BS EN ISO 16484-5:2022

BSI Standards Publication

Building automation and control systems (BACS)

Part 5: Data communication protocol

Documents
Document Sharing Hub

Full document details: BS EN ISO 16484-5:2022

National foreword

This British Standard is the UK implementation of EN ISO 16484-5:2022. It is identical to ISO 16484-5:2022. It supersedes BS EN ISO 16484-5:2017+A1:2020, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee RHE/2, Ventilation for buildings, heating and hot water services.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2022
Published by BSI Standards Limited 2022

ISBN 978 0 539 22433 7

ICS 35.240.67; 91.040.01

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 November 2022.

Amendments/corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS EN ISO 16484-5:2022

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN ISO 16484-5

October 2022

ICS 91.040.01; 35.240.67

Supersedes EN ISO 16484-5:2017, EN ISO 16484-5:2017/A1:2020

English Version

Building automation and control systems (BACS) - Part 5: Data communication protocol

Systèmes d'automatisation et de gestion technique du bâtiment - Partie 5: Protocole de communication de données

Systeme der Gebäudeautomation - Teil 5: Datenkommunikationsprotokoll

This European Standard was approved by CEN on 4 September 2022.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and United Kingdom.

EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

© 2022 CEN All rights of exploitation in any form and by any means reserved worldwide for CEN national Members.

Ref. No. EN ISO 16484-5:2022 E

Documents
Document Sharing Hub

Full document details: BS EN ISO 16484-5:2022

Contents

Page

European foreword **3**

Documents
Document Sharing Hub

Full document details: BS EN ISO 16484-5:2022

European foreword

This document (EN ISO 16484-5:2022) has been prepared by Technical Committee ISO/TC 205 "Building environment design" in collaboration with Technical Committee CEN/TC 247 “Building Automation, Controls and Building Management” the secretariat of which is held by SNV.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by April 2023, and conflicting national standards shall be withdrawn at the latest by April 2023.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN shall not be held responsible for identifying any or all such patent rights.

This document supersedes EN ISO 16484-5:2017.

Any feedback and questions on this document should be directed to the users’ national standards body/national committee. A complete listing of these bodies can be found on the CEN website.

According to the CEN-CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and the United Kingdom.

Endorsement notice

The text of ISO 16484-5:2022 has been approved by CEN as EN ISO 16484-5:2022 without any modification.

Documents
Document Sharing Hub

Full document details: BS EN ISO 16484-5:2022

INTERNATIONAL STANDARD

ISO 16484-5

Seventh edition
2022-09

Building automation and control systems (BACS) —

Part 5: Data communication protocol

Systèmes d'automatisation et de gestion technique du bâtiment — Partie 5: Protocole de communication de données

Reference number
ISO 16484-5:2022(E)

Documents
Document Sharing Hub
© ISO 2022

Full document details: BS EN ISO 16484-5:2022

COPYRIGHT PROTECTED DOCUMENT

© ISO 2022

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS EN ISO 16484-5:2022

CONTENTS

Page

FOREWORD xii
1 PURPOSE 1
2 SCOPE 1
3 DEFINITIONS 1
3.1 Terms Adopted from International Standards 1
3.2 Terms Defined for this Standard 3
3.3 Abbreviations and Acronyms Used in this Standard 7
4 BACnet PROTOCOL ARCHITECTURE 11
4.1 The BACnet Collapsed Architecture 12
4.2 BACnet Network Topology 14
4.3 Security 14
5 THE APPLICATION LAYER 16
5.1 The Application Layer Model 16
5.2 Segmentation of BACnet Messages 20
5.3 Transmission of BACnet APDUs 21
5.4 Application Protocol State Machines 25
5.5 Application Protocol Time Sequence Diagrams 41
5.6 Application Layer Service Conventions 50
6 THE NETWORK LAYER 52
6.1 Network Layer Service Specification 52
6.2 Network Layer PDU Structure 54
6.3 Messages for Multiple Recipients 59
6.4 Network Layer Protocol Messages 60
6.5 Network Layer Procedures 64
6.6 BACnet Routers 66
6.7 Point-To-Point Half-Routers 71
7 DATA LINK/PHYSICAL LAYERS: Ethernet (ISO 8802-3) LAN 76
7.1 The Use of ISO 8802-2 Logical Link Control (LLC) 76
7.2 Parameters Required by the LLC Primitives 76
7.3 Parameters Required by the MAC Primitives 76
7.4 Physical Media 76
8 DATA LINK/PHYSICAL LAYERS: ARCNET (ATA 878.1) LAN 77
8.1 The Use of ISO 8802-2 Logical Link Control (LLC) 77
8.2 Parameters Required by the LLC Primitives 77
8.3 Mapping the LLC Services to the ARCNET MAC Layer 77
8.4 Parameters Required by the MAC Primitives 77
8.5 Physical Media 77
9 DATA LINK/PHYSICAL LAYERS: MASTER-SLAVE/TOKEN-PASSING (MS/TP) LAN 79
9.1 Service Specification 79
9.2 Physical Layer 81
9.3 MS/TP Frame Format 90
9.4 Overview of the MS/TP Network 92
9.5 MS/TP Medium Access Control 92
9.6 Cyclic Redundancy Check (CRC) 111
9.7 Interfacing MS/TP LANs with Other BACnet LANs 112
9.8 Responding BACnet User Processing of Messages from MS/TP 112
9.9 Repeaters 113
9.10 COBS (Consistent Overhead Byte Stuffing) Encoding 114
9.11 Documenting MS/TP Device Design Choices 118
10 DATA LINK/PHYSICAL LAYERS: POINT-TO-POINT (PTP) 119
10.1 Overview 119
10.2 Service Specification 119
10.3 Point-to-Point Frame Format 124
10.4 PTP Medium Access Control Protocol 126
11 DATA LINK/PHYSICAL LAYERS: LonTalk (ISO/IEC 14908.1) LAN 147
11.1 The Use of ISO 8802-2 Logical Link Control (LLC) 147
11.2 Parameters Required by the LLC Primitives 147

AVA Documents
Document Sharing Hub

© ISO 2022 – All rights reserved

Full document details: BS EN ISO 16484-5:2022

11.3 Mapping the LLC Services to the LonTalk Application Layer ... 147
11.4 Parameters Required by the Application Layer Primitives ... 147
11.5 Physical Media ... 148
12 MODELING CONTROL DEVICES AS A COLLECTION OF OBJECTS ... 149
12.1 Object Characteristics and Requirements ... 149
12.2 Analog Input Object Type ... 155
12.3 Analog Output Object Type ... 162
12.4 Analog Value Object Type ... 169
12.5 Averaging Object Type ... 177
12.6 Binary Input Object Type ... 181
12.7 Binary Output Object Type ... 188
12.8 Binary Value Object Type ... 197
12.9 Calendar Object Type ... 205
12.10 Command Object Type ... 208
12.11 Device Object Type ... 214
12.12 Event Enrollment Object Type ... 226
12.13 File Object Type ... 234
12.14 Group Object Type ... 238
12.15 Life Safety Point Object Type ... 241
12.16 Life Safety Zone Object Type ... 248
12.17 Loop Object Type ... 255
12.18 Multi-state Input Object Type ... 265
12.19 Multi-state Output Object Type ... 271
12.20 Multi-state Value Object Type ... 278
12.21 Notification Class Object Type ... 285
12.22 Program Object Type ... 291
12.23 Pulse Converter Object Type ... 298
12.24 Schedule Object Type ... 306
12.25 Trend Log Object Type ... 313
12.26 Access Door Object Type ... 323
12.27 Event Log Object Type ... 332
12.28 Load Control Object Type ... 340
12.29 Structured View Object Type ... 350
12.30 Trend Log Multiple Object Type ... 355
12.31 Access Point Object Type ... 365
12.32 Access Zone Object Type ... 382
12.33 Access User Object Type ... 390
12.34 Access Rights Object Type ... 394
12.35 Access Credential Object Type ... 400
12.36 Credential Data Input Object Type ... 410
12.37 CharacterString Value Object Type ... 416
12.38 DateTime Value Object Type ... 423
12.39 Large Analog Value Object Type ... 429
12.40 BitString Value Object Type ... 437
12.41 OctetString Value Object Type ... 444
12.42 Time Value Object Type ... 448
12.43 Integer Value Object Type ... 454
12.44 Positive Integer Value Object Type ... 462
12.45 Date Value Object Type ... 470
12.46 DateTime Pattern Value Object Type ... 476
12.47 Time Pattern Value Object Type ... 482
12.48 Date Pattern Value Object Type ... 488
12.49 Deleted Clause ... 494
12.50 Global Group Object Type ... 495
12.51 Notification Forwarder Object Type ... 502
12.52 Alert Enrollment Object Type ... 510
12.53 Channel Object Type ... 514
12.54 Lighting Output Object Type ... 524
12.55 Binary Lighting Output Object Type ... 539
12.56 Network Port Object Type ... 549

© ISO 2022 – All rights reserved
AVA Documents Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

12.57 Timer Object Type ... 573
12.58 Elevator Group Object Type ... 585
12.59 Lift Object Type ... 589
12.60 Escalator Object Type ... 601
12.61 Accumulator Object Type ... 608
12.62 Staging Object Type ... 618
12.63 Audit Reporter Object Type ... 628
12.64 Audit Log Object Type ... 633
13 ALARM AND EVENT SERVICES ... 640
13.1 Change of Value Reporting ... 640
13.2 Event Reporting ... 645
13.3 Event Algorithms ... 656
13.4 Fault Algorithms ... 685
13.5 AcknowledgeAlarm Service ... 692
13.6 ConfirmedCOVNotification Service ... 694
13.7 UnconfirmedCOVNotification Service ... 696
13.8 ConfirmedEventNotification Service ... 697
13.9 UnconfirmedEventNotification Service ... 700
13.10 GetAlarmSummary Service ... 703
13.11 GetEnrollmentSummary Service ... 705
13.12 GetEventInformation Service ... 708
13.13 LifeSafetyOperation Service ... 710
13.14 SubscribeCOV Service ... 712
13.15 SubscribeCOVProperty Service ... 715
13.16 SubscribeCOVPropertyMultiple Service ... 718
13.17 ConfirmedCOVNotificationMultiple Service ... 723
13.18 UnconfirmedCOVNotificationMultiple Service ... 726
13.19 AuditLogQuery ... 728
13.20 ConfirmedAuditNotification ... 732
13.21 UnconfirmedAuditNotification ... 733
14 FILE ACCESS SERVICES ... 734
14.1 AtomicReadFile Service ... 734
14.2 AtomicWriteFile Service ... 737
15 OBJECT ACCESS SERVICES ... 740
15.1 AddListElement Service ... 740
15.2 RemoveListElement Service ... 742
15.3 CreateObject Service ... 744
15.4 DeleteObject Service ... 747
15.5 ReadProperty Service ... 748
15.6 Deleted Clause ... 750
15.7 ReadPropertyMultiple Service ... 751
15.8 ReadRange Service ... 754
15.9 WriteProperty Service ... 762
15.10 WritePropertyMultiple Service ... 764
15.11 WriteGroup Service ... 767
16 REMOTE DEVICE MANAGEMENT SERVICES ... 769
16.1 DeviceCommunicationControl Service ... 769
16.2 ConfirmedPrivateTransfer Service ... 771
16.3 UnconfirmedPrivateTransfer Service ... 773
16.4 ReinitializeDevice Service ... 774
16.5 ConfirmedTextMessage Service ... 776
16.6 UnconfirmedTextMessage Service ... 778
16.7 TimeSynchronization Service ... 779
16.8 UTCTimeSynchronization Service ... 780
16.9 Who-Has and I-Have Services ... 781
16.10 Who-Is and I-Am Services ... 783
16.11 Who-Am-I and You-Are Services ... 785
17 VIRTUAL TERMINAL SERVICES ... 787
17.1 Virtual Terminal Model ... 787
17.2 VT-Open Service ... 791

