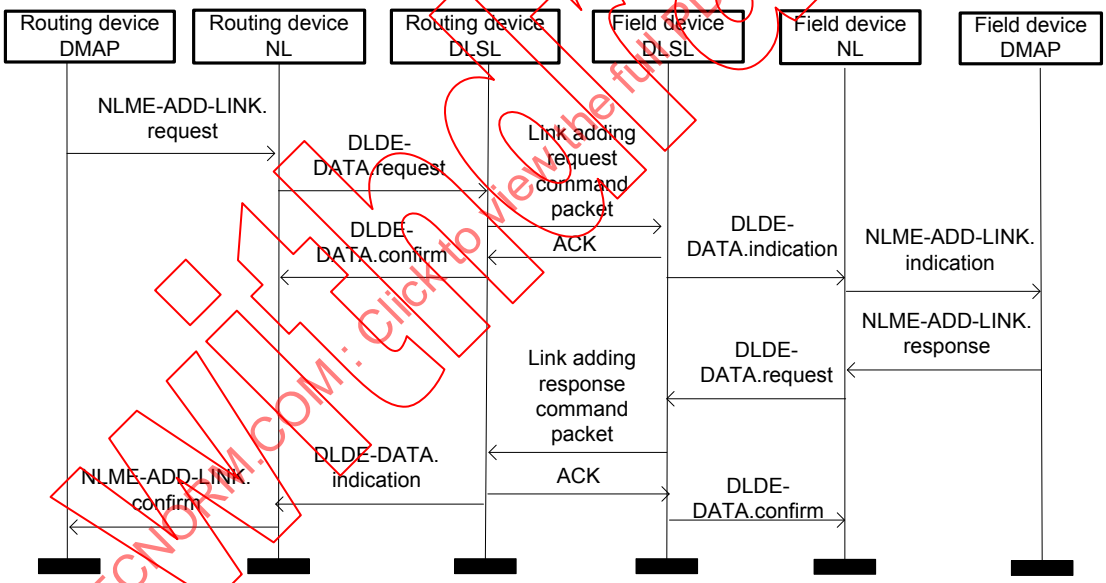

Figure 54 – Adding a link originated from routing device to field device

9.5.8.3 Link update services

9.5.8.3.1 NLME-UPDATE-LINK.request

NLME-UPDATE-LINK.request is used to update one or more record(s) of existing link(s), which is originated either from the gateway device to a routing device or from a routing device to a field device.

The semantics of NLME-UPDATE-LINK.request are described as follows:

```
NLME-UPDATE-LINK.request (
                                DstAddr,
                                LinkCount,
                                LinkStructure
                            )
```

Table 96 specifies the parameters for NLME-UPDATE-LINK.request.

Table 96 – NLME-UPDATE-LINK.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	16-bit destination address
LinkCount	Unsigned16	0 to 65 535	Count of added links to support updating multiple links each time
LinkStructure	Link_Struct structure (See 0)		Information of the links

9.5.8.3.2 NLME-UPDATE-LINK.confirm

NLME-UPDATE-LINK.confirm reports the results of NLME-UPDATE-LINK.request.

The semantics of NLME-UPDATE-LINK.confirm are described as follows:

```
DLME-UPDATA-LINK.confirm (
                                    Status
                                )
```

Table 97 specifies the parameters for NLME-UPDATE-LINK.confirm.

Table 97 – NLME-UPDATE-LINK.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Results of the link update request: 0 = SUCCESS; 1 = TRANSACTION_OVERFLOW; 2 = TRANSACTION_EXPIRED; 3 = NO_ACK; 4 = CHANNEL_ACCESS_FAILURE; 5 = UNAVAILABLE_KEY; 6 = FAILED_SUCURITY_CHECK; 7 = INVALID_PARAMETER; Others are reserved.

The details of the results are shown in Table 42.

9.5.8.3.3 NLME-UPDATE-LINK.indication

NLME-UPDATE-LINK.indication is used to report to the DMAP that the device has successfully received a link update request packet.

The semantics of NLME-UPDATE-LINK.indication are described as follows:

```
NLME-UPDATE-LINK.indication (
                                    LinkCount,
                                    LinkStructure
                                )
```

Table 96 specifies the parameters for NLME-UPDATE-LINK.indication.

9.5.8.3.4 NLME-UPDATE-LINK.response

NLME-UPDATE-LINK.response is the response to NLME-UPDATE-LINK.indication.

The semantics of NLME-UPDATE-LINK.response are described as follows:

NLME-UPDATE-LINK.response (
Status
)

Table 97 specifies the parameters for NLME-UPDATE-LINK.response.

9.5.8.3.5 Time sequence for link update

The time sequence for updating a link originated from the gateway device to a routing device is shown in Figure 55.

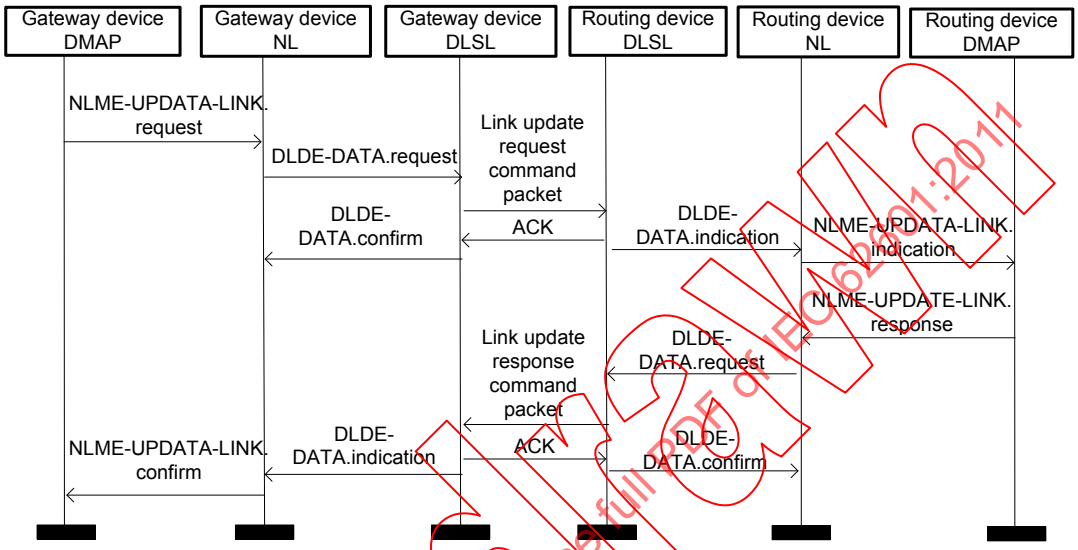

Figure 55 – Updating a link originated by gateway device to routing device

The time sequence for updating a link originated from a routing device to a field device is shown in Figure 56.

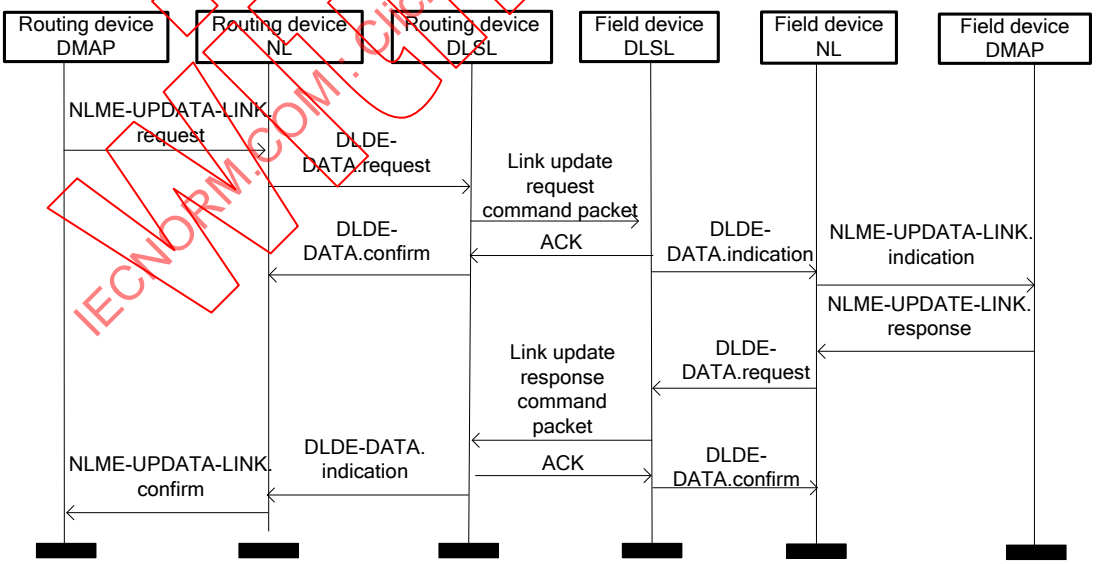

Figure 56 – Updating a link originated from routing device to field device

9.5.8.4 Link release services

9.5.8.4.1 NLME-RELEASE-LINK.request

NLME-RELEASE-LINK.request is used to delete one or more existing link(s), which is originated either from the gateway device to a routing device or from a routing device to a field device.

The semantics of NLME-RELEASE-LINK.request are described as follows:

```
NLME-RELEASE-LINK.request (
                                    DstAddr,
                                    LinkCount,
                                    LinkID[LinkCount]
                                )
```

Table 98 specifies the parameters for NLME-RELEASE-LINK.request.

Table 98 – NLME-RELEASE-LINK.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	16-bit destination address
LinkCount	Unsigned16	0 to 65 535	Count of added links to support releasing multiple links each time
LinkID[LinkCount]	Unsigned16	0 to 65 535	IDs of the deleted links

9.5.8.4.2 NLME-RELEASE-LINK.confirm

NLME-RELEASE-LINK.confirm reports the results of NLME-RELEASE-LINK.request.

The semantics of NLME-RELEASE-LINK.confirm are described as follows:

```
NLME-RELEASE-LINK.confirm (
                                    Status
                                )
```

Table 99 specifies the parameters for NLME-RELEASE-LINK.confirm.

Table 99 – NLME-RELEASE-LINK.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Results of the link release request: 0 = SUCCESS; 1 = TRANSACTION_OVERFLOW; 2 = TRANSACTION_EXPIRED; 3 = NO_ACK; 4 = CHANNEL_ACCESS_FAILURE; 5 = UNAVAILABLE_KEY; 6 = FAILED_SUCURITY_CHECK; 7 = INVALID_PARAMETER; Others are reserved.

The details of the results are shown in Table 42.

9.5.8.4.3 NLME-RELEASE-LINK.indication

NLME-RELEASE-LINK.indication is used to report to the DMAP that the device has successfully received a link release request packet.

The semantics of NLME-RELEASE-LINK.indication are described as follows:

```
NLME-RELEASE-LINK.indication (
                                    LinkCount,
                                    LinkID[LinkCount]
                                )
```

Table 98 specifies the parameters for NLME-RELEASE-LINK.indication.

9.5.8.4.4 NLME-RELEASE-LINK.response

NLME-RELEASE-LINK.response is the response to NLME-RELEASE-LINK.indication.

The semantics of NLME-RELEASE-LINK.response are described as follows:

```
NLME-RELEASE-LINK.response (
                                    Status
                                )
```

Table 99 specifies the parameters for NLME-RELEASE-LINK.response.

9.5.8.4.5 Time sequence for link release

The time sequence for releasing a link originated from the gateway device to a routing device is shown in Figure 57.

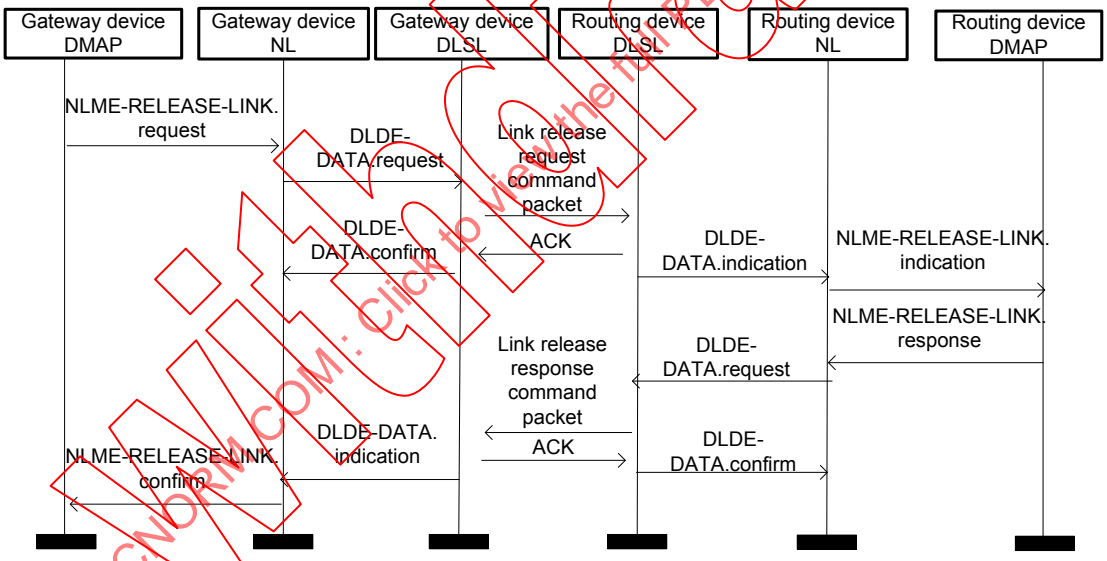

Figure 57 – Releasing a link originated from gateway device to routing device

The time sequence for releasing a link originated from a routing device to a field device is shown in Figure 58.

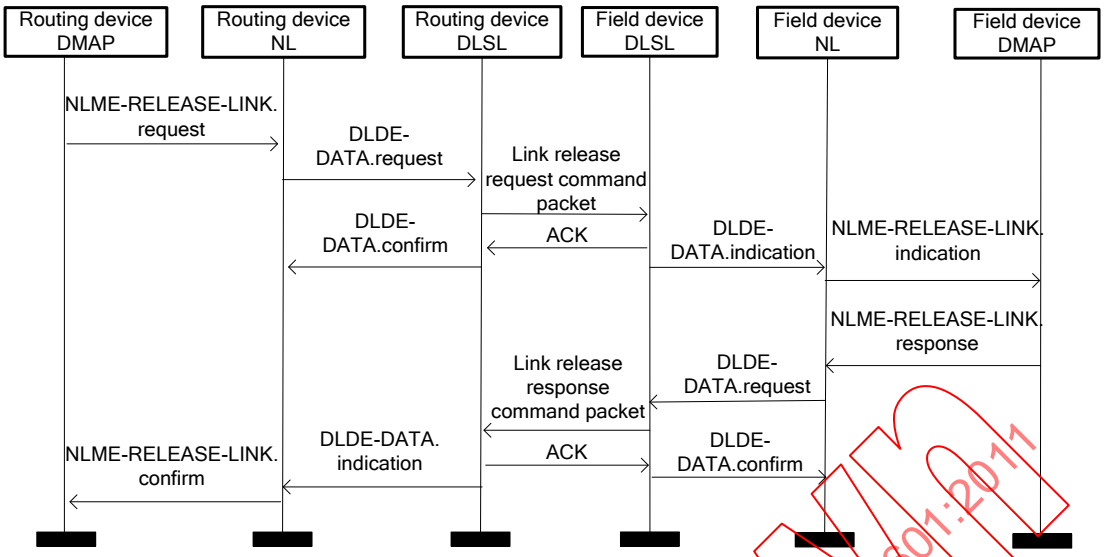

Figure 58 – Releasing a link originated from routing device to field device

9.5.8.5 Superframe adding services

9.5.8.5.1 NLME-ADD-SFR.request

NLME-ADD-SFR.request is used to add a record of a new superframe, which is originated either from the gateway device to a routing device or from a routing device to a field device.

The semantics of NLME-ADD-SFR.request are described as follows:

```
NLME-ADD-SFR.request (
                        DstAddr,
                        SuperframeStructure
                    )
```

Table 100 specifies the parameters for NLME-ADD-SFR.request.

Table 100 – NLME-ADD-SFR.request parameters

Name	Data type	Valid range	Attribute description
DstAddr	Unsigned16	0 to 65 535	16-bit.destination address
SuperframeStructure	Superframe_Struct structure (See Table 15)		Information of Superframe attribute

9.5.8.5.2 NLME-ADD-SFR.confirm

NLME-ADD-SFR.confirm reports the result of NLME-ADD-SFR.request.

The semantics of NLME-ADD-SFR.confirm are described as follows:

```
NLME-ADD-SFR.confirm (
                        Status
                    )
```

Table 101 specifies the parameters for NLME-ADD-SFR.confirm.

Table 101 – NLME-ADD-SFR.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Results of superframe release request: 0 = SUCCESS; 1 = TRANSACTION_OVERFLOW; 2 = TRANSACTION_EXPIRED; 3 = NO_ACK; 4 = CHANNEL_ACCESS_FAILURE; 5 = UNAVAILABLE_KEY; 6 = FAILED_SUCURITY_CHECK; 7 = INVALID_PARAMETER; Others are reserved.

The details of the results are shown in Table 42.

9.5.8.5.3 NLME-ADD-SFR.indication

NLME-ADD-SFR.indication is used to report to the DMAP that the device has successfully received a superframe adding request packet.

The semantics of NLME-ADD-SFR.indication are described as follows:

NLME-ADD-SFR.indication (
SuperframeStructure
)

Table 100 specifies the parameters for NLME-ADD-SFR.indication.

9.5.8.5.4 NLME-ADD-SFR.response

NLME-ADD-SFR.response is the response to NLME-ADD-SFR.indication.

The semantics of NLME-ADD-SFR.response are described as follows:

NLME-ADD-SFR.response (
Status
)

Table 101 specifies the parameters for NLME-ADD-SFR.response.

9.5.8.5.5 Time sequence for superframe adding

The time sequence for adding a superframe originated from the gateway device to a routing device is shown in Figure 59.

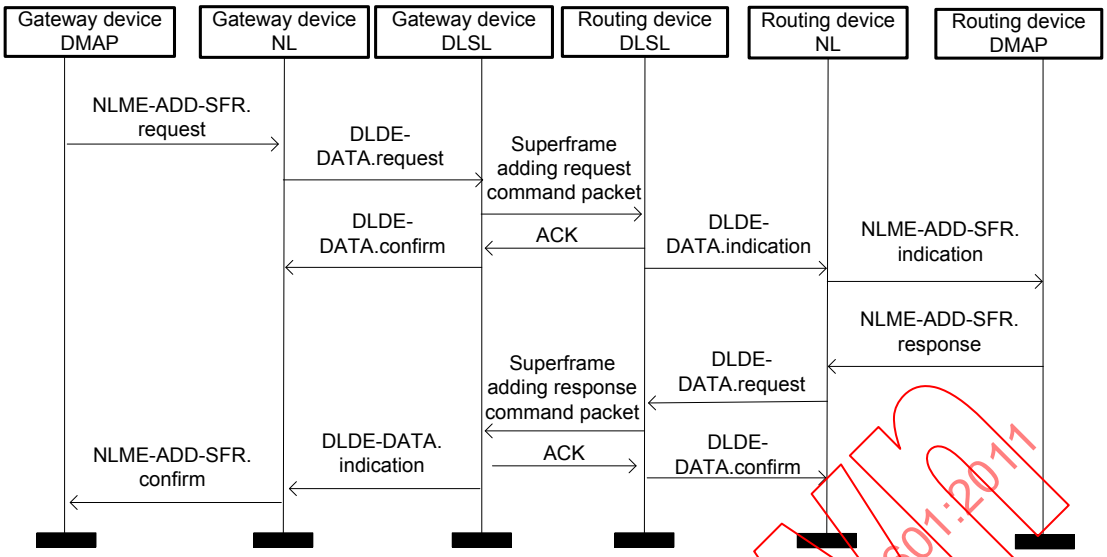

Figure 59 – Adding a superframe originated from gateway device to routing device

The time sequence for adding a superframe originated from a routing device to a field device is shown in Figure 60.

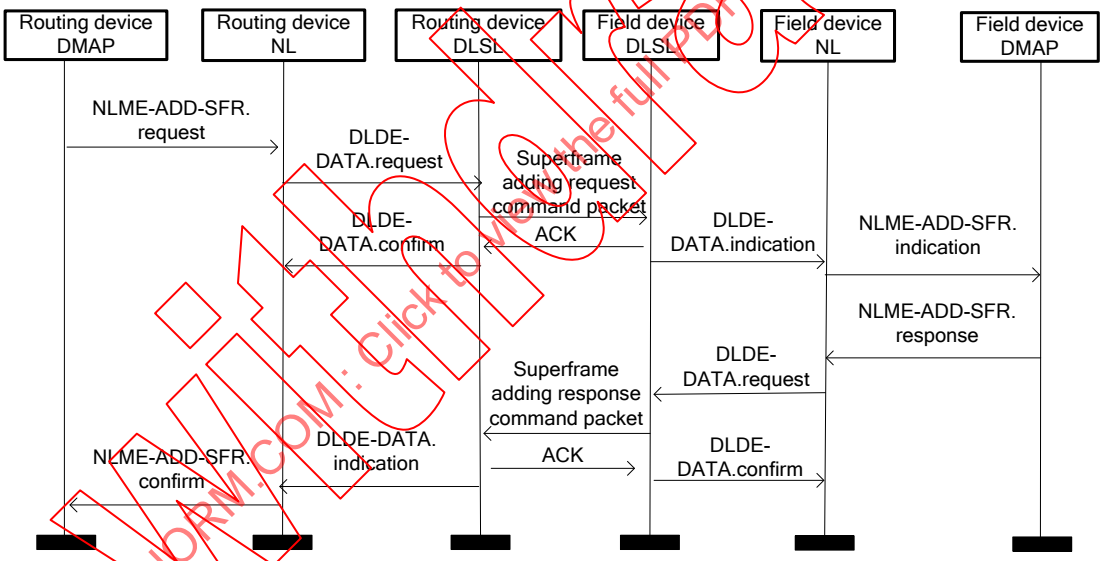

Figure 60 – Adding a superframe originated from routing device to field device

9.5.8.6 Superframe update services

9.5.8.6.1 NLME-UPDATA-SFR.request

NLME-UPDATA-SFR.request is used to modify a record of an existing superframe, which is originated from the gateway device to a routing device or from a routing device to a field device.