AVA Documents
Document Sharing Hub

© ISO 2022 – All rights reserved

Full document details: BS EN ISO 16484-5:2022

17.3 VT-Close Service ... 793
17.4 VT-Data Service ... 794
17.5 Default Terminal Characteristics ... 796
18 ERROR, REJECT, and ABORT CODES ... 800
18.1 Error Class - DEVICE ... 800
18.2 Error Class - OBJECT ... 800
18.3 Error Class - PROPERTY ... 801
18.4 Error Class - RESOURCES ... 802
18.5 Error Class - SECURITY ... 803
18.6 Error Class - SERVICES ... 804
18.7 Error Class - COMMUNICATION ... 805
18.8 Error Class - VT ... 809
18.9 Reject Reason ... 810
18.10 Abort Reason ... 810
18.11 Confirmed Service Common Errors ... 811
19 BACnet PROCEDURES ... 812
19.1 Backup and Restore ... 812
19.2 Command Prioritization ... 817
19.3 Device Restart Procedure ... 822
19.4 Determining Maximum Conveyable APDU ... 823
19.5 Value Source Mechanism ... 825
19.6 Audit Logging ... 827
19.7 Unconfigured Device Discovery and Address Assignment ... 838
20 ENCODING BACnet PROTOCOL DATA UNITS ... 839
20.1 Encoding the Fixed Part of BACnet APDUs ... 840
20.2 Encoding the Variable Part of BACnet APDUs ... 851
21 FORMAL DESCRIPTION OF APPLICATION PROTOCOL DATA UNITS ... 866
21.1 APDU Definitions ... 866
21.2 Confirmed Service Productions ... 868
21.3 Unconfirmed Service Productions ... 878
21.4 Error Productions ... 881
21.5 Application Types ... 891
21.6 Base Types ... 892
22 CONFORMANCE AND INTEROPERABILITY ... 960
22.1 Conformance to BACnet ... 960
22.2 BACnet Interoperability ... 961
23 EXTENDING BACnet TO ACCOMMODATE VENDOR PROPRIETARY INFORMATION ... 963
23.1 Extending Enumeration Values ... 963
23.2 Using the PrivateTransfer Services to Invoke Non-Standardized Services ... 964
23.3 Adding Proprietary Properties to a Standardized Object ... 964
23.4 Adding Proprietary Object Types to BACnet ... 965
23.5 Restrictions on Extending BACnet ... 965
24 DELETED CLAUSE ... 966
25 REFERENCES ... 967
ANNEX A - PROTOCOL IMPLEMENTATION CONFORMANCE STATEMENT (NORMATIVE) ... 971
ANNEX B - GUIDE TO SPECIFYING BACnet DEVICES (INFORMATIVE) ... 974
ANNEX C - Removed ... 975
ANNEX D - Removed ... 976
ANNEX E - EXAMPLES OF BACnet APPLICATION SERVICES (INFORMATIVE) ... 977
E.1 Alarm and Event Services ... 977
E.2 File Access Services ... 981
E.3 Object Access Services ... 983
E.4 Remote Device Management Services ... 989
E.5 Virtual Terminal Services ... 993
ANNEX F - EXAMPLES OF APDU ENCODING (INFORMATIVE) ... 995
F.1 Example Encodings for Alarm and Event Services ... 995
F.2 Example Encodings for File Access Services ... 1006
F.3 Example Encodings for Object Access Services ... 1008
F.4 Example Encodings for Remote Device Management Services ... 1018
F.5 Example Encodings for Virtual Terminal Services ... 1024

AVA Documents
Document Sharing Hub

© ISO 2022 – All rights reserved

Full document details: BS EN ISO 16484-5:2022

ANNEX G - CALCULATION OF CRC (INFORMATIVE) 1026
G.1 Calculation of the Header CRC 1026
G.2 Calculation of the Data CRC 1032
G.3 Calculation of the Encoded CRC-32K 1036
ANNEX H - COMBINING BACnet NETWORKS WITH NON-BACnet NETWORKS (NORMATIVE) 1040
H.1 BACnet Gateways 1040
H.2 Requirements and Best Practices for BACnet Gateway Implementations 1040
H.3 Using BACnet with the DARPA Internet Protocols 1042
H.4 Using BACnet with the IPX Protocol 1044
H.5 Using BACnet with EIB/KNX 1045
H.6 Using BACnet with the Former BACnet/WS Web Services Interface Defined by Annex N 1055
H.7 Virtual MAC Addressing 1057
ANNEX I - COMMANDABLE PROPERTIES WITH MINIMUM ON AND OFF TIMES (INFORMATIVE) 1059
ANNEX J - BACnet/IP (NORMATIVE) 1061
J.1 General 1061
J.2 BACnet Virtual Link Layer 1062
J.3 BACnet/IP Directed Messages 1065
J.4 BACnet/IP Broadcast Messages 1065
J.5 Addition of Foreign B/IP Devices to an Existing B/IP Network 1068
J.6 Routing Between B/IP and non-B/IP BACnet Networks 1069
J.7 Routing Between Two B/IP BACnet Networks 1070
J.8 Use of IP Multicast within BACnet/IP 1075
ANNEX K - BACnet INTEROPERABILITY BUILDING BLOCKS (BIBBs) (NORMATIVE) 1077
K.1 Data Sharing BIBBs 1077
K.2 Alarm and Event Management BIBBs 1098
K.3 Scheduling BIBBs 1111
K.4 Trending BIBBs 1115
K.5 Device Management BIBBs 1118
K.6 Network Management BIBBs 1125
K.7 Gateway BIBBs 1128
K.8 Audit Reporting BIBBs 1129
ANNEX L - DESCRIPTIONS AND PROFILES OF STANDARDIZED BACnet DEVICES (NORMATIVE) 1131
L.1 Operator Interface Profiles 1131
L.2 Life Safety Operator Interface Profiles 1134
L.3 Access Control Operator Interface Profiles 1136
L.4 Controller Profiles 1139
L.5 Life Safety Controller Profiles 1142
L.6 Access Control Controller Profiles 1143
L.7 Miscellaneous Profiles 1145
L.8 BACnet General (B-GENERAL) Profile 1148
L.9 Lighting Operator Interface Profiles 1149
L.10 Lighting Control Station Profiles 1151
L.11 Lighting Controller Profiles 1152
L.12 Elevator Operator Interface Profiles 1154
L.13 Elevator Controller Profiles 1156
ANNEX M - GUIDE TO EVENT NOTIFICATION PRIORITY ASSIGNMENTS (INFORMATIVE) 1159
M.1 Life Safety Message Group (0 - 31) 1159
M.2 Property Safety Message Group (32 - 63) 1160
M.3 Supervisory Message Group (64 - 95) 1160
M.4 Trouble Message Group (96 - 127) 1161
M.5 Miscellaneous Higher Priority Message Group (128 - 191) 1162
M.6 Miscellaneous Lower Priority Message Group (192 - 255) 1162
ANNEX N - FORMER BACnet/WS WEB SERVICES INTERFACE (INFORMATIVE) 1163
N.1 Data Model 1163
N.2 Paths 1164
N.3 Normalized Points 1164
N.4 Reference Nodes 1165
N.5 Localization 1165
N.6 Security 1165
N.7 Sessions 1166

AVA Documents
Document Sharing Hub

© ISO 2022 – All rights reserved

Full document details: BS EN ISO 16484-5:2022

N.8 Attributes 1166
N.9 Standard Nodes 1172
N.10 Encodings 1172
N.11 Service Options 1173
N.12 Services 1176
N.13 Errors 1193
N.14 Extending BACnet/WS 1194
ANNEX O - BACnet OVER ZigBee AS A DATA LINK LAYER (NORMATIVE) 1195
O.1 General 1195
O.2 ZigBee Overview 1195
O.3 Definitions 1196
O.4 Unicast Addressing 1196
O.5 Broadcast Addressing 1196
O.6 BACnet/ZigBee Data Link Layer (BZLL) 1197
O.7 Maximum Payload Size 1200
O.8 Vendor Specific Commands 1200
ANNEX P - BACnet ENCODING OF STANDARD AUTHENTICATION FACTOR FORMATS (NORMATIVE) 1201
ANNEX Q - XML DATA FORMATS (NORMATIVE) 1208
Q.1 Introduction 1208
Q.2 XML Document Structure 1211
Q.3 Expressing Data 1215
Q.5 Expressing Values 1216
Q.6 Binary Encoding and Access Rules 1218
Q.7 Extensibility 1218
Q.8 BACnet URI Scheme 1219
ANNEX R - MAPPING NETWORK LAYER ERRORS (NORMATIVE) 1220
ANNEX S - Removed 1221
ANNEX T - COBS (CONSISTENT OVERHEAD BYTE STUFFING) FUNCTIONS (INFORMATIVE) 1222
T.1 Preparing a COBS-Encoded MS/TP Frame for Transmission 1222
T.2 Decoding an Extended MS/TP Frame upon Reception 1224
T.3 Example COBS-Encoded Frame - Who-Has Service 1226
ANNEX U - BACnet/IPv6 (NORMATIVE) 1228
U.1 General 1228
U.2 BACnet/IPv6 BACnet Virtual Link Layer 1229
U.3 BACnet/IPv6 Directed Messages 1233
U.4 BACnet/IPv6 Broadcast Messages 1233
U.5 BACnet /IPv6 VMAC Table Management 1238
ANNEX V - MIGRATION FROM SOAP SERVICES (INFORMATIVE) 1239
V.1 Services 1239
V.2 Service Options 1241
ANNEX W - BACnet/WS RESTful WEB SERVICES INTERFACE (NORMATIVE) 1242
W.1 Data Model 1242
W.2 Paths 1242
W.3 Security 1243
W.4 Sessions 1252
W.5 Standard Data Items 1252
W.6 Metadata 1257
W.7 Functions 1257
W.8 Query Parameters 1259
W.9 Representation of Data 1261
W.10 Representation of Metadata 1261
W.11 Representation of Logs 1262
W.12 Filtering Items 1266
W.13 Limiting Number of Items 1268
W.14 Selecting Children 1268
W.15 Controlling Content of Data Representations 1269
W.16 Specifying Ranges 1272
W.17 Localized Values 1274
W.18 Accessing Individual Tags and Bits 1275
W.19 Semantics 1275