The semantics of NLME-UPDATA-SFR.request are described as follows:

```
NLME-UPDATA-SFR.request (
                          DstAddr,
                          SuperframeStructure
                        )
```

Table 102 specifies the parameters for NLME-UPDATA-SFR.request.

Table 102 – NLME-UPDATA-SFR.request parameters

Name	Data type	Valid range	Attribute description
DstAddr	Unsigned16	0 to 65 535	16-bit address of destination device
SuperframeStructure	Superframe_Struct structure (See Table 15)		Information of Superframe attribute

9.5.8.6.2 NLME-UPDATE-SFR.confirm

NLME-UPDATE-SFR.confirm reports the results of NLME-UPDATA-SFR.request.

The semantics of NLME-UPDATE-SFR.confirm are described as follows:

```
NLME-UPDATE-SFR.confirm (
                                    Status
                                  )
```

Table 103 specifies the parameters for NLME-UPDATE-SFR.confirm.

Table 103 – NLME-UPDATE-SFR.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Results of superframe update request: 0 = SUCCESS; 1 = TRANSACTION_OVERFLOW; 2 = TRANSACTION_EXPIRED; 3 = NO_ACK; 4 = CHANNEL_ACCESS_FAILURE; 5 = UNAVAILABLE_KEY; 6 = FAILED_SUCURITY_CHECK; 7 = INVALID_PARAMETER; Others are reserved.

The details of the results are shown in Table 42.

9.5.8.6.3 NLME-UPDATE-SFR.indication

NLME-UPDATE-SFR.indication is used to report to the DMAP that the device has successfully received a superframe update request packet.

The semantics of NLME-UPDATE-SFR.indication are described as follows:

```
NLME-UPDATE-SFR.indication (
                                    SuperframeStructure
                                  )
```

Table 102 specifies the parameters for NLME-UPDATE-SFR.indication.

9.5.8.6.4 NLME-UPDATE-SFR.response

NLME-UPDATE-SFR.response is the response to NLME-UPDATE-SFR.indication.

The semantics of NLME-UPDATE-SFR.response are described as follows:

```
NLME-UPDATE-SFR.response (
                                    Status
                                  )
```

Table 103 specifies the parameters for NLME-UPDATE-SFR.response.

9.5.8.6.5 Time sequence for superframe update

The time sequence for updating a superframe originated from the gateway device to a routing device is shown in Figure 61.

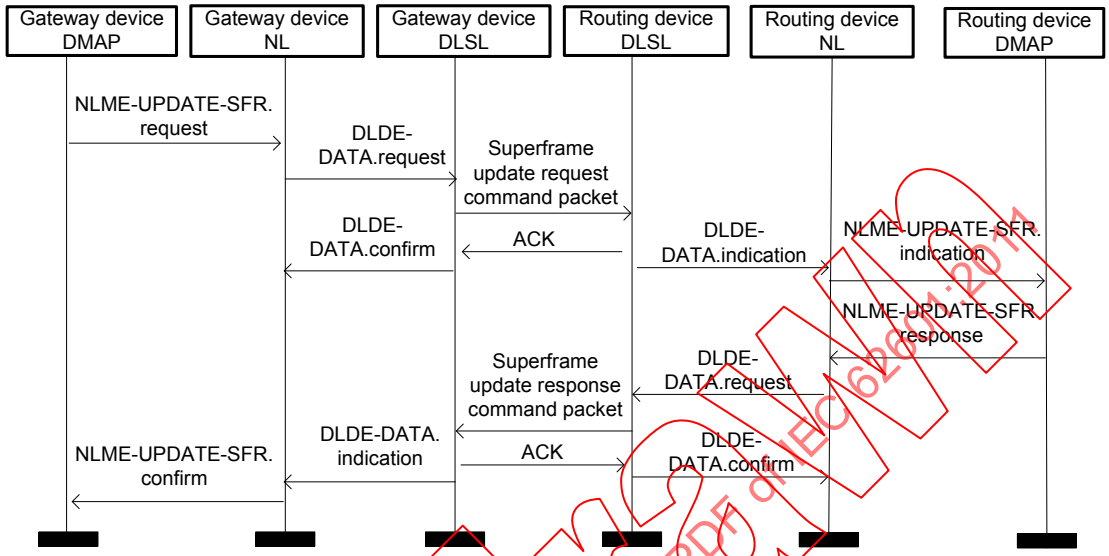

Figure 61 – Updating a superframe originated from gateway device to routing device

The time sequence for updating a superframe originated from a routing device to a field device is shown in Figure 62.

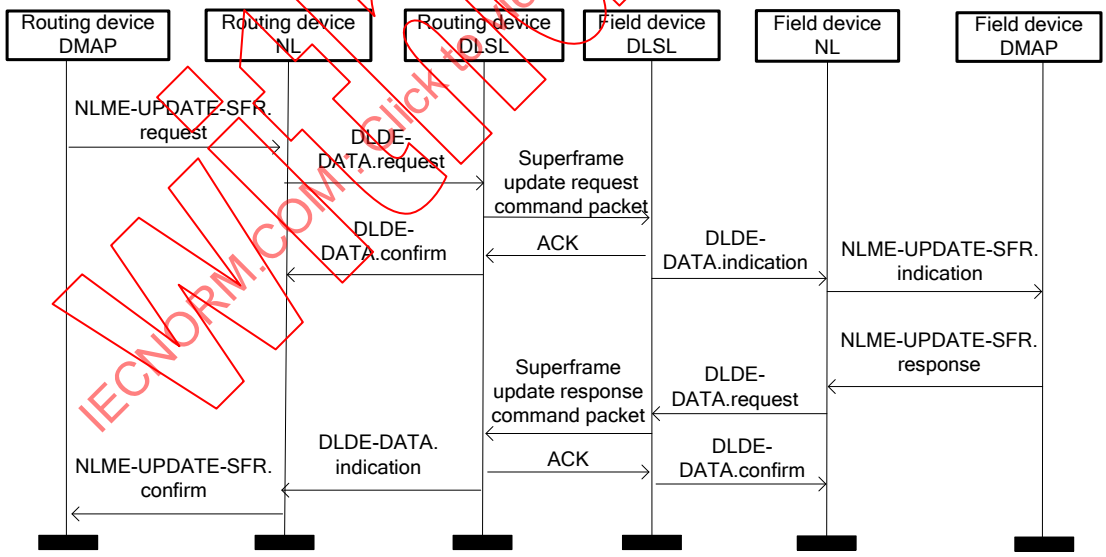

Figure 62 – Updating a superframe originated from routing device to field device

9.5.8.7 Superframe release services

9.5.8.7.1 NLME-RELEASE-SFR.request

NLME-RELEASE-SFR.request is used to delete a record of an existing superframe, which is originated either from the gateway device to a routing device or from a routing device to a field device.

The semantics of NLME-RELEASE-SFR.request are described as follows:

NLME-RELEASE-SFR.request (
DstAddr,
SuperframeID
)

Table 104 specifies the parameters for NLME-RELEASE-SFR.request.

Table 104 – NLME-RELEASE-SFR.request parameters

Name	Data type	Valid range	Descrition
DstAddr	Unsigned16	0 to 65 535	16-bit.destination address
SuperframeID	Unsigned16	0 to 65 535	ID of the deleted superframe

9.5.8.7.2 NLME-RELEASE-SFR.confirm

NLME-RELEASE-SFR.confirm reports the result of NLME-RELEASE-SFR.request.

The semantics of NLME-RELEASE-SFR.confirm are described as follows:

NLME-RELEASE-SFR.confirm (
Status
)

Table 105 specifies the parameters for NLME-RELEASE-SFR.confirm.

Table 105 – NLME-RELEASE-SFR.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Results of superframe release request: 0 = SUCCESS; 1 = TRANSACTION_OVERFLOW; 2 = TRANSACTION_EXPIRED; 3 = NO_ACK; 4 = CHANNEL_ACCESS_FAILURE; 5 = UNAVAILABLE_KEY; 6 = FAILED_SUCURITY_CHECK; 7 = INVALID_PARAMETER; Others are reserved.

The details of the results are shown in Table 42.

9.5.8.7.3 NLME-RELEASE-SFR.indication

NLME-RELEASE-SFR.indication is used to report to the DMAP that the device has successfully received a superframe release request packet.

The semantics of NLME-RELEASE-SFR.indication are described as follows:

NLME-RELEASE-SFR.indication (
SuperframeID
)

Table 104 specifies the parameters for NLME-RELEASE-SFR.indication.

9.5.8.7.4 NLME-RELEASE-SFR.response

NLME-RELEASE-SFR.response is the response of NLME-RELEASE-SFR.indication.

The semantics of NLME-RELEASE-SFR.response are described as follows:

NLME-RELEASE-SFR.response (
Status
)

Table 105 specifies the parameters for NLME-RELEASE-SFR.response.

9.5.8.7.5 Time sequence for superframe release

The time sequence for releasing a superframe originated from the gateway device to a routing device is shown in Figure 63.