AVA Documents
Document Sharing Hub

© ISO 2022 – All rights reserved

Full document details: BS EN ISO 16484-5:2022

W.20 Links and Relationships ... 1275
W.21 Foreign XML and Other Media Types ... 1275
W.22 Logical Modeling ... 1276
W.23 Mapped Modeling ... 1277
W.24 Commandability ... 1277
W.25 Writability and Visibility ... 1277
W.26 Working with Optional Data ... 1279
W.27 Working with Optional Metadata ... 1279
W.28 Creating Data ... 1280
W.29 Setting Data ... 1280
W.30 Deleting Data ... 1282
W.31 Parentally Inherited Values ... 1283
W.32 Concurrency Control ... 1283
W.33 Server Support for Data Definitions ... 1283
W.34 Server Support for Metadata ... 1284
W.35 Client Implementation Guidelines ... 1285
W.36 Subscriptions ... 1285
W.37 Reading Multiple Resources ... 1287
W.38 Writing Multiple Resources ... 1288
W.39 Mapping of BACnet Systems ... 1288
W.40 Errors ... 1292
W.41 Examples ... 1294
ANNEX X - EXTENDED DISCOVERY OF DEVICES, PROFILES, AND VIEWS (NORMATIVE) ... 1321
X.1 Profiles ... 1321
X.2 xdd Files ... 1322
X.3 Example of Definition of Objects, Properties, and Datatypes. ... 1323
X.4 Views ... 1324
X.5 PICS Declarations ... 1330
ANNEX Y - ABSTRACT DATA MODEL (NORMATIVE) ... 1331
Y.1 Model Components ... 1331
Y.2 Trees ... 1333
Y.3 Base Types ... 1335
Y.4 Common Metadata ... 1335
Y.5 Named Values ... 1349
Y.6 Named Bits ... 1352
Y.7 Primitive Values ... 1353
Y.8 Range Restrictions ... 1355
Y.9 Engineering Units ... 1356
Y.10 Length Restrictions ... 1357
Y.11 Collections ... 1359
Y.12 Primitive Data ... 1360
Y.13 Constructed Data ... 1364
Y.14 Data of Undefined Type ... 1368
Y.15 Logical Modeling ... 1368
Y.16 Links ... 1368
Y.17 Change Indications ... 1370
Y.18 Definitions, Types, Instances, and Inheritance ... 1370
Y.19 Data Revisions ... 1376
Y.20 BACnet-Specific Base Types ... 1379
Y.21 BACnet-Specific Metadata ... 1380
ANNEX Z - JSON DATA FORMATS (NORMATIVE) ... 1384
Z.1 Introduction ... 1384
Z.2 JSON Document Structure ... 1387
Z.3 Expressing Data ... 1390
Z.4 Expressing Metadata ... 1390
Z.5 Expressing Values ... 1392
Z.6 Extensibility ... 1393
ANNEX AA – TIME SERIES DATA EXCHANGE FILE FORMAT (NORMATIVE) ... 1395
AA.1 File Format ... 1395
AA.2 Representation of Data ... 1395

AVA Documents
Document Sharing Hub

© ISO 2022 – All rights reserved

Full document details: BS EN ISO 16484-5:2022

AA.3 File Generation 1396
AA.4 Example Files 1397
ANNEX AB – BACnet Secure Connect (NORMATIVE) 1398
AB.1 BACnet Secure Connect Data link 1398
AB.2 BACnet/SC Virtual Link Layer Messages 1404
AB.3 BACnet/SC Node Operation 1414
AB.4 Node Switch and Direct Connections 1416
AB.5 Hub Function and Hub Connector 1419
AB.6 BACnet/SC Connections 1422
AB.7 Application of WebSockets in BACnet/SC 1426
HISTORY OF REVISIONS 1432

NOTE

Approved addenda, errata, or interpretations for this standard can be downloaded free of charge from the ASHRAE Web site at www.ashrae.org/technology.

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 205, *Building environment design*, in collaboration with the European Committee for Standardization (CEN) Technical Committee CEN/TC 247, *Building automation, controls and building management,* in accordance with the Agreement on technical cooperation between ISO and CEN (Vienna Agreement) and with the American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (ASHRAE).

This seventh edition cancels and replaces the sixth edition (ISO 16484-5:2017), which has been technically revised. It also incorporates the Amendment ISO 16484-5:2017/Amd 1:2020.

The main changes are as follows:

- See the detailed list of changes on pages 1432 to 1451.

A list of all parts in the ISO 16484 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

FOREWORD

BACnet, the ASHRAE building automation and control networking protocol, has been designed specifically to meet the communication needs of building automation and control systems for applications such as heating, ventilating, and air-conditioning control, lighting control, access control, and fire detection systems. The BACnet protocol provides mechanisms by which computerized equipment of arbitrary function may exchange information, regardless of the particular building service it performs. As a result, the BACnet protocol may be used by head-end computers, general-purpose direct digital controllers, and application specific or unitary controllers with equal effect.

The motivation for this Standard was the widespread desire of building owners and operators for "interoperability," the ability to integrate equipment from different vendors into a coherent automation and control system - and to do so competitively. To accomplish this, the Standard Project Committee (SPC) solicited and received input from dozens of interested firms and individuals; reviewed all relevant national and international data communications standards, whether de facto or the result of committee activity; and spent countless hours in debate and discussion of the pros and cons of each element of the protocol.

What has emerged from the committee deliberations is a network protocol model with these principal characteristics:

(a) All network devices (except MS/TP slaves) are peers, but certain peers may have greater privileges and responsibilities than others.

(b) Each network device is modeled as a collection of network-accessible, named entities called "objects." Each object is characterized by a set of attributes or "properties." While this Standard prescribes the most widely applicable object types and their properties, implementors are free to create additional object types if desired. Because the object model can be easily extended, it provides a way for BACnet to evolve in a backward compatible manner as the technology and building needs change.

(c) Communication is accomplished by reading and writing the properties of particular objects and by the mutually acceptable execution of other protocol "services." While this Standard prescribes a comprehensive set of services, mechanisms are also provided for implementors to create additional services if desired.

(d) Because of this Standard's adherence to the ISO concept of a "layered" communication architecture, the same messages may be exchanged using various network access methods and physical media. This means that BACnet networks may be configured to meet a range of speed and throughput requirements with commensurately varying cost. Multiple BACnet networks can be interconnected within the same system forming an internetwork of arbitrarily large size. This flexibility also provides a way for BACnet to embrace new networking technologies as they are developed.

BACnet was designed to gracefully improve and evolve as both computer technology and demands of building automation systems change. Upon its original publication in 1995, a Standing Standards Project Committee was formed to deliberate enhancements to the protocol under ASHRAE rules for "continuous maintenance." Much has happened since the BACnet standard was first promulgated. BACnet has been translated into Chinese, Japanese, and Korean, and embraced across the globe. BACnet devices have been designed, built and deployed on all seven continents. Suggestions for enhancements and improvements have been continually received, deliberated, and, ultimately, subjected to the same consensus process that produced the original standard. This publication is the result of those deliberations and brings together all of the corrections, refinements, and improvements that have been adopted.

Among the features that have been added to BACnet are: increased capabilities to interconnect systems across wide area networks using Internet Protocols, new objects and services to support fire detection, other life safety applications, lighting, physical access control, and elevator monitoring, capabilities to backup and restore devices, standard ways to collect trend data, new tools to make specifying BACnet systems easier, a mechanism for making interoperable extensions to the standard visible, and many others. The successful addition of these features demonstrates that the concept of a protocol deliberately crafted to permit extension of its capabilities over time as technology and needs change is viable and sound.

The latest update to this standard includes many new capabilities and several clarifications. These new capabilities include lighting and elevator device profiles that provide a standardized description for these types of devices along with audit reporting and logging to capture auditable actions. BACnet Secure Connect, a BACnet datalink layer, was also added. This datalink is fully compatible with all existing BACnet datalinks and provides an interoperable secure path for BACnet communications.

AVA Documents
Document Sharing Hub

© ISO 2022 – All rights reserved

Full document details: BS EN ISO 16484-5:2022

All communication protocols are, in the end, a collection of arbitrary solutions to the problems of information exchange and all are subject to change as time and technology advance. BACnet is no exception. Still, it is the hope of those who have contributed their time, energies, and talents to this work that BACnet will help to fulfill, in the area of building automation and control, the promise of the information age for the public good!

Documents
Document Sharing Hub

© ISO 2022 – All rights reserved

Full document details: BS EN ISO 16484-5:2022

Documents
Document Sharing Hub

Building automation and control systems (BACS) —

Part 5:
Data communication protocol

1 PURPOSE

The purpose of this standard is to define data communication services and protocols for computer equipment used for monitoring and control of HVAC&R and other building systems and to define, in addition, an abstract, object-oriented representation of information communicated between such equipment, thereby facilitating the application and use of digital control technology in buildings.

2 SCOPE

2.1 This protocol provides a comprehensive set of messages for conveying encoded binary, analog, and alphanumeric data between devices including, but not limited to:

(a) hardware binary input and output values,
(b) hardware analog input and output values,
(c) software binary and analog values,
(d) text string values,
(e) schedule information,
(f) alarm and event information,
(g) files, and
(h) control logic.

2.2 This protocol models each building automation and control computer as a collection of data structures called "objects," the properties of which represent various aspects of the hardware, software, and operation of the device. These objects provide a means of identifying and accessing information without requiring knowledge of the details of the device's internal design or configuration.

3 DEFINITIONS

3.1 Terms Adopted from International Standards

The following terms used in this standard are defined by international standards or draft standards for open system interconnection (OSI). The definitions are repeated here and a reference to the appropriate standard is provided. Clause 25 contains the titles of all national and international standards referenced in this clause and elsewhere in this standard. Words or phrases in italics refer to terms defined elsewhere in this clause.

abstract syntax: the specification of application layer data or application-protocol-control-information by using notation rules which are independent of the encoding technique used to represent them (ISO 8822).

application: a set of a USER's information processing requirements (ISO 8649).

application-entity: the aspects of an application-process pertinent to OSI (ISO 7498).

application-process: an element within a real open system which performs the information processing for a particular application (ISO 7498).

AVA Documents
Document Sharing Hub

© ISO 2022 – All rights reserved

Full document details: BS EN ISO 16484-5:2022

application-protocol-control-information: information exchanged between application-entities, using presentation services, to coordinate their joint operation (ISO 9545).

application-protocol-data-unit: a unit of data specified in an application protocol and consisting of application-protocol-control-information and possibly application-user-data (ISO 9545).

application-service-element: that part of an application-entity which provides an OSI environment capability, using underlying services when appropriate (ISO 7498).

concrete syntax: those aspects of the rules used in the formal specification of data which embody a specific representation of that data (ISO 7498).

confirm (primitive): a representation of an interaction in which a service-provider indicates, at a particular service-access-point, completion of some procedure previously invoked, at that service-access-point, by an interaction represented by a request primitive (ISO TR 8509).

indication (primitive): a representation of an interaction in which a service-provider either
(a) indicates that it has, on its own initiative, invoked some procedure; or
(b) indicates that a procedure has been invoked by the service-user at the peer service-access-point (ISO TR 8509).

peer-entities: entities within the same layer (ISO 7498).

persistent: when used to describe property values, the property value is stored in non-volatile storage so as to survive device resets and power failures.

real open system: a real system which complies with the requirements of OSI standards in its communication with other real systems (ISO 7498).

real system: a set of one or more computers, the associated software, peripherals, terminals, human operators, physical processes, information transfer means, etc., that forms an autonomous whole capable of performing information processing and/or information transfer (ISO 7498).

request (primitive): a representation of an interaction in which a service-user invokes some procedure (ISO TR 8509).

response (primitive): a representation of an interaction in which a service-user indicates that it has completed some procedure previously invoked by an interaction represented by an indication primitive (ISO TR 8509).

(N)-service-access-point: the point at which (N)-services are provided by an (N)-entity to an (N+1)-entity (ISO 7498).

(N)-service-data-unit: an amount of (N)-interface-data whose identity is preserved from one end of an (N)-connection to the other (ISO 7498).

service-user: an entity in a single open system that makes use of a service through service-access-points (ISO TR 8509).

service-primitive; **primitive**: an abstract, implementation-independent representation of an interaction between the service-user and the service-provider (ISO TR 8509).

service-provider: an abstract of the totality of those entities which provide a service to peer service-users (ISO TR 8509).

transfer-syntax: that concrete syntax used in the transfer of data between open systems (ISO 7498).

user element: the representation of that part of an application-process which uses those application-service-elements needed to accomplish the communications objectives of that application-process (ISO 7498).

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

3.2 Terms Defined for this Standard

access control: a method for regulating or restricting access to network resources.

access rights (physical access control): the access privileges granted to a credential.

access user (physical access control): the person or asset holding one or more credentials.

alarm: 1. An annunciation, either audible or visual or both, that alerts an operator to an off-normal condition that may require corrective action. **2.** An abnormal condition detected by a device or controller that implements a rule or logic specifically designed to look for that condition.

alarm-acknowledgment: the process of indicating that a human operator has seen and responded to an event notification.

algorithmic change reporting: the detection and reporting of an alarm or event, based on an algorithm specified in an Event Enrollment object. See intrinsic reporting.

authentication: the act of verifying identity

authentication factor: a data element of the credential which is used to verify a credential's identity.

authorization (network security): the control of access to network resources based on known identity and access rules.

authorization (physical access control): the process of determining whether the access user is permitted to enter a protected zone through an access controlled point.

BACnet address: address format used by the BACnet network layer, as defined in Clause 6, consisting of a network number and a MAC address.

BACnet device: any device, real or virtual, that supports digital communication using the BACnet protocol.

BACnet network: a network of BACnet devices that share the MAC or VMAC address space under a particular BACnet network number.

BACnet-user: that portion of an application-process that is represented by the BACnet user element.

blink-warn: in lighting control, typically a method of notifying room occupants of an impending automated command to turn off the lights whereby the lights may be blinked, once or multiple times, or an audible signal is generated. After the warning occurs, the room lights are held on for a grace period to allow occupants to either safely leave the room or to initiate a request to keep the room lights on. Also known as "flick warn" or "flash warn."

bridge: a device that connects two or more segments at the physical and data link layers. This device may also perform message filtering based upon MAC layer addresses.

broadcast: a message sent as a single unit, which may apply to more than one device.

change of state: an event that occurs when a measured or calculated Boolean or discrete enumerated value changes.

change of value: an event that occurs when a measured or calculated analog value changes by a predefined amount.

client: a system or device that makes use of another device for some particular purpose via a service request instance. A client requests service from a server.

confidentiality: the property that information is not made available or disclosed to unauthorized individuals, entities, or processes.

AVA Documents
Document Sharing Hub

© ISO 2022 – All rights reserved

Full document details: BS EN ISO 16484-5:2022

configurable: a property, setting, or value in a device is configurable if it can be changed via BACnet services or some other method. A property, setting, or value that is one-time writable or not changeable in situ is not considered to be configurable.

context: a set of data or information that completely describes a particular communication environment at a particular point in time.

controller: a device for regulation or management of a system or component.

credential (physical access control): the combination of authentication factors and access rights.

date pattern: a date that may contain one or more unspecified octets or special date values.

directly connected network: a network that is accessible from a BACnet router without messages being relayed through an intervening BACnet router. A PTP connection is to a directly connected network if the PTP connection is currently active and no intervening BACnet router is used.

download: a particular type of file transfer that refers to the transfer of an executable program or database to a remote device where it may be executed.

encrypted message: a message that is wrapped in a security header, signed, and encrypted.

entity: something that has a separate and distinct existence. An identifiable item that is described by a set or collection of properties.

error detection: a procedure used to identify the presence of errors in a communication.

error recovery: a procedure invoked in response to a detected error that permits the information exchange to continue.

event algorithm: the rules that determine when an event-initiating object changes between normal and offnormal states. The event algorithm has no impact on an event-initiating object's transition to or from fault.

event-initiating object: an object that generates event notifications.

event-notification-distribution: the process that a notification-server performs in the determination of notification-clients and in the sending of notifications to notification-clients when an event-initiating object changes the event or acknowledgment state.

event notification message: a ConfirmedEventNotification or UnconfirmedEventNotification service request used to indicate a change in the event or acknowledgment state of an event-initiating object.

event-state-detection: the process of executing an event-initiating object's event algorithm and monitoring the object's Reliability property to detect changes in the object's event state.

event-summarization: the querying of event-initiating objects in a device through one of the event summarization services to determine those that meet specific event state or reporting conditions.

fading: the gradual increase or decrease of the actual output from one setting to another over a fixed period of time.

gateway: a device that connects two or more dissimilar networks, permitting information exchange between them.

global: pertaining to all devices or nodes on a communication internetwork.

global broadcast: a message addressed to all devices or nodes on all networks in a BACnet internet.

half router: a device or node that can participate as one partner in a PTP connection. The two half-router partners that form an active PTP connection together make up a single BACnet router.

initialization: the process of establishing a known state, usually from a power up condition. Initialization may require re-establishment of a node's logical or physical address.

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

integrity: the property that data has not been altered or destroyed in an unauthorized manner.

internetwork: a set of two or more networks interconnected by BACnet routers. In a BACnet internetwork interconnected by BACnet routers, there exists exactly one message path between any two nodes.

intrinsic reporting: the detection and reporting of an alarm or event, based on an algorithm defined as part of the object type specification. No external reference to an Event Enrollment is involved. See algorithmic change reporting.

invalid character: Character-level encoding within a BACnet character string that is not defined by the governing body of the character encoding, or that breaks the rules of the governing body, or is not allowed by the rules of BACnet such as a null character when printable characters are required.

inverted network: a BACnet internetwork where two or more BACnet networks are connected by a BACnet network with an NPDU size smaller than the networks it joins.

key: a sequence of symbols that controls the operations of encipherment and decipherment.

link: a communication facility or medium over which nodes can communicate at the IP link layer.

local: pertaining to devices on the same network as the referenced device.

local broadcast: a message addressed to all devices or nodes on the same BACnet network as the originator.

medium: the physical transmission entity. Typical media are twisted-pair wire, fiber optic cable, and coaxial cable.

medium access control: a process used to maintain order and provide access to the communication medium.

multicast domain: A group of B/IPv6 nodes that can communicate with each other using a particular IPv6 multicast address and UDP port combination.

network: a set of one or more segments interconnected by bridges that have the same network address.

network resource: any physical or logical entity that may be accessed via a communication medium.

node: an addressable device connected to the communication medium.

notification-client: a BACnet device that receives and processes event notification messages.

notification-server: a BACnet device that contains event-initiating objects and performs event-notification-distribution.

numeric quantity: a value that is the magnitude of a quantity. A representation of a specific amount such as temperature, volume or counts.

object: a specific instance of an object type. While an object type is identified by a unique Object_Type property, an object is identified by its Object_Identifier property.

object profile: an object profile is a means of defining objects beyond those defined in Clause 12. A profile defines the set of properties, behavior, and/or requirements for a proprietary object, or for proprietary extensions to a standard object.

object type: a generic classification of data that is defined by a set of properties.

operator authentication: the corroboration that the operator logging on to a device is as claimed.

peer entity authentication: the corroboration that a peer entity in an association is the one claimed.

physical access control system: an electronic system that controls the ability of people or vehicles to enter a protected area, by means of authentication and authorization at access control points.

AVA Documents
Document Sharing Hub

© ISO 2022 – All rights reserved

Full document details: BS EN ISO 16484-5:2022

physical segment: a single contiguous medium to which BACnet nodes are attached.

physically secure: a device or network that is protected from physical access by unauthorized individuals.

printable character: a character that represents a printable symbol as opposed to a device control character. Printable characters include, but are not limited to, upper and lowercase letters, punctuation marks, and mathematical symbols. The exact set depends upon the character set being used.

property: a particular characteristic of an object type.

proprietary: within the context of BACnet, any extension of or addition to object types, properties, PrivateTransfer services, or enumerations specified in this standard.

ramping: the gradual increase or decrease of the actual output from one setting to another at a fixed rate of change.

receiving BACnet-user: the BACnet-user that receives an indication or confirm service primitive.

reliability-evaluation: the process by which an object determines its reliability and thus the value to set into its Reliability property.

remote: pertaining to devices or nodes on a different network than the referenced device.

remote broadcast: a message addressed to all devices or nodes on a different BACnet network than the originator.

repeater: a device that connects two or more physical segments at the physical layer.

requesting BACnet-user: the BACnet-user that assumes the role of a client in a confirmed service.

responding BACnet-user: the BACnet-user that assumes the role of a server in a confirmed service.

role-based access control: access privileges that are assigned to specific roles. Access users acquire privileges through their assigned role.

router: a device that connects two or more BACnet networks at the network layer.

secure network: a network on which all traffic is required to be signed or encrypted.

security: any of a variety of procedures used to ensure that information exchange is guarded to prevent disclosure to unauthorized individuals. Security measures are intended to prevent disclosure of sensitive information even to those who have valid access to the communication network. Security is distinct from access control, although some security can be provided by limiting physical access to the communication medium itself.

segment: a segment consists of one or more physical segments interconnected by repeaters.

sending BACnet-user: the BACnet-user that issues a request or response service primitive.

server: a system or device that responds to a service request instance for some particular purpose. The server provides service to a client.

special date value: a date value that is one of the special values such as "even months", "last day of month", etc. These special date values are used in subcomponents (octets) of a value of type Date.

specific date: a fully specified date. For example, January 24, 1991, Day of week = Thursday. A specific date shall contain no unspecified octets or Special Date Values.

specific datetime: a BACnetDateTime construct composed of a specific date and a specific time.

specific time: a fully specified time. For example, 17:35:45.17 (= 5:35:45.17 P.M.). A specific time shall contain no unspecified octets.