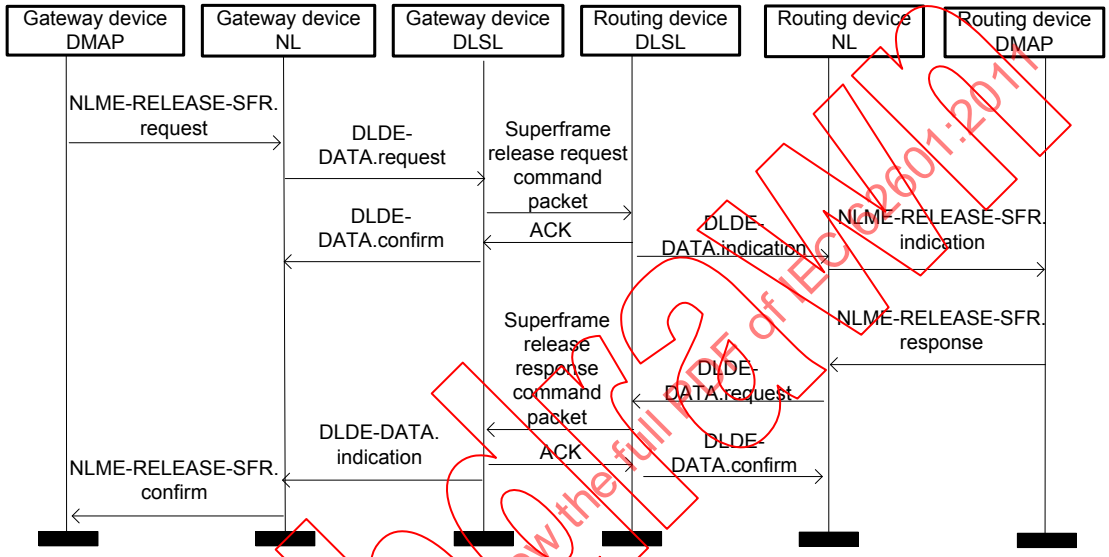

Figure 63 – Releasing a superframe originated from gateway device to routing device

The time sequence for releasing a superframe originated from a routing device to a field device is shown in Figure 64.

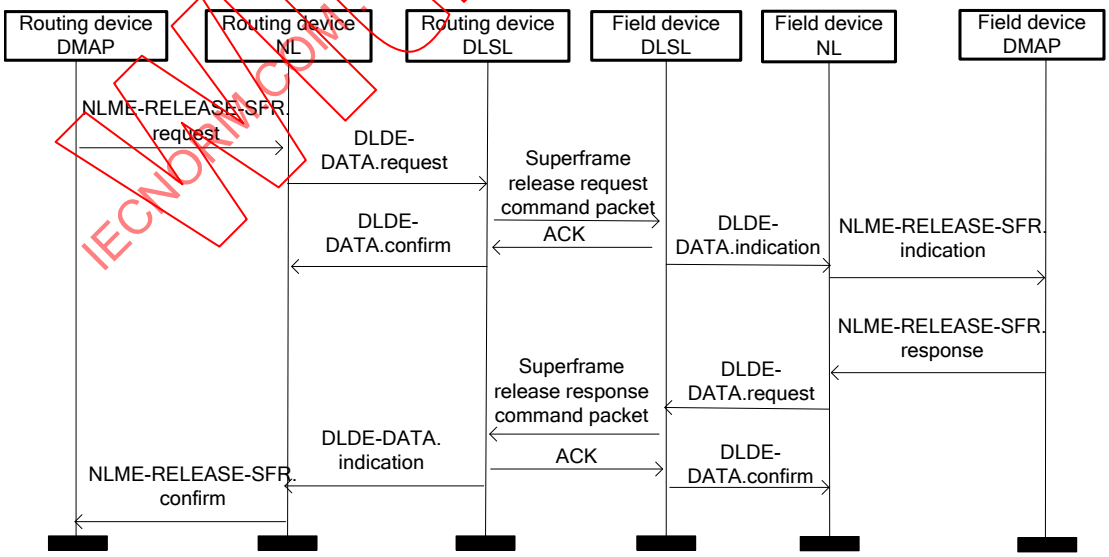

Figure 64 – Releasing a superframe originated from routing device to field device

9.5.9 Aggregation and disaggregation services

9.5.9.1 NLME-AGG.indication

NLME-AGG.indication is used for the NL to report to the DMAP that the received packets need to be aggregated.

The semantics of NLME-AGG.indication are described as follows:

```
NLME-AGG.indication (
                        SrcAddr,
                        NwkPayload
                    )
```

Table 106 specifies the parameters for NLME-AGG.indication.

Table 106 – NLME-AGG.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	ID of the aggregation VCR
NwkPayload	Octetstring		Payload of NL

9.5.9.2 NLME-AGO-SEND.request

NLME-AGO-SEND.request is used by the DMAP asking the NL to send the aggregated packet after the aggregation is completed.

The semantics of NLME-AGO-SEND.request are described as follows:

```
NLME-AGO-SEND.request (
                        GatewayAddr,
                        AggPacketFlag,
                        Priority,
                        AggNumber,
                        AGGRouteID,
                        PSFlag,
                        AggPayload
                      )
```

Table 107 specifies the parameters for NLME-AGO-SEND.request.

Table 107 – NLME-AGO-SEND.request parameters

Name	Data type	Valid range	Description
GatewayAddr	Unsigned16	0 to 65 535	Network address of the gateway device
AggPacketFlag	Unsigned8	0, 1	Indicating whether this packet is a data packet or aggregation packet: 0 = Data packet; 1 = Aggregation packet.
Priority	Unsigned8	0 to 15	Priority of process data
AggNumber	Unsigned8	0 to 255	Number of aggregated packets
AGGRouteID	Unsigned16	0 to 65 535	RouteID of the aggregation
PSFlag	Unsigned8	0, 1	Indicating whether this packet is P/S type: 0 = Not P/S type; 1 = P/S type.
AggPayload	Octetstring		DMAP aggregated packet

9.5.9.3 NLME-DAG.indication

NLME-DAG.indication is used to send the aggregated packet to the DMAP for disaggregating when the NL receives an aggregated packet.

The semantics of NLME-DAG.indication are described as follows:

```
NLME-DAG.indication (
                        AggNumber,
                        NwkPalyload
                    )
```

Table 108 specifies the parameters for NLME-DAG.indication.

Table 108 – NLME-DAG.indication parameter

Name	Data type	Valid range	Description
AggNumber	Unsigned8	0 to 255	Number of aggregated packets
NwkPayload	Octetstring		Payload of NL

9.5.10 Device status report services

9.5.10.1 NLME-DEVICE-STATUS.request

NLME-DEVICE-STATUS.request is used to report the device status either to the routing devices by a field device or to the gateway device by a routing device.

The semantics of NLME- DEVICE -STATUS.request are described as follows:

```
NLME-CHANNEL- DEVICE.request (
                                DstAddr,
                                DeviceCount,
                                DeviceConditionInfo
                            )
```

Table 109 specifies the parameters for NLME- DEVICE -STATUS.request.

Table 109 – NLME- DEVICE -STATUS.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	Destination address
DeviceCount	Unsigned8	0 to 255	Total number of reported devices
DeviceConditionInfo	DevConRep_Struct structure(see Table 21)		Information of device condition attributes

9.5.10.2 NLME- DEVICE-STATUS.indication

NLME-DEVICE-STATUS.indication is used to report the receipt of a device condition report command packet to the DMAP.

The semantics of NLME- DEVICE -STATUS.indication are described as follows:

```
NLME- DEVICE -STATUS.indication (
                                    SrcAddr,
                                    DeviceCount,
                                    DeviceConditionInfo
                                )
```

Table 110 specifies the parameters for NLME- DEVICE -STATUS.indication.

Table 110 – NLME- DEVICE -STATUS.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	Source address
DeviceCount	Unsigned8	0 to 255	Total number of reported devices
DeviceConditionInfo	DevConRep_Struct structure (see Table 21)		Information of device condition attributes

9.5.10.3 NLME- DEVICE-STATUS.confirm

NLME-DEVICE-STATUS.confirm is used to return the results of NLME-DEVICE-STATUS.request.

The semantics of NLME-DEVICE-STATUS.confirm are described as follows:

NLME- DEVICE -STATUS.confirm (
Status
)

Table 111 specifies the parameters for NLME- DEVICE -STATUS.confirm.

Table 111 – NLME- DEVICE -STATUS.confirm parameter

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Result of the channel condition report request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.5.10.4 Time sequence for device status reporting

The time sequence diagram for device status information is shown in Figure 65 and Figure 66.

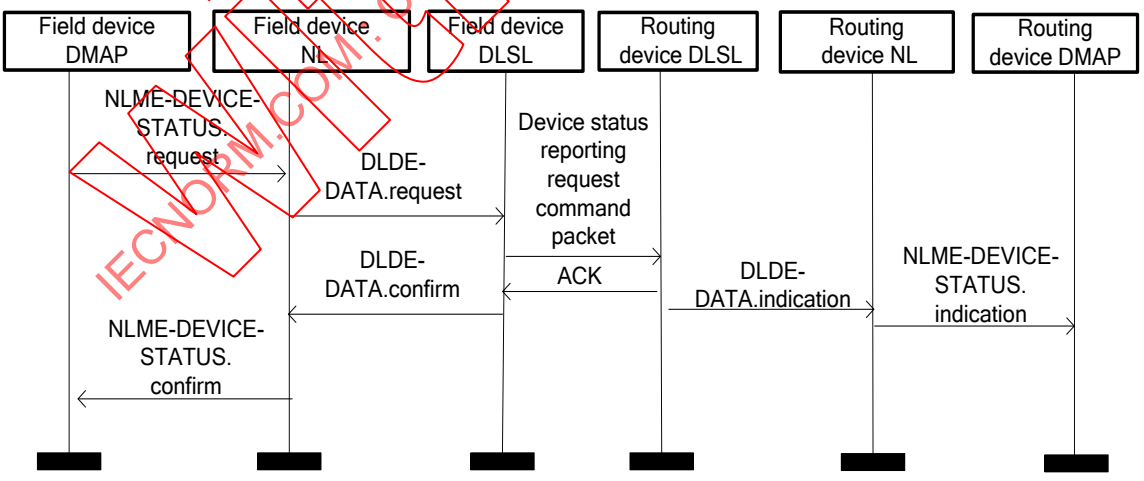

Figure 65 – Device status reporting process from field device to routing device

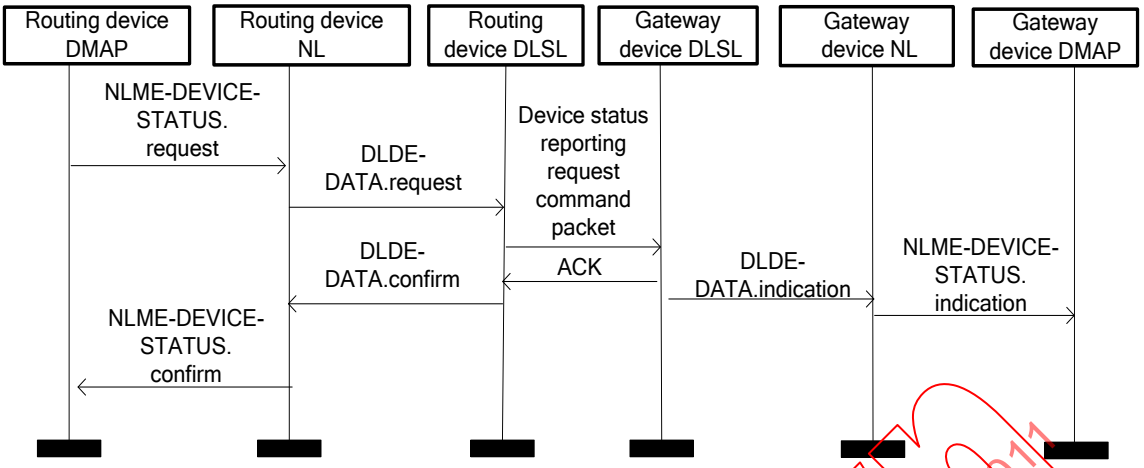

Figure 66 – Device status reporting process from routing device to gateway device

9.5.11 Channel condition report services

9.5.11.1 NLME-CHANNEL-CONDITION.request

NLME-CHANNEL-CONDITION.request is used to report the communication channel condition either to the routing device by a field device or to the gateway device by a routing device.