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

standard object type: an object type defined by this standard where the numerical value is within the range reserved for ASHRAE.

standard property: a required or optional property of a standard object type where the numerical value of the property identifier is within the range reserved for ASHRAE and the property is listed in the object type's properties table in Clause 12.

stepping: the increase or decrease of an output value in discrete steps.

synchronization: a facility that allows processes to define and identify specific places in a transmission or exchange that can be used to reset a communication session to a predefined state.

time pattern: a time that may contain one or more unspecified octets.

timestamp: the indication of the point in time recorded for and accompanying the record of an event or operation.

trusted: a term used to refer to devices or networks from which messages are believed to be authentic, either through the use of secure messages or based on the physical security of that device or network.

unit_time: the length of time required to transmit one octet with a start bit and a single stop bit. Ten bit-times.

universal floor number: the number of a floor, beginning with 1 for the absolute lowest floor of the building and incrementing by one for each subsequent floor up.

unspecified date: a date composed entirely of unspecified octets (A value of X'FF' = D'255').

unspecified datetime: a BACnetDateTime construct composed of an unspecified date and an unspecified time.

unspecified octet: an octet used in the context of date, time or BACnetWeekNDay values that contains the value X'FF' = D'255'.

unspecified time: a time composed entirely of unspecified octets (A value of X'FF' = D'255').

upload: the process of transferring an executable program image or a database from a remote device in such a manner as to allow subsequent download.

virtual BACnet device: a BACnet device that is modeled in software, usually in a gateway, and does not exist as a physical BACnet device.

virtual BACnet network: a BACnet network of virtual BACnet devices, usually modeled by a gateway where no physical BACnet network exists.

3.3 Abbreviations and Acronyms Used in this Standard

A	application layer (prefix)
ABA	American Bankers Association
AE	application entity
AHU	air handling unit
ANSI	American National Standards Institute
APCI	application protocol control information
APDU	application layer protocol data unit
API	application program interface
ARCNET	attached resource computer network
ASE	application service element
ASN.1	Abstract Syntax Notation One (ISO 8824)
B' '	denotes that binary notation is used between the single quotes
BAC	building automation and control
BAS	building automation system
BBMD	BACnet/IP broadcast management device

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

BDT	broadcast distribution table
B/IP	BACnet/IP
B/IP-M	BACnet/IP multicast
B/IPv6	BACnet/IPv6
BVLC	BACnet virtual link control
BVLCI	BACnet virtual link control information
BVLL	BACnet virtual link layer
C	conditional
C(=)	conditional (The parameter is semantically equivalent to the parameter in the service primitive to its immediate left in the table.)
CA	certificate authoritiy
CNF	confirm primitive
COV	change of value
CRC	cyclic redundancy check
CSML	control system modeling language
CSV	comma-separated values, as defined by RFC 4180
D' '	denotes that decimal notation is used between the single quotes
DA	local destination MAC layer address
DADR	ultimate destination MAC layer address
DER	data expecting reply (parameter in primitives and messages)
DER	distinguished encoding rules (see ISO/IEC 8825-1)
DESFire	Data Encryption Standard Fast, Innovative, Reliable and Secure
DHCP	Dynamic Host Configuration Protocol
DID	ARCNET destination MAC address
DLEN	1-octet length of ultimate destination MAC layer address
DNET	2-octet ultimate destination network number
DNS	Domain Name Service
DSAP	LLC destination service access point (X'82' for BACnet)
EIB	European Installation Bus
EIBA	European Installation Bus Association
EUI	Extended Unique Identifier (see IEEE 802)
EXEC	capable of executing a service request
FDT	foreign device table
HTTP	hypertext transfer protocol - RFC 2616
IANA	The Internet Assigned Numbers Authority
ICI	interface control information
IEEE	Institute of Electrical and Electronics Engineers
IL	ARCNET information length field
IND	indication primitive
INF	"Infinity", a unique binary pattern representing positive infinity (see ANSI/IEEE 754-1985)
-INF	"Negative infinity", a unique binary pattern representing negative infinity (see ANSI/IEEE 754-1985)
INIT	capable of initiating a service request
IP	Internet Protocol - RFC 791
IPv4	Internet Protocol version 4 - RFC 791
IPv6	Internet Protocol version 6 - RFC 2460
IRI	internationalized resource identifier - RFC 3987
ISO	International Organization for Standardization
JSON	JavaScript object notation - RFC 4627, 7159
JWT	JSON web token - RFC 7519
KNX	The Konnex System Specification: EIB is the core protocol of the Konnex standard. The Konnex System Specification reflects the current status for EIB.
L	data link (prefix)
LAN	local area network
LLC	logical link control (ISO 8802-2)
LPCI	link protocol control information
LPDU	link protocol data unit
LRC	Longitudinal Redundancy Check
LSAP	link service access point
LSDU	link service data unit

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

M	mandatory
M(=)	mandatory (The parameter is semantically equivalent to the parameter in the service primitive to its immediate left in the table.)
MA	medium access (prefix)
MAC	medium access control
MPCI	MAC protocol control information
MPDU	MAC layer protocol data unit
MSDU	MAC service data unit
MS/TP	master-slave/token-passing
N	network layer (prefix)
NaN	"Not a Number", a unique binary pattern representing an invalid number (see ANSI/IEEE 754-1985)
NAT	Network Address Translation or Port Address Translation - RFC 2663
NP	network priority
NPCI	network protocol control information
NPDU	network layer protocol data unit
NRZ	non-return to zero
NSAP	network service access point
NSDU	network service data unit
O	indicates that support of a property is optional
OEO	occupant evacuation operation
OSI	open systems interconnection
P	physical layer (prefix)
PAC	ARCNET data packet header octet
PACS	physical access control system
PCI	protocol control information
PDU	protocol data unit
PICS	protocol implementation conformance statement
PKI	public key infrastructure
PPCI	physical layer protocol control information
PPDU	physical protocol data unit
PPP	Point-To-Point protocol - RFC 1661
PSDU	physical service data unit
PTP	point-to-point
R	indicates that a property shall be supported and readable using BACnet services
RBAC	role-based access control
REQ	request primitive
REST	representational state transfer
RFC	request for comment
RSA	Rivest-Shamir-Adleman cryptosystem
RSP	response primitive
S	selection
S(=)	selection (The parameter is semantically equivalent to the parameter in the service primitive to its immediate left in the table.)
SA	local network source MAC layer address
SAP	service access point
SC	ARCNET system code (X'CD' for BACnet)
SDU	service data unit
SIA	Security Industry Association
SID	ARCNET source MAC address
SLAAC	IPv6 Stateless Auto Address Configuration - RFC 4862
SLEN	1-octet length of original source MAC layer address
SLIP	Serial Line Internet Protocol - RFC 1055
SNET	2-octet original source network number
SOAP	simple object access protocol
SPC	standard project committee
SSAP	LLC source service access point (X'82' for BACnet)
TLS	transport layer security - RFC 2246
TSM	transaction state machine

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

U	user option
U(=)	user option (The parameter is semantically equivalent to the parameter in the service primitive to its immediate left in the table.)
UART	universal asynchronous receiver/transmitter
UDP	User Datagram Protocol - RFC 768
URI	uniform resource identifier - RFC 3986
URL	uniform resource locator - RFC 1738
UTC	Universal Time Coordinated
UUID	universally unique identifier - RFC 4122
VAV	variable air volume
VT	virtual terminal
W	indicates that a property shall be supported, readable, and writable using BACnet services
X' '	denotes that hexadecimal notation is used between the single quotes
XID	eXchange IDentification (ISO 8802-2)
XML	extensible markup language

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

4 BACnet PROTOCOL ARCHITECTURE

The Open System Interconnection (OSI) - Basic Reference Model (ISO 7498) is an international standard that defines a model for developing multi-vendor computer communication protocol standards. The OSI model addresses the general problem of computer-to-computer communication and breaks this very complex problem into seven smaller, more manageable sub-problems, each of which concerns itself with a specific communication function. Each of these sub-problems forms a "layer" in the protocol architecture.

The seven layers are arranged in a hierarchical fashion as shown in Figure 4-1. A given layer provides services to the layers above and relies on services provided to it by the layers below. Each layer can be thought of as a black box with carefully defined interfaces on the top and bottom. An application process connects to the OSI application layer and communicates with a second, remote application process. This communication appears to take place between the two processes as if they were connected directly through their application layer interfaces. Minimal knowledge or understanding of the other layers is required. In a similar manner, each layer of the protocol relies on lower layers to provide communication services and establishes a virtual peer-to-peer communication with its companion layer on the other system. The only real connection takes place at the physical layer.

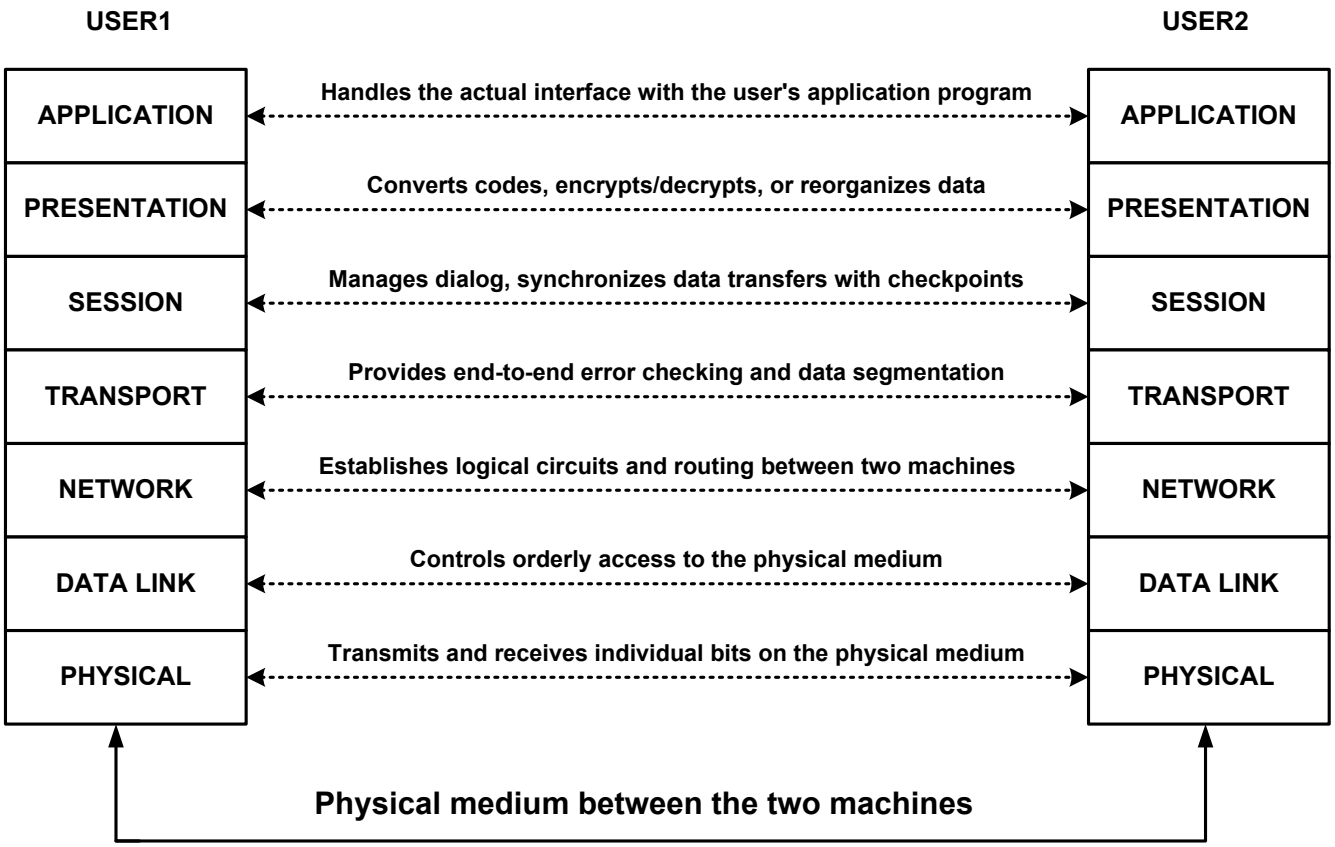

Figure 4-1. The ISO Open Systems Interconnection Basic Reference Model.

The OSI model addresses computer-to-computer communication from a very general perspective. It was designed to deal with the problems associated with computers in large, complex networks communicating with other computers in networks anywhere in the world. In this environment, computers can be separated by long distances and the messages might pass through several intermediate points, each of which may have to make routing decisions or perform some type of translation. Complex synchronization and error recovery schemes may also be needed.

The cost of implementing such a protocol today is prohibitively high for most building automation applications and is not generally required. Nevertheless, the OSI model is a good one to use for a building automation protocol if consideration is given to including only the OSI functionality that is actually needed, thereby collapsing the seven-layer architecture. In a collapsed architecture, only selected layers of the OSI model are included. The other layers are effectively null, thus reducing message length and communication processing overhead. Such a collapsed architecture permits the building automation industry to take advantage of lower cost, mass-produced processor and local area

© ISO 2022 – All rights reserved

AVA Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

network technologies such as have been developed for the process control and office automation industries. The use of readily available, widespread technologies, such as Ethernet[1], ARCNET[2], and LonTalk[3], will lower the cost, increase performance, and open new doors to system integration.

[1] Ethernet is a registered trademark of Digital Equipment Corporation, Intel, and Xerox and is the basis for international standard ISO 8802-3.
[2] ARCNET is a registered trademark of Datapoint Corporation.
[3] LonTalk is a registered trademark of Echelon Corporation.

4.1 The BACnet Collapsed Architecture

BACnet is based on a four-layer collapsed architecture that corresponds to the physical, data link, network, and application layers of the OSI model as shown in Figure 4-2. The application layer and a simple network layer are defined in the BACnet standard. BACnet provides the follwing options that correspond to the OSI data link and physical layers.

Ethernet (ISO 8802-3)	Clause 7
ARCNET (ATA 878.1)	Clause 8
MS/TP	Clause 9
PTP	Clause 10
LonTalk (ISO/IEC 14908.1)	Clause 11
BACnet/IP	Annex J
BACnet/IPv6	Annex U
ZigBee	Annex O
BACnet/SC	Annex AB

Collectively these options provide a master/slave MAC, deterministic token-passing MAC, high-speed contention MAC, dial-up access, Internet access, star and bus topologies, and a choice of twisted-pair, coax, or fiber optic media, in addition to wireless connectivity.

A four-layer collapsed architecture was chosen after careful consideration of the particular features and requirements of BAC networks, including a constraint that protocol overhead needed to be as small as possible. The reasoning behind the selection of the physical, data link, network, and application layers for inclusion in the BACnet architecture is outlined in this clause.

What layers are required for the proper operation of a BAC network? BAC networks function as local area networks, either physically, as with MS/TP, or logically, as with BACnet/IP. This is true even though in some applications it is necessary to exchange information with devices in a building that is very far away. This long-distance communication is done through the telephone networks or across the Internet. The routing, relaying, and guaranteed delivery issues are handled by the telephone and Internet systems and can be considered external to the BAC network. BAC devices are static. They don't move from place to place and the functions that they are asked to perform do not change in the sense that a manufacturing device may make one kind of part today and some very different part tomorrow. These are among the features of BAC networks that can be used to evaluate the appropriateness of the layers in the OSI model.

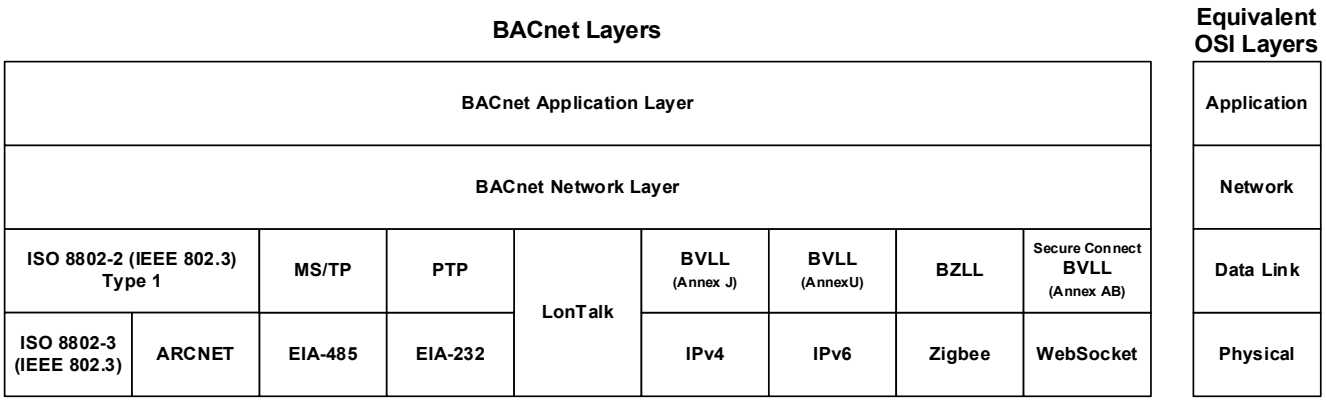

Figure 4-2. BACnet collapsed architecture.

© ISO 2022 – All rights reserved
Documents Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

The physical layer provides a means of connecting the devices and transmitting the electronic signals that convey the data. Clearly the physical layer is needed in a BAC protocol.

The data link layer organizes the data into frames or packets, regulates access to the medium, provides addressing, and handles some error recovery and flow control. These are all functions that are required in a BAC protocol. The conclusion is that the data link layer is needed. For the purposes of BACnet, a data link layer entity may ignore any service request primitive that is not defined for the particular data link type.

Functions provided by the network layer include translation of global addresses to local addresses, routing messages through one or more networks, accommodating differences in network types and in the maximum message size permitted by those networks, sequencing, flow control, error control, and multiplexing. BACnet is designed so that there is only one logical path between devices, thus eliminating the need for optimal path routing algorithms. A network is made up of one or more physical segments connected by repeaters or bridges but with a single local address space. In the case of a single network, most network layer functions are either unnecessary or duplicate data link layer functions. For some BACnet systems, however, the network layer is a necessity. This is the case when two or more networks in a BACnet internet use different MAC layer options. When this occurs, there is a need to recognize the difference between local and global addresses and to route messages to the appropriate networks. BACnet provides this limited network layer capability by defining a network layer header that contains the necessary addressing and control information.

The transport layer is responsible for guaranteeing end-to-end delivery of messages, segmentation, sequence control, flow control, and error recovery. Most of the functions of the transport layer are similar to functions in the data link layer, though different in scope. The scope of transport layer services is end-to-end whereas the scope of data link services is point-to-point across a single network. Since BACnet supports configurations with multiple networks, the protocol must provide the end-to-end services of the transport layer. Guaranteed end-to-end delivery and error recovery are provided in the BACnet application layer via message retry and timeout capabilities. Message segmentation and end-to-end flow control is required for buffer and processor resource management. This is because potentially large amounts of information may be returned for even simple BACnet requests. These functions are provided in the BACnet application layer. Last, sequence control is required in order to properly reassemble segmented messages. This is provided in the BACnet application layer within the segmentation procedure. Since BACnet is based on a connectionless communication model, the scope of the required services is limited enough to justify implementing these at a higher layer, thus saving the communication overhead of a separate transport layer.

The session layer is used to establish and manage long dialogues between communicating partners. Session layer functions include establishing synchronization checkpoints and resetting to previous checkpoints in the event of error conditions to avoid restarting an exchange from the beginning. Most communications in a BAC network are very brief. For example, reading or writing one or a few values, notifying a device about an alarm or event, or changing a setpoint. Occasionally longer exchanges take place, such as uploading or downloading a device. The few times when the services of this layer would be helpful do not justify the additional overhead that would be imposed on the vast majority of transactions, which are very brief and do not need them.

The presentation layer provides a way for communicating partners to negotiate the transfer syntax that will be used to conduct the communication. This transfer syntax is a translation from the abstract user view of data at the application layer to sequences of octets treated as data at the lower layers. If only one transfer syntax is permitted, then the presentation layer function reduces to an encoding scheme for representing the application data. BACnet defines such a fixed encoding scheme and includes it in the application layer, making an explicit presentation layer unnecessary.

The application layer of the protocol provides the communication services required by the applications to perform their functions, in this case monitoring and control of HVAC&R and other building systems. Clearly an application layer is needed in the protocol.

In summary:

(a) The resource and overhead costs for implementing a full OSI seven-layer architecture make it impractical for current building automation devices.
(b) Following the OSI model offers advantages in terms of adopting existing computer networking technology. This can result in cost savings and make integration with other computer network systems easier.
(c) The expectations and environment of building automation systems permit simplification of the OSI model by eliminating the functionality of some of the layers.
(d) A collapsed architecture made up of the physical, data link, network, and application layers is the optimum solution for today's building automation systems.

AVA Documents
Document Sharing Hub

© ISO 2022 – All rights reserved

Full document details: BS EN ISO 16484-5:2022

4.2 BACnet Network Topology

In the interest of application flexibility, BACnet does not prescribe a rigid network topology. Rather, BACnet devices are physically connected to one of the types of local area networks (LANs) or via dedicated or dial-up serial, asynchronous lines. These networks may then be further interconnected by BACnet routers as described in Clause 6.

In terms of LAN topology, each BACnet device is attached to an electrical medium or physical segment. A BACnet segment consists of one or more physical segments connected at the physical layer by repeaters. A BACnet network consists of one or more segments interconnected by bridges, devices that connect the segments at the physical and data link layers and may perform message filtering based upon MAC addresses. A BACnet network forms a single MAC address domain. Multiple BACnet networks, possibly employing different LAN technologies, may be interconnected by BACnet routers to form a BACnet internetwork. In a BACnet internetwork, there exists exactly one message path between any two nodes. These concepts are shown graphically in Figure 4-3.