The semantics of NLME-CHANNEL-CONDITION request are described as follows:

```
NLME-CHANNEL-CONDITION.request (
                                    DstAddr,
                                    ChannelCount,
                                    ChannelConditionInfo
                                )
```

Table 112 specifies the parameters for NLME-CHANNEL-CONDITION.request.

Table 112 – NLME-CHANNEL-CONDITION.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	Destination address
ChannelCount	Unsigned8	0 to 255	Total number of channels
ChannelConditionInfo	ChanCon_Struct structure (See Table 18)		Information of channel condition attributes

9.5.11.2 NLME-CHANNEL-CONDITION.indication

NLME-CHANNEL-CONDITION.indication is used to report the receipt of a channel condition report command packet to the DMAP.

The semantics of NLME-CHANNEL-CONDITION.indication are described as follows:

```
NLME-CHANNEL-CONDITION.indication (
                                    SrcAddr,
                                    ChannelCount,
                                    ChannelConditionInfo
                                )
```

Table 113 specifies the parameters for NLME-CHANNEL-CONDITION.indication.

Table 113 – NLME-CHANNEL-CONDITION.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	Source address
ChannelCount	Unsigned8	0 to 255	Total number of channels
ChannelConditionInfo	ChanCon_Struct structure (see Table 18)		Information of channel condition attributes

9.5.11.3 NLME-CHANNEL-CONDITION.confirm

NLME-CHANNEL-CONDITION.confirm is used to return the results of NLME-CHANNEL-CONDITION.request.

The semantics of NLME-CHANNEL-CONDITION.confirm are described as follows:

NLME-CHANNEL-CONDITION.confirm (
Status
)

Table 114 specifies the parameters for NLME-CHANNEL-CONDITION.confirm.

Table 114 – NLME-CHANNEL-CONDITION.confirm parameter

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Result of the channel condition report request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.5.11.4 Time sequence for channel condition reporting

The time sequence diagram for channel condition information is shown in Figure 67 and Figure 68.

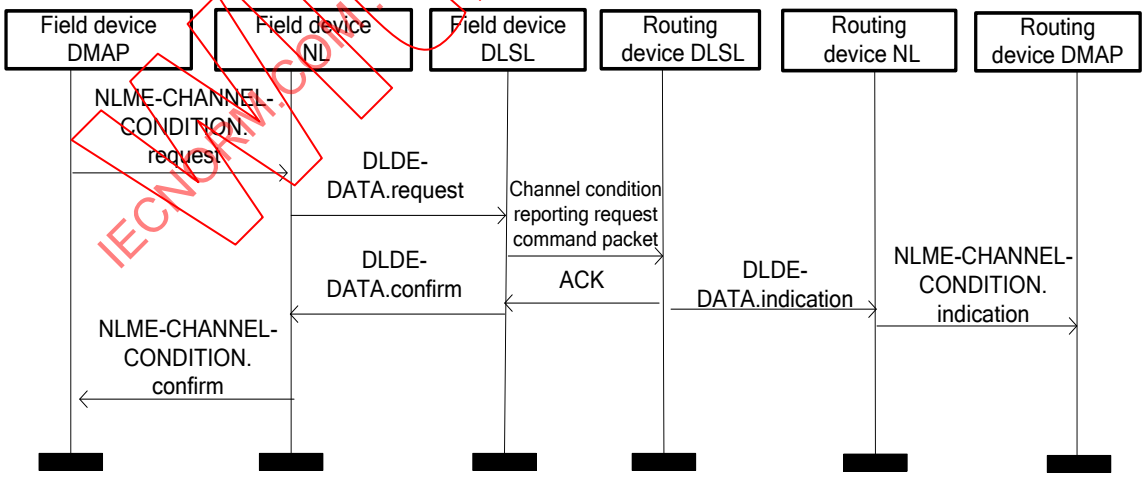

Figure 67 – Channel condition reporting process from field device to routing device

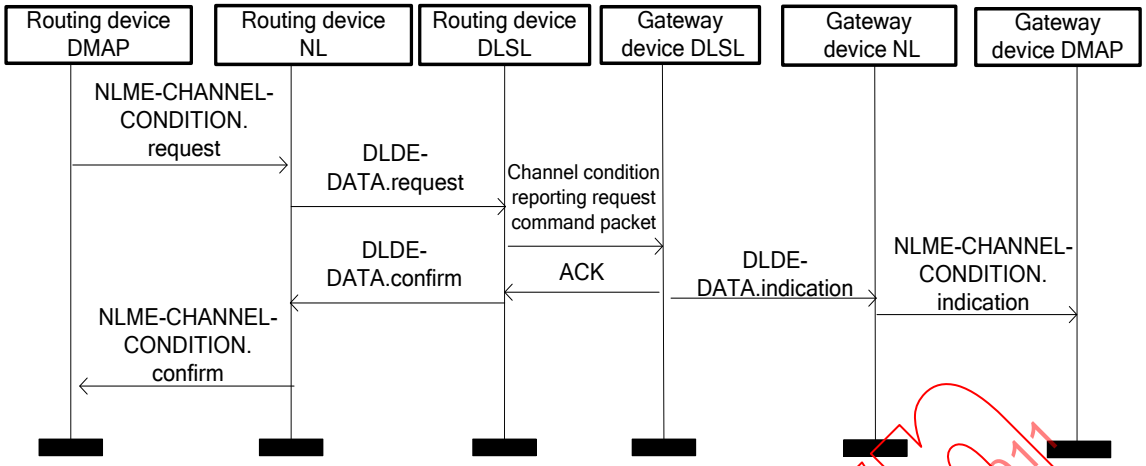

Figure 68 – Channel condition reporting process from routing device to gateway device

9.5.12 Failure path report services

9.5.12.1 General

The NL defines the expired time *EtoEACKTimeOut* of end-to-end packet transmission for each path. If the NL has not received a response during this time, the NL retries the packet. For each path, the NL sets a retry counter. The retry counter should be increased by one after each retry. If the retry is interrupted, the counter should be set to 0; otherwise, if the number of the counter exceeds the set value *MaxEtoERetry*, the path is failed. When the path is failed, the routing device should report the failure of this path to the gateway device by using the redundant path.

9.5.12.2 NLME-PATH_FAILURE.request

NLME-PATH_FAILURE.request is used to report failed paths to the GW by routing devices.

The semantics of NLME-PATH_FAILURE.request are described as follows:

```
NLME-PATH_FAILURE.request (
                                SrcAddr,
                                DstAddr,
                                RouteID
                            )
```

Table 115 specifies the parameters for NLME-PATH_FAILURE.request.

Table 115 – NLME-PATH_FAILURE.request parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	16-bit source address of packets
DstAddr	Unsigned16	0 to 65 535	16-bit destination address of packets
RouteID	Unsigned16	0 to 65 535	Route ID of failing path

9.5.12.3 NLME-PATH_FAILURE.indication

NLME-PATH_FAILURE.indication is used by the NL to report to the DMAP that the device has successfully received a failed-path reporting request packet.

The semantics of NLME-PATH_FAILURE.indication are described as follows:

```
NLME-PATH_FAILURE.indication (
                                SrcAddr,
```

RouteID
)

Table 116 specifies the parameters for NLME-LEAVE.response.

Table 116 – NLME-PATH_FAILURE.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	16-bit source address of packets
RouteID	Unsigned16	0 to 65 535	Route ID of failing path

9.5.12.4 NLME-PATH_FAILURE.confirm

NLME-PATH_FAILURE.confirm is used to return the results of NLME-PATH_FAILURE.request.

The semantics of NLME-PATH_FAILURE.confirm are described as follows:

NLME-PATH_FAILURE.confirm (
Status
)

Table 117 specifies the parameters for NLME-PATH_FAILURE.confirm.

Table 117 – NLME-PATH_FAILURE.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	The result of NLME-PATH_FAILURE.request: 0 = SUCCESS; 1 = FAILURE ; Others are reserved.

9.5.12.5 Time sequence for failure path reporting

The time sequence diagram for failure path information is shown in Figure 69.

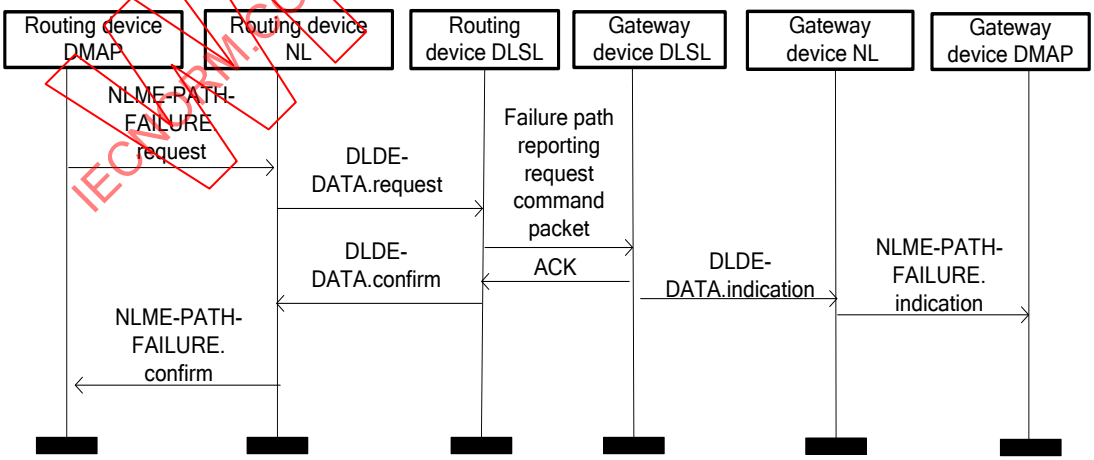

Figure 69 – Failure path reporting process

9.5.13 Network attribute getting services

9.5.13.1 NLME-INFO_GET.request

NLME-INFO_GET.request is used to remotely read the values of the attributes in the MIB.