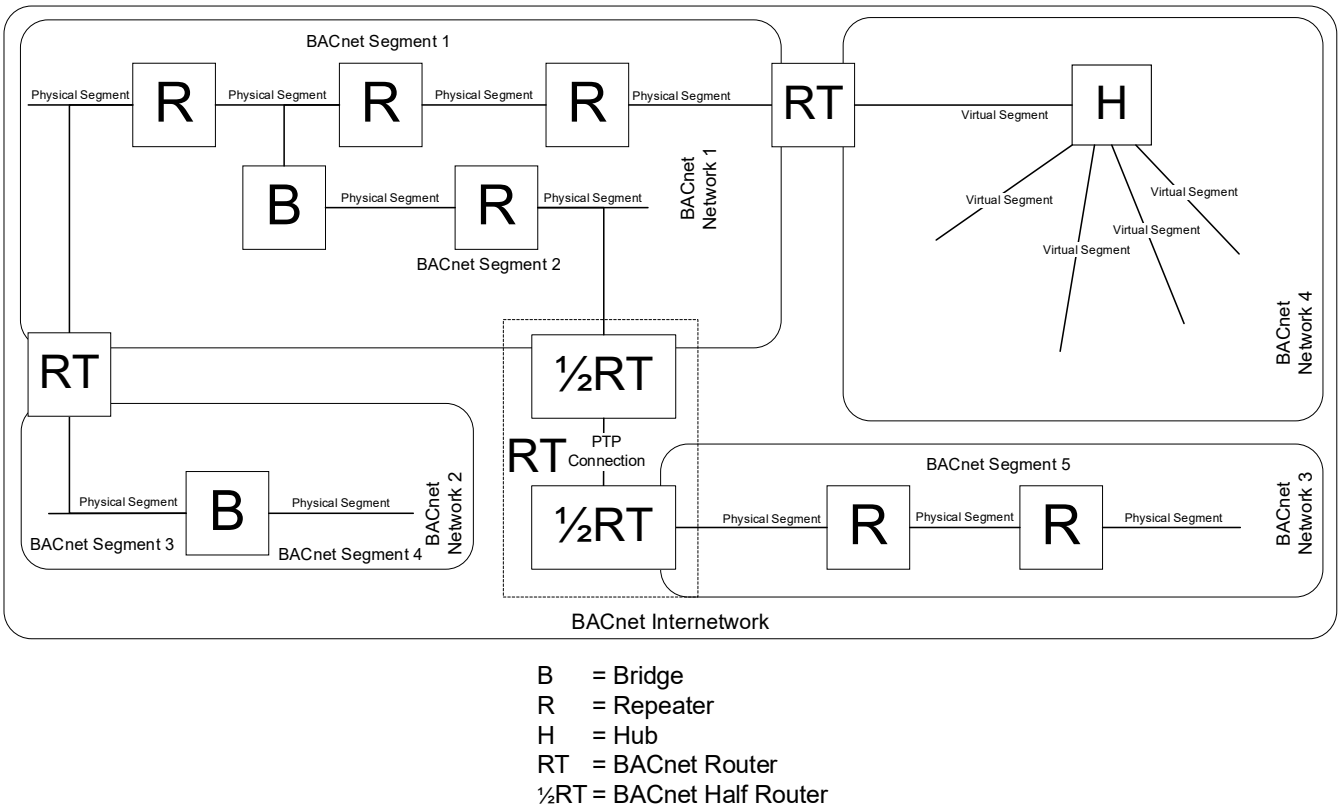

Figure 4-3. A BACnet internetwork illustrating the concepts of Physical Segments, Virtual Segments, Repeaters, Bridges, Hubs, BACnet Segments, BACnet Networks, BACnet Half Routers, and BACnet Routers.

4.3 Security

The principal security threats to BACnet systems are people who, intentionally or by accident, modify a device's configuration or control parameters. Problems due to a malfunctioning or misconfigured computer are outside the realm of security considerations. One important place for security measures is the operator-machine interface. Since the operator-machine interface is not part of the communication protocol, vendors are free to include password protection, audit trails, or other controls to this interface as needed. In addition, write access to any properties that are not explicitly required to be "writable" by this standard may be restricted to modifications made only in virtual terminal mode or be prohibited entirely. This permits vendors to protect key properties with a security mechanism that is as sophisticated as they consider appropriate.

It is recommended that BACnet devices support updating of the device's firmware and software. The procedures for firmware and software upgrades are a local matter.

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

For the BACnet/SC data link layer option, standard network security mechanisms based on Transport Layer Security (TLS, successor of SSL) are used to provide peer authentication, message integrity, and encryption for communication within a BACnet/SC network. See Annex AB.

Documents
Document Sharing Hub

© ISO 2022 – All rights reserved

Full document details: BS EN ISO 16484-5:2022

5 THE APPLICATION LAYER

5.1 The Application Layer Model

This clause presents a model of the BACnet application layer. The purpose of the model is to describe and illustrate the interaction between the application layer and application programs, the relationship between the application layer and lower layers in the protocol stack, and the peer-to-peer interactions with a remote application layer. This model is not an implementation specification.

An Application Process is that functionality within a system that performs the information processing required for a particular application. All parts of the Application Process outside the Application Layer, (i.e. those that do not concern the communication function) are outside the scope of BACnet. The part of the Application Process that is within the Application Layer is called the Application Entity. In other words, an Application Entity is that part of the Application Process related to the BACnet communication function. An application program interacts with the Application Entity through the Application Program Interface (API). This interface is not defined in BACnet, but it would probably be a function, procedure, or subroutine call in an actual implementation. These concepts are illustrated in Figure 5-1. The shaded region indicates the portion of the Application Process that is within the BACnet Application Layer.

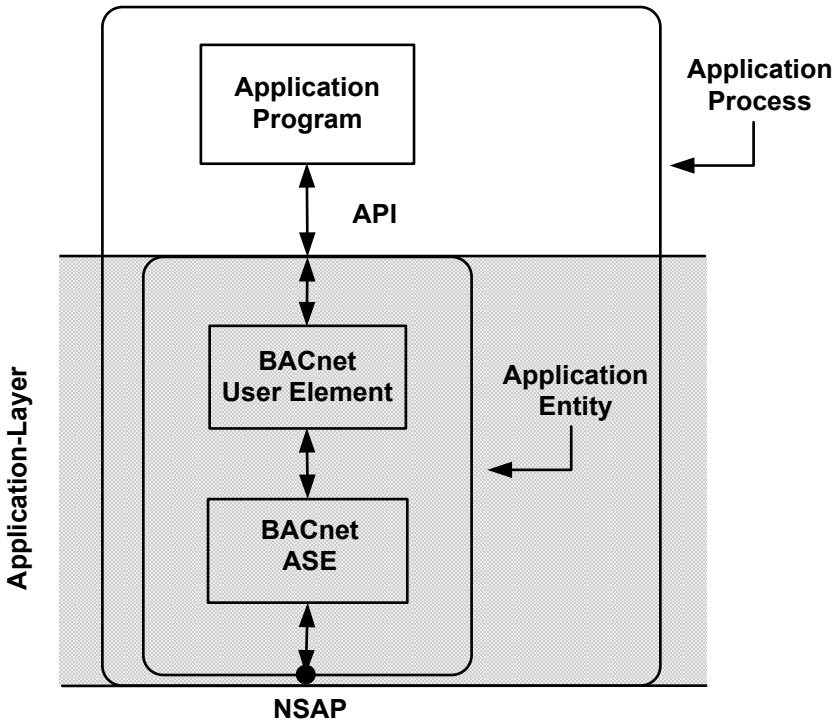

Figure 5-1. Model of a BACnet Application Process.

The Application Entity is itself made up of two parts: the BACnet User Element and the BACnet Application Service Element (ASE). The BACnet ASE represents the set of functions or application services specified in Clauses 13 through 17. The BACnet User Element carries out several functions in addition to supporting the local API. It represents the implementation of the "service procedure" portion of each application service. It is responsible for maintaining information about the context of a transaction, including generating invoke IDs and remembering which invoke ID goes with which application service request (response) to (from) which device. It is also responsible for maintaining the time-out counters that are required for the retrying of a transmission. The BACnet User Element also presides over the mapping of a device's activities into BACnet objects.

Information exchanged between two peer application processes is represented in BACnet as an exchange of abstract service primitives, following the ISO conventions contained in the OSI technical report on service conventions, ISO TR 8509. These primitives are used to convey service-specific parameters that are defined in Clauses 13 through 17. Four service primitives are defined: request, indication, response, and confirm. The information contained in the primitives

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

is conveyed using a variety of protocol data units (PDUs) defined in this standard. In order to make clear which BACnet PDU is being used, the notation will be as follows:

CONF_SERV.request	CONF_SERV.indication	CONF_SERV.response	CONF_SERV.confirm
UNCONF_SERV.request	UNCONF_SERV.indication		
SEGMENT_ACK.request	SEGMENT_ACK.indication		
REJECT.request	REJECT.indication		
ABORT.request	ABORT.indication		
	SEC_ERR.indication		

The designation CONF_SERV indicates that BACnet confirmed service PDUs are being used. Similarly, the designations UNCONF_SERV, SEGMENT_ACK, ERROR, REJECT, and ABORT indicate that unconfirmed service PDUs, segment acknowledge PDUs, error PDUs, reject PDUs, and abort PDUs, respectively, are being used. The designation SEC_ERR indicates that an error occurred in the BACnet security layer and is being indicated up to the BACnet application program. The format of a SEC_ERR.indication is a local matter.

An application program that needs to communicate with a remote application process accesses the local BACnet User Element through the API. Some of the API parameters, such as the identity (address) of the device to which the service request is to be sent and protocol control information, is passed directly down to the network or data link layers. The remainder of the parameters make up an application service primitive that is passed from the BACnet User Element to the BACnet ASE. Conceptually, the application service primitive results in the generation of an APDU that becomes the data portion of a network service primitive, which is passed to the network layer through the Network Service Access Point (NSAP). Similarly this request passes down through the lower layers of the protocol stack in the local device. This process is illustrated in Figure 5-2. The message is then transmitted to the remote device, where it is passed up through the protocol stack in the remote device, eventually appearing as an indication primitive passed from the remote BACnet ASE to the remote BACnet User Element. The response from the remote device, if any, returns to the initiator of the service in a similar fashion (see Clause 5.5).

In addition to the service primitives and the service specific parameters, the application entity exchanges interface control information (ICI) parameters with the application program via the API. The content of the ICI is dependent upon the service primitive type. The ICI parameters received by the application entity provide the information that is passed on to the lower layers (as ICI across layer interfaces) to help them construct their PDUs. The ICI parameters that are provided by the application entity to the application programs contain information recovered by the lower layers from their respective PDUs.

The following ICI parameters are exchanged with the various service primitives across an API:

'destination_address' (DA): the address of the device(s) intended to receive the service primitive. Its format (device name, network address, etc.) is a local matter. This address may also be a multicast, local broadcast or global broadcast type.

'source_address' (SA): the address of the device from which the service primitive was received. Its format (device name, network address, etc.) is a local matter.