The semantics of NLME-INFO_GET.request are described as follows:

```
NLME-INFO_GET.request (
                        DstAddr,
                        AttributeID,
                        AttributeMemID,
                        FirstValueStorIndex,
                        Count
                      )
```

Table 118 specifies the parameters for NLME-INFO_GET.request.

Table 118 – NLME-INFO_GET.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	16-bit short address of destination
AttributeID	Unsigned8	0 to 255	ID of attribute in MIB
AttributeMemID	Unsigned8	0 to 255	The ID of attribute member, which is used to read the structured MIB attributes The value 255 means that all attributes should be read.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to read the structured MIB attributes
Count	Unsigned8	0 to 255	Number of attribute values or attributes member values, which is used to read the structured MIB attributes; Getting all attribute values from *FirstValueStorIndex* if *Count* = 0.

9.5.13.2 NLME-INFO_GET.indication

NLME-INFO_GET.indication is used to inform the DMAP of the successful receipt of an attribute getting request packet.

The semantics of NLME-INFO_GET.indication are described as follows:

```
NLME-INFO_GET.indication (
                           SrcAddr,
                           AttributeID,
                           AttributeMemID,
                           FirstValueStorIndex,
                           Count
                         )
```

Table 119 specifies the parameters for NLME-INFO_GET.indication.

Table 119 – NLME-INFO_GET. indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	16-bit short address of source
AttributeID	Unsigned8	0 to 255	The ID of attribute in MIB
AttributeMemID	Unsigned8	0 to 255	The ID of attribute member, which is used to read the structured MIB attributes The value 255 means that all attributes should be read.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to read the structured MIB attributes
Count	Unsigned8	0 to 255	Number of attribute values or attributes member values, which is used to read the structured MIB attributes; all attribute values from *FirstValueStorIndex* if *Count* = 0.

9.5.13.3 NLME-INFO_GET.response

NLME-INFO_GET.response is used to respond to NLME-INFO_GET.request.

The semantics of NLME-INFO_GET.response are described as follows:

```
NLME-INFO_GET.response (
                        DstAddr,
                        Status,
                        AttributeID,
                        AttributeMemID,
                        FirstValueStorIndex,
                        Count,
                        AttributeValue
                        )
```

Table 120 specifies the parameters for NLME-INFO_GET.response.

Table 120 – NLME-INFO_GET.response parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	16-bit short address of the destination
Status	Unsigned8	0 to 255	Result of the attribute read request: 0 = SUCCESS; 1 = UNSUPPORTED_ATTRIBUTE; 2 to 255 = Reserved.
AttributeID	Unsigned8	0 to 255	ID of attribute in MIB
AttributeMemID	Unsigned8	0 to 255	ID of attribute member, which is used to read the structured MIB attributes The value 255 means that all attributes should be read.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to read the structured MIB attributes
Count	Unsigned8	0 to 255	Number of attribute values or attributes member values, which is used to read the structured MIB attributes; all attribute values from *FirstValueStorIndex* if *Count* = 0.
AttributeValue	Octetstring		Value of the attribute to be read

If the operation of getting attributes is successful, the "*Status*" should be "SUCCESS"; if the MIB does not have the needed attributes, the "*Status*" should be "UNSUPPORTED_ATTRIBUTE".

9.5.13.4 NLME-INFO_GET.confirm

NLME-INFO_GET.confirm is used to return the result of NLME-INFO_GET.request.

The semantics of NLME-INFO_GET.confirm are described as follows:

```
NLME-INFO_GET.confirm (
                        SrcAddr,
                        Status,
                        AttributeID,
                        AttributeMemID,
                        FirstValueStorIndex,
                        Count,
                        AttributeValue
                    )
```

Table 121 specifies the parameters for NLME-INFO_GET.confirm.

Table 121 – NLME-INFO_GET.response parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	16-bit short address of the source
Status	Unsigned8	0 to 255	Result of the attribute read request: 0 = SUCCESS; 1 = UNSUPPORTED_ATTRIBUTE; Others are reserved.
AttributeID	Unsigned8	0 to 255	ID of attribute in MIB
AttributeMemID	Unsigned8	0 to 255	ID of attribute member, which is used to read the structured MIB attributes The value 255 means that all attributes should be read.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to read the structured MIB attributes
Count	Unsigned8	0 to 255	Number of attribute values or attributes member values, which is used to read the structured MIB attributes; all attribute values from *FirstValueStorIndex* if *Count* = 0.
AttributeValue	Octetstring		Value of the attribute to be read

9.5.14 Network attribute setting services

9.5.14.1 NLME-INFO_SET.request

NLME-INFO_SET.request is used to remotely modify the values of attributes in the MIB.

The semantics of NLME-INFO_SET.request are described as follows:

```
NLME-INFO_SET.request (
                        DstAddr,
                        AttributeID,
                        AttributeMemID,
                        FirstValueStorIndex,
                        Count,
                        AttributeValue
                    )
```

Table 122 specifies the parameters for NLME-INFO_SET.request.

Table 122 – NLME-INFO_SET.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	16-bit short address of the destination
AttributeID	Unsigned8	0 to 255	ID of attribute in MIB
AttributeMemID	Unsigned8	0 to 255	The ID of attribute member, which is used to write the structured MIB attributes The value 255 means that all attributes should be written
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to write the structured MIB attributes
Count	Unsigned8	0 to 255	Number of attribute values or attributes member values, which is used to read the structured MIB attributes
AttributeValue	Octetstring		Value of the attribute to be written

9.5.14.2 NLME-INFO_SET.indication

NLME-INFO_SET.indication is used to inform the DMAP of the successful receipt of an attribute setting request command packet.

The semantics of NLME-INFO_SET.indication are described as follows:

```
NLME-INFO_SET.indication (
                    SrcAddr,
                    AttributeID,
                    AttributeMemID,
                    FirstValueStorIndex,
                    Count,
                    AttributeValue
                    )
```

Table 123 specifies the parameters for NLME-INFO_SET.indication.

Table 123 – NLME-INFO_SET.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	16-bit short address of the source
AttributeID	Unsigned8	0 to 255	ID of attribute in MIB
AttributeMemID	Unsigned8	0 to 255	ID of attribute member, which is used to write the structured MIB attributes The value 255 means that all attributes should be written.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to write the structured MIB attributes
Count	Unsigned8	0 to 255	Number of attribute values or attributes member values, which is used to read the structured MIB attributes
AttributeValue	Octetstring		Value of the attribute to be written

9.5.14.3 NLME-INFO_SET.response

NLME-INFO_SET.response is used to respond to NLME-INFO_SET.indication.

The semantics of NLME-INFO_SET.response are described as follows:

```
NLME-INFO_SET.response (
                        DstAddr,
                        Status,
                        AttributeID,
                        AttributeMemID,
                        FirstValueStorIndex,
                        Count
                    )
```

Table 124 specifies the parameters for NLME-INFO_SET.response.

Table 124 – NLME-SET. response parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	16-bit short address of the destination
Status	Unsigned8	0 to 255	Result of the attribute setting request: 0 = SUCCESS; 1 = UNSUPPORTED_ATTRIBUTE; 2 = INVALID_PARAMETER; Others are reserved.
AttributeMemID	Unsigned8	0 to 255	ID of attribute member, which is used to write the structured MIB attributes The value 255 means that all attributes should be written.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to write the structured MIB attributes
AttributeValue	Octetstring		Value of the attribute to be written

If the operation of setting attributes is successful, the "*Status*" should be "SUCCESS"; if the MIB does not have the needed attributes, the "*Status*" should be "UNSUPPORTED_ATTRIBUTE"; otherwise, if the set attributes do not conform to the specified attributes, the "*Status*" should be "INVALID_PARAMETER".

9.5.14.4 NLME-INFO_SET.confirm

NLME-INFO_SET.confirm is used to return the results of NLME-INFO_SET.request.

The semantics of NLME-INFO_SET.confirm are described as follows:

```
NLME-INFO_SET.confirm (
                        SrcAddr,
                        Status,
                        AttributeID,
                        AttributeMemID,
                        FirstValueStorIndex,
                        Count
                    )
```

Table 125 specifies the parameters for NLME-INFO_SET.confirm.

Table 125 – NLME-SET.confirm parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	16-bit short address of the source
Status	Unsigned8	0 to 255	Result of the attribute setting request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.
AttributeMemID	Unsigned8	0 to 255	ID of attribute member, which is used to write the structured MIB attributes The value 255 means that all attributes should be written.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to write the structured MIB attributes
AttributeValue	Octetstring		Value of the attribute to be written

9.6 Network layer packet formats

9.6.1 Common packet format

The NL packet format is shown in Figure 70.

Network layer header										Network layer payload
Control field	Routing field			Time-stamp	Priority	Sequence Number	Number of fragment	Fragment sequence number	Payload length	Network layer payload
	Destination address	Source address	RouteID							
Octet:1	2	2	2	4	1	1	0/1	0/1	1	Variable length

Figure 70 – Network layer common packet format

The control field format is shown in Table 126.

Table 126– Control field format

Length in bit(s)	2	1	1	1	3
Subfield name	Packet type	Fragment flag	P/S flag	Authentication flag	Reserved

The NL header field has the following subfields:

a) Control subfield, which includes:

- Packet type: this subfield has 2-bit length; the value 0 indicates a data packet, 1 indicates an aggregation packet, 2 indicates a command packet, and others are reserved;
- Fragment flag: this subfield has 1-bit length and is used to indicate whether this packet is a fragment packet; the valud 0 indicates no fragment and 1 indicates fragment;
- P/S flag: this subfield has 1-bit length, which is used to indicate whether the data in this packet is P/S type; the valud 0 indicates non P/S type and 1 indicates P/S type;
- Authentication flag: this subfield has 1-bit length; the valud 0 indicates that the network does not need authentication and 1 indicates that the network needs authentication;
- Reserved: this subfield has 3-bit length and is used for extension.

b) Routing field, which includes:

- Destination address: final destination address of a packet (16 bits);

- Source address: the address from which a packet originates (16 bits);
- RouteID: unique identifier of the path (16 bits);

c) Timestamp: this subfield has 4-octet length and is labeled in microseconds;

d) Priority: this subfield has 1-octet length and indicates the priority of a packet;

e) Sequence number: this subfield has 1-octet length and indicates the sequence number of the NL packet;

f) Number of fragment: this subfield has 0- or 1-octet length; it is used to indicate the number of fragments; if the packet is a fragment packet, this subfield has 1-octet length and its value is valid; otherwise, this subfield is 0-octet long.

g) Fragment sequence number: the fragment sequence number subfield is used to indicate the fragment sequence of the fragmentation packet. If the packet is a fragmentation packet, this field has 1-octet length and is valid; otherwise, this field is invalid.

h) Payload length: this field has 1-octet length and is used to indicate the length of the NL payload;

i) Network layer payload: this field has variable length and is used to fill in the NL data.

9.6.2 Data packet format

The data packet format is shown in Figure 71.

Network layer header										Network layer payload
Control field	Routing field			Timestamp	Priority	Sequence Number	Number of fragment	Fragment sequence number	Payload length	Network layer payload
	Destination address	Source address	RouteID							
Octet:1	2	2	2	4	1	1	0/1	0/1	1	Variable length

NOTE The value of packet type in control field of Network layter header = 0, which indicates a data packet.

Figure 71 – Network layer data packet format

The definitions of all the subfields are the same as those in Figure 70 and Table 126.

9.6.3 Aggregated packet format

The aggregated packet format is described in NOTE In this Figure, the value of packet type in control field of Network layter header = 1, which indicates an aggregation packet

Figure 72.

Network layer header	Network layer payload							
Network layer header	Aggregated number	The first aggregated data			...	The n^{th} aggregated data		
		Source address 1	Data length1	Data1	...	Source address n	Data length n	Data n
Octet:14/16	1	2	1	Variable length	...	2	1	Variable length

NOTE In this Figure, the value of packet type in control field of Network layter header = 1, which indicates an aggregation packet

Figure 72 – Aggregated packet format

The aggregated packet has the following fields or subfields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Aggregated number: this subfield is used to record the number of packets that are aggregated;

c) First aggregated data: this subfield is used to fill in the first aggregated data, which includes:

- Source address 1: this field has 2–octet length and indicates the 16-bit address of the first aggregated data;
- Data length 1: this field has 1-octet length, which is used to fill in the length of the first aggregated data;
- Data 1: this field has variable length and is used to fill in the first aggregated data.

d) n^{th} aggregated data: this subfield is used to fill in the n^{th} aggregated data, which is the same as the first aggregated data.

9.6.4 Command packet format

9.6.4.1 Common format of command packet

The NL command packets encapsulate the NL commands to accomplish some management functions. The format of the command packet is shown in Figure 73.

Network layer header										Network layer payload	
Control field	Routing field			Timestamp	Priority	Sequence Number	Number of fragment	Fragment sequence number	Payload length	Command packet identifier	Command packet payload
	Destination address	Source address	RouteID								
Octet:1	2	2	2	4	1	1	0/1	0/1	1	1	Variable length

NOTE The value of packet type in control field of Network layer header = 2, which indicates a command packe.

Figure 73 – Format of NL command packet

The data packet has the following fields or subfields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) The network layer payload field includes the following subfields:

- Command packet identifier: this subfield has 1-octet length and is used to indicate the identifier of the NL command packet (see Table 127);
- Command packet payload: this subfield has variable length and is used to fill in the payload of the NL command packet.

The NL command packets are listed in Table 127.

Table 127 – Network layer command packet

Command packet identifier	command	User
0	Joining request	Routing device
1	Joining response	Gateway device
2	Communication status report request	Routing device
3	Leaving request	Routing device/ Gateway device
4	Leaving response	Gateway device/ Routing device
5	Cluster member report request	Routing device
6	Cluster member report response	Gateway device
7	Neighbor information report request	Routing device
8	Route adding request	Gateway device
9	Route adding response	Routing device
10	Route update request	Gateway device
11	Route update response	Routing device
12	Route deleting request	Gateway device
13	Route deleting response	Routing device
14	Link adding request	Gateway device/ Routing device
15	Link adding response	Routing device / Field device
16	Link update request	Gateway device/ Routing device
17	Link update response	Routing device / Field device
18	Link release request	Gateway device/ Routing device
19	Link release response	Routing device / Field device
20	Superframe adding request	Gateway device/ Routing device
21	Superframe adding response	Routing device / Field device
22	Superframe update request	Gateway device/ Routing device
23	Superframe update response	Routing device / Field device
24	Superframe release request	Gateway device/ Routing device
25	Superframe release response	Routing device / Field device
26	Device condition report request	Field device/Routing device
27	Channel condition report request	Field device/Routing device
28	Failure path report request	Routing device
29	Attribute getting request	Gateway device/ Routing device
30	Attribute getting response	Routing device / Field device
31	Attribute setting request	Gateway device/ Routing device
32	Attribute setting response	Routing device / Field device
33 to 255	Reserved	Reserved

The execution results of command are shown in Table 128.

Table 128 – Execution results of commands

Command implementing result	Identifier	Description
SUCCESS	0	Command execution succeeds
FAILURE	1	Command execution fails

9.6.4.2 Joining request and response

The joining request packet is used by a routing device to forward the joining request. The format of the joining request packet is shown in Table 129.

Table 129 – Format of joining request packet

Length in octet(s)	14/16	1	8	0/4	1
Field name	Network layer header	Command ID = 0	Physical address of the new device	Security material	Device type

The joining request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 0;

c) Physical address of the new device: this field is used to indicate the 64-bit physical address of the device that is joining the network;

d) Security material: this field has 0- or 4-octet length; if the authentication flag in the NL header is 0, the Security material is 0-octet long, otherwise, it is 4-octet long (see Clause 11 for the detailed information);

e) Device type: this field has 1-octet length and is used to indicate the type of the joining device. The value 0 indicates the gateway device, 1 indicates a routing device, 2 indicates a field device, and 3 indicates a handheld device; otherwise, this field is reserved.

The joining response packet is used to return the joining request result. The format of the joining response packet is shown in Table 130.

Table 130 – Format of joining response packet

Length in octet(s)	14/16	1	1	8	2
Field name	Network layer header	Command ID = 1	Execution result	Physical address of the new device	Short address of the new device

The joining response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 1;

c) Execution result: this field has 1-octet length. The value 0 indicates that the joining process is "*SUCCESS*"; otherwise, 1 indicates that the joining process is "FAILURE" (see Table 128);

d) Physical address of the new device: this field is used to indicate the 64-bit physical address of the device that is joining the network;

e) Short address of the new device: this field has 2-octet length. If the value of the Execution result field is 0, this field is valid and the value is the newly allocated 16-bit short address for the joining device; otherwise, this field is invalid.

9.6.4.3 Communication status report request

The communication status report request packet is used by a routing device to forward the communication status report request. The format of the communication status report request packet is shown in Table 131.

Table 131 – Format of communication status report request packet

Length in octet(s)	14/16	1	8	1	1
Field name	Network layer header	Command ID = 2	Physical address of the new device	Device type	Status

The communication status report request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 2;

c) Physical address of the new device: this field is used to indicate the 64-bit physical address of the device that has just joined the network;

d) Device type: this field has 1-octet length and is used to indicate the type of newly joined device. The value 0 indicates the gateway device, 1 indicates a routing device, 2 indicates a field device, and 3 indicates a handheld device; otherwise, this field is reserved;

e) Status: this field has 1-octet length and is used to indicate the result of the joining process; the value 0 indicates that the joining process is successful and 1 indicates that the joining process fails.

9.6.4.4 Leaving request and response

The leaving request packet is used either by a routing device to request departure from the gateway device or by the gateway device to request a routing device to leave the network. The format of the leaving request packet is shown in Table 132.

Table 132 – Format of leaving request packet

Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 3	Leaving reason

The leaving request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 3;

c) Leaving reason: this field is 1 octet, the valid value of which is shown in Table 133.

Table 133 – Value of leaving reason

Value	Description
0	Active leaving of routing device
1	Gateway device requests routing device to leave
2 to 255	Reserved

The leaving response packet is used to return the execution results. The format of the leaving response packet is shown in Table 134.

Table 134 – Format of leaving response packet

Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 4	Execution result

The leaving response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 4;

c) Execution result: this field has 1-octet length and indicates the result of the departure process. If the departure process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 128).

9.6.4.5 Cluster member report request and response

The cluster member report request packet is used by routing devices to report cluster member information to the gateway device. The format of the cluster member report request packet is described in Table 135.

Table 135 – Format of cluster member report request packet

Length in octet(s)	14/16	1	1	2
Field name	Network layer header	Command ID = 5	Cluster member modification flag	Cluster member address

The cluster member report request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 5;

c) Cluster member modification flag: this field has 1-octet length and indicates the kinds of modification; the value 0 indicates the adding action, and 1 indicates the deleting action; otherwise, this field is reserved (see Table 83);

d) Cluster member address: this field has 2-octet length and indicates the 16-bit address of the modified cluster member.

The cluster member report response packet is used to return the execution results of the cluster member report request command. The format of the cluster member report response packet is described in Table 136.

Table 136 – Format of cluster member report response packet

Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 6	Execution result

The cluster member report response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 6;

c) Execution result: this field has 1-octet length and indicates the result of the cluster member report process. If the report process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 128).

9.6.4.6 Neighbor information report request

The neighbor information report request packet is used by routing devices to report its one-hop neighbor information to the NM. The format of the neighbor information report request packet is described in Table 137.

Table 137 – Format of neighbor information report request packet

Length in octet(s)	14/16	1	1	Variable length
Field name	Network layer header	Command ID = 7	Number of neighbors	Items of neighbor table

The neighbor information report request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 andTable 126;

b) Command ID: this field has 1-octet length and the value of it is 7;

c) Number of neighbors: this field has 1-octet length and indicates the number of reported neighbors;

d) Items of neighbor table: this field has variable length; the structure of neighbor table item is shown in Table 14.

9.6.4.7 Route adding request and response

The route adding request packet is generated by the NM, and is sent to the destination routing device for adding a new routing record into its routing table.

The format of the route adding request packet is illustrated in Table 138.

Table 138 – Format of route adding request packet

Length in octet(s)	14/16	1	9
Field name	Network layer header	Command ID = 8	A record of routing table

The route adding request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 8;

c) A record of routing table: this field has 9-octet length; the detailed information is shown in Table 14.

The route adding response packet is used to return the execution results of the route adding request command.

The format of the route adding response packet is illustrated in Table 139.

Table 139 – Format of route adding response packet

Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 9	Execution result

The route adding response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 9;

c) Execution result: this field has 1-octet length and indicates the results of the route adding response process. If the response process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 128).

9.6.4.8 Route update request and response

The route update request packet is generated by the NM, and is sent to the destination routing device for modifying a routing table record in its routing table.

The format of the route update request packet is illustrated in Table 140.

Table 140 – Format of route update request packet

Length in octet(s)	14/16	1	9
Field name	Network layer header	Command ID = 10	A record of routing table

The route update request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 10;

c) A record of routing table: this field has 9-octet length; the detailed information is shown in Table 14.