'network_priority' (NP): a four-level network priority parameter described in Clause 6.2.2.

'data_expecting_reply' (DER): a Boolean parameter that indicates whether (TRUE) or not (FALSE) a reply service primitive is expected for the service being issued.

'data_attributes' (DAT): The optional parameter that provides extra information about the data for the request to send, or from the received request. This optionally includes security related information in a portion called 'security_parameters'. The format of this parameter is a local matter.

'security_parameters' (SEC): An optional portion of the 'data_attributes' parameter that provides security related information for the request to send, or from the received request. Its format is a local matter.

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

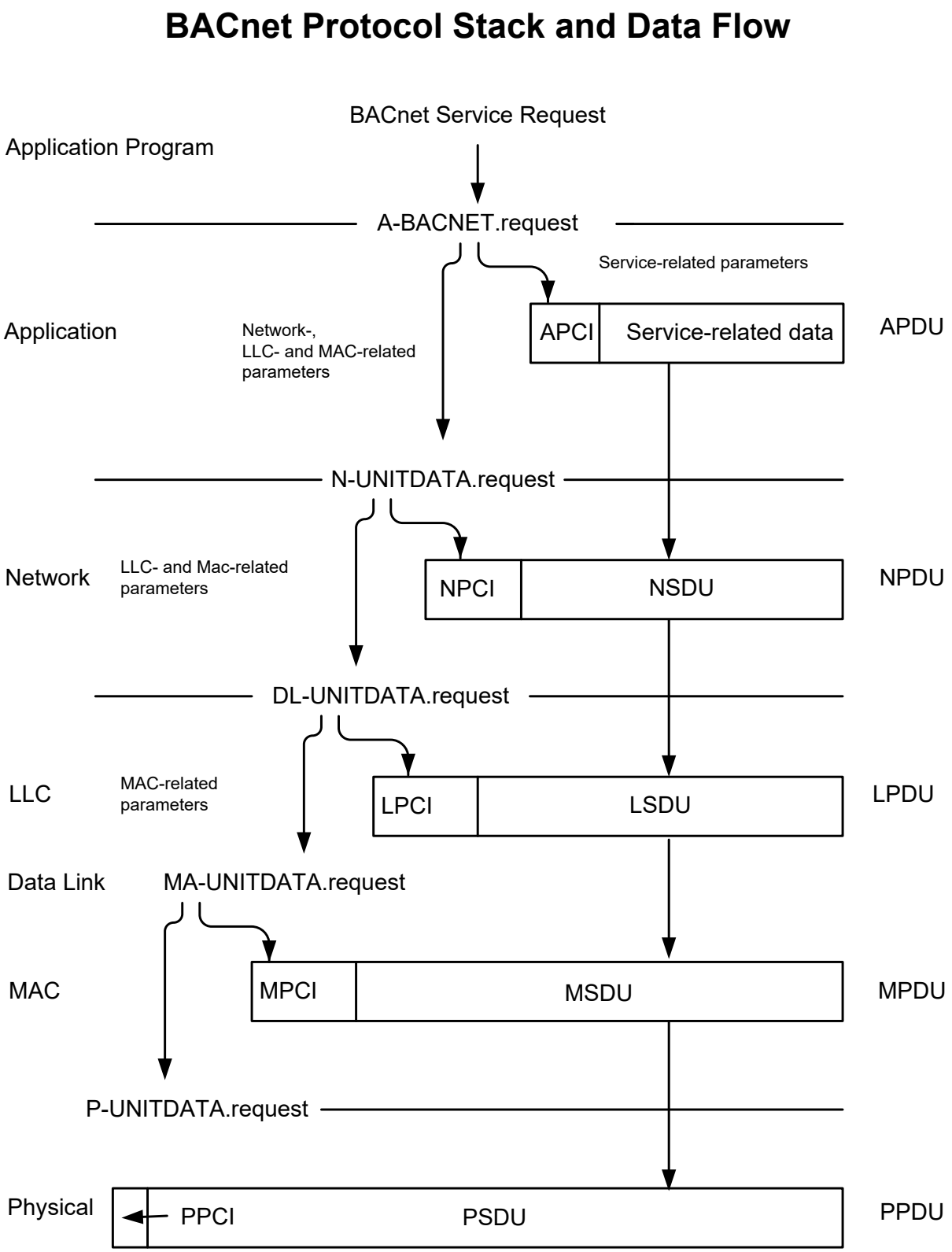

Figure 5-2. BACnet protocol stack and data flow.

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

Table 5-1 describes the applicability of the ICI parameters to the service primitives.

Table 5-1. Applicability of ICI Parameters for Abstract Service Primitives

Service Primitive	DA	SA	NP	DER	SEC
CONF_SERV.request	Yes	No	Yes	Yes	Yes
CONF_SERV.indication	Yes	Yes	Yes	Yes	Yes
CONF_SERV.response	Yes	No	Yes	Yes	Yes
CONF_SERV.confirm	Yes	Yes	Yes	No	Yes
UNCONF_SERV.request	Yes	No	Yes	No	Yes
UNCONF_SERV.indication	Yes	Yes	Yes	No	Yes
REJECT.request	Yes	No	Yes	No	Yes
REJECT.indication	Yes	Yes	Yes	No	Yes
SEGMENT_ACK.request	Yes	No	Yes	No	Yes
SEGMENT_ACK.indication	Yes	Yes	Yes	No	Yes
ABORT.request	Yes	No	Yes	No	Yes
ABORT.indication	Yes	Yes	Yes	No	Yes
SEC_ERR.indication	Yes	Yes	No	No	Yes

A "BACnet device" is any device, real or virtual, that supports digital communication using the BACnet protocol. Each BACnet device contains exactly one Device object, as defined in Clause 12.11. A BACnet device is uniquely located by an NSAP, which consists of a network number and a MAC address.

In most cases, a physical device will implement a single BACnet device. It is possible, however, that a single physical device may implement a number of "virtual" BACnet devices. This is described in Annex H.

Note that some application layer requirements are relaxed for BACnet gateways to non-BACnet protocols. See Annex H.

5.1.1 Confirmed Application Services

BACnet defines confirmed application services based on a client and server communication model. A client requests service from a server via a particular service request instance. The server provides service to a client and responds to the request. This relationship is illustrated in Figure 5-3. The BACnet-user that assumes the role of a client is called the "requesting BACnet-user" and the BACnet-user that assumes the role of the server is called the "responding BACnet-user."

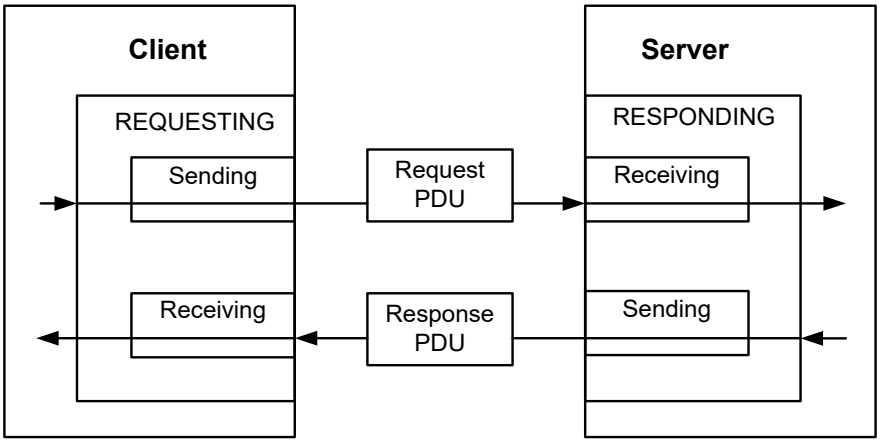

Figure 5-3. Relationship of a client and server.

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

A requesting BACnet-user issues a CONF_SERV.request primitive, which causes a request PDU to be sent. When a response PDU arrives, the requesting BACnet-user receives a CONF_SERV.confirm primitive. When a request PDU arrives, the responding BACnet-user receives a CONF_SERV.indication primitive. The responding BACnet-user issues a CONF_SERV.response primitive, which causes a response PDU to be sent. Thus, the requesting BACnet-user and the responding BACnet-user play a role in both sending and receiving PDUs. The term "sending BACnet-user" applies to a BACnet user that initiates the sending of a PDU. The term "receiving BACnet-user" applies to a BACnet-user that receives an indication that a PDU has arrived.

5.1.2 Unconfirmed Application Services

The client and server model, and the terms "requesting BACnet-user" and "responding BACnet-user," do not apply to unconfirmed services. The terms "sending BACnet-user" and "receiving BACnet-user" do apply, however, and they are used to define the service procedure for unconfirmed services.

In some cases, errors will be encountered when sending unconfirmed requests. While the standard does not provide a mechanism for reporting such errors to the application program, it is acceptable for an implementation to provide such a mechanism, if desired.

5.2 Segmentation of BACnet Messages

To provide for messages that are longer than the maximum length supported by a communications network, or by the sending or receiving device, BACnet provides a method to perform application layer segmentation. In BACnet, only Confirmed-Request and ComplexACK messages may be segmented. Segmentation is an optional feature of BACnet.

5.2.1 Message Segmentation Rules

This clause prescribes rules for dividing a message into segments.

5.2.1.1 Rules for Segmenting APDU Data Streams

Each BACnet message is encoded into a sequence of tags and values according to the relevant ASN.1 definitions in Clause 21 and the encoding rules of Clause 20. The following rules apply to segmenting this data stream:

(a) If possible, an entire message shall be sent in a single APDU.
(b) If an entire message cannot be sent in a single APDU, the message shall be segmented into the minimum number of APDUs possible.
(c) Messages shall be segmented only at octet boundaries.

5.2.1.2 Maximum APDU Length

The maximum length of a BACnet APDU shall be the smallest of

(a) the maximum APDU size transmittable by a device, which may be restricted by local buffer limitations and is a local matter;
(b) the maximum APDU size conveyable by the internetwork to the remote device, which is constrained by the maximum NPDU length permitted by the data links used by the local, remote, and any intervening networks, as specified in Clause 6;
(c) the maximum APDU size accepted by the remote peer device, which must be at least 50 octets.

If the sending device is the requesting BACnet-user, i.e. the APDU to be sent is a BACnet-Confirmed-Request-PDU or a BACnet-Unconfirmed-Request-PDU, then the maximum APDU size accepted by the remote peer is specified by the Max_APDU_Length_Accepted property of the remote peer's Device object. The value of this property may be read using the read property services described in Clause 15 or the value may be obtained from the 'Max APDU Length Accepted' parameter of an I-Am service request received from the remote device. The remote peer may be solicited to transmit an I-Am service request by sending it a Who-Is service request, as described in Clause 16.9

If the sending device is not the requesting BACnet-user, i.e. the APDU to be sent is a BACnet-ComplexACK-PDU, then the maximum APDU size accepted by the remote peer is specified in the 'Max APDU Length Accepted' parameter of the BACnet-Confirmed-Request-PDU for which this is a response.

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

The value determined by the above constraints will be designated