The route update response packet is used to return the execution results of the route update request command.

The format of the route update response packet is illustrated in Table 141.

Table 141 – Format of route update response packet

Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 11	Execution result

The route update response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 11;

c) Execution result: this field has 1-octet length and indicates the result of the route update response process. If the response process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 128).

9.6.4.9 Route deleting request and response

The route deleting request packet is generated by the NM, and is sent to the destination routing device for deleting a routing record from its routing table.

The format of the route deleting request packet is illustrated as Table 142.

Table 142 – Format of route deleting request packet

Length in octet(s)	14/16	1	2
Field name	Network layer header	Command ID = 12	Route ID

The route update request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 12;

c) Route ID: this field has 2-octet length and indicates the identifier of the deleted route.

The route deleting response packet is used to return the execution result of the Route deleting request command.

The format of the route deleting response packet is illustrated as Table 143.

Table 143 – Format of route deleting response packet

Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 13	Execution result

The route update response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 13;

c) Execution result: this field has 1-octet length and indicates the result of the route deleting response process. If the response process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 128).

9.6.4.10 Link adding request and response

The link adding request packet is used by the NM to add one or more new link(s) to a routing device, and is also used by the routing device to add one or more new link(s) to a field device. After receiving this command packet, the routing device or field device should add a record to its link table.

The format of the link adding request packet is shown in Table 144.

Table 144 – Format of link adding request packet

Length in octet(s)	14/16	1	2	12	…	12
Field name	Network layer header	Command ID = 14	Number of links	Link table item 1	…	Link table item N

The link adding request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 14;

c) Number of links: this field has 2-octet length and indicates the number of added links;

d) Link table item 1 to N (Number of links): each of these link table items has 12-octet length, which indicates the detailed information of the added link. The structure of the link table item is shown in Table 16.

The link adding response packet is used to return the execution results of link adding. The format of the link adding response packet is shown in Table 145.

Table 145 – Format of link adding response packet

Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 15	Execution result

The link adding response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 15;

c) Execution result: this field has 1-octet length and indicates the result of the link adding response process. If the response process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 128).

9.6.4.11 Link update request and response

The link update request packet is used by the NM to update one or more existing link(s) a routing device, and is also used by the routing device to update one or more existing link(s) to a field device. After receiving this command packet, the routing device or field device should update a record in its link table.

The format of the link update request packet is shown in Table 146.

Table 146 – Format of link update request packet

Length in octet(s)	14/16	1	2	12	…	12
Field name	Network layer header	Command ID = 16	Number of links	Link table item 1	…	Link table item N

The link update request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 16;

c) Number of links: this field has 2-octet length and indicates the number of updated links;

d) Link table item 1 to N (N = Number of links): each of these link table items has 12-octet length, which indicates the detailed information of the updated links. The structure of the link table item is shown in Table 16.

The link update response packet is used to return the execution results of link update. The format of the link update response packet is shown in Table 147.

Table 147 – Format of link update response packet

Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 17	Execution result

The route update response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 17;

c) Execution result: this field has 1-octet length and indicates the result of the link update response process. If the response process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 128).

9.6.4.12 Link release request and response

The link release request packet is used by the NM to release one or more existing link(s) to a routing device, and is also used by the routing device to release one or more existing link(s) to a field device. After receiving this command packet, the routing device or field device should release a record in its link table.

The format of the link release request packet is shown in Table 148.

Table 148 – Format of link release request packet

Length in octet(s)	14/16	1	1	2	…	2
Field name	Network layer header	Command ID = 18	Number of links	Link ID 1	…	Link ID N

The link release request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 18;

c) Number of links: this field has 2-octet length and indicates the number of released links;

d) Link ID 1 to N (N = Number of links): each Link ID has2-octet length, which indicates the identifiers of the released links.

The link release response packet is used to return the execution results of link release. The format of the link release response packet is shown in Table 149.

Table 149 – Format of link release response packet

Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 19	Execution result

The link release response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 19;

c) Execution result: this field has 1-octet length and indicates the result of the link release response process. If the response process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 128).

9.6.4.13 Superframe adding request and response

The superframe adding request packet is used by the NM to add a new superframe to a routing device, and is also used by the routing device to add a new superframe to a field device. After receiving this command packet, the routing device or field device should add a record in its superframe table.

The format of the superframe adding request packet is shown in Table 150.

Table 150 – Format of superframe adding request packet

Length in octet(s)	14/16	1	12
Field name	Network layer header	Command ID = 20	Superframe table item

The superframe adding request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 20;

c) Superframe table item: this field has 12-octet length and indicates the detailed information of the added superframe (see Table 15).

The superframe adding response packet is used to return the execution results of superframe adding. The format of the superframe adding response packet is shown in Table 151.

Table 151 – Format of superframe adding response packet

Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 21	Execution result

The superframe adding response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 21;

c) Execution result: this field has 1-octet length and indicates the result of the superframe adding response process. If the response process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 128).

9.6.4.14 Superframe update request and response

The superframe update request packet is used by the NM to update an existing superframe to a routing device, and is also used by the routing device to update an existing superframe to a field device. After receiving this command packet, the routing device or field device should update a record in its superframe table.

The format of the superframe update request packet is shown in Table 152.

Table 152 – Format of superframe update request packet

Length in octet(s)	14/16	1	12
Field name	Network layer header	Command ID = 22	Superframe table item

The superframe update request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 22;

c) Superframe table item: this field has 12-octet length and indicates the detailed information of the updated superframe (see Table 15).

The superframe update response packet is used to return the execution results of superframe update. The format of the superframe update response packet is shown in Table 153.

Table 153 – Format of superframe update response packet

Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 23	Execution result

The superframe update response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 23;

c) Execution result: this field has 1-octet length and indicates the result of the superframe update response process. If the response process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 128).

9.6.4.15 Superframe release request and response

The superframe release request command frame is used by the NM to release an existing superframe to a routing device, and is also used by the routing device to release an existing superframe to a field device. After receiving this command packet, the routing device or field device should release a record in its superframe table.

The format of the superframe release request packet is shown in Table 154.

Table 154 – Format of superframe release request packet

Length in octet(s)	14/16	1	2
Field name	Network layer header	Command ID = 24	Superframe ID

The superframe release request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 24;

c) Superframe ID: this field has 2-octet length and indicates the identifier of the superframe to be the released (see Table 15).

The superframe release response packet is used to return the execution results of superframe release. The format of the superframe release response packet is shown in Table 155.

Table 155 – Format of superframe release response packet

Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 25	Execution result

The superframe release response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 25;

c) Execution result: this field has 1-octet length and indicates the result of the superframe release response process. If the response process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 128).

9.6.4.16 Device condition report request

The device condition report request packet is used to report the conditions of the devices remotely. The format of the device condition report request packet is described in Table 156.

Table 156 – Format of device condition report request packet

Length in octet(s)	14/16	1	1	Variable length
Field name	Network layer header	Command ID = 26	Number of devices	Items of device condition information

The device condition report request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 26;

c) Number of devices: this field has 1-octet length and indicates the number of reported devices.

d) Items of device condition information: the detailed format of the device condition information field is described in Table 157. This field has N×11 octet length, where N = Number of devices.

Table 157 – Format of device condition information

Length in octet(s)	2	2	2	2	1	2
Field name	Device short address	Number of packets transmitted after last report	Number of packets terminated in this device after last report	Number of packets with MAC MIC failure after last report	Battery level	Restart count of device after last report

The device condition information field includes the following subfields (see Table 21):

a) Device short address: this subfield has 2-octet length and indicates the short address of the reported device;

b) Number of packets transmitted after last report: this subfield has 2-octet length and indicates the number of transmitted packets since the last report of the reported device;

c) Number of packets terminated in this device after last report: this subfield has 2-octet length and indicates the number of received packets since the last report of the reported device;

d) Number of packets with MAC MIC failure after last report: this subfield has 2-octet length and indicates the number of packets with MAC MIC failure since the last report;

e) Battery level: this subfield has 1-octet length and indicates the residual power level;

f) Restart count of device after last report: this subfield has 2-octet length and indicates the number of restarts since the last report.

9.6.4.17 Channel condition report request

The channel condition report request packet is used to report the quality status of the communication channels remotely. The format of the channel condition report request packet is described in Table 158.

Table 158 – Format of channel condition report request packet

Length in octet(s)	14/16	1	1	Variable length
Field name	Network layer header	Command ID = 27	Number of channels	Items of channel quality information

The channel condition report request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 27;

c) Number of channels: this field has 1-octet length and indicates the number of reported channels;

d) Channel quality information: this field has variable length;

e) Items of channel quality information: the detailed format of the channel quality information is shown in Table 159. This field has N×5 octet length, where N = Number of channels.

Table 159 – Format of channel quality information

Length in octet(s)	1	1	2	1
Field name	Channel index	Link quality	Packet loss rate	Number of retries

The channel quality information field includes the following subfields:

a) Channel index: this field has 1-octet length and indicates the sequence number of channels;

b) Link quality: this field has 1-octet length and indicates the LQI value of each channels;

c) Packet loss rate: this field has 2-octet length and indicates the packet loss rate of each channel;

d) Number of retries: this field has 1-octet length and indicates the number of retransmissions of each channel.

9.6.4.18 Path failure report request

The path failure report request packet is used by routing devices to report the failure of a path. The format of the path failure report request packet is described in Table 160.

Table 160 – Format of path failure report request packet

Length in octet(s)	14/16	1	2
Field name	Network layer header	Command ID = 28	Route ID

The path failure report request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 28;

c) Route ID: this field has 1-octet length and indicates the identifier of the reported route.

9.6.4.19 Attribute getting request

The attribute getting request packet is used to ask a specified device to report the attributes of its MIB. The format of the attribute getting request packet is described in Table 161.

Table 161 – Format of attribute getting request packet

Length in octet(s)	14/16	1	1	1	2	1
Field name	Network layer header	Command ID = 29	Attribute ID	Attribute member ID	First storage index of multiple attribute values	Number of attributes

The attribute getting request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Figure 70 and Table 126;

b) Command ID: this field has 1-octet length and the value of it is 29;

c) Attribute ID: this field has 1-octet length and indicates the index of the requested attribute;

d) Attributed member ID: this field has 1-octet length and indicates the index of the requested attribute member;

e) First storage index of multiple attribute values: this field has 2-octet length and indicates the first storage index of multiple attribute values;

f) Number of attributes: this field has 1-octet length and indicates either the number of attribute values or the number of attribute member values.

9.6.4.20 Attribute getting response

The attribute getting response packet is used to respond to the attribute getting request, which should return the values of the attributes. The format of the attribute getting response packet is described in Table 162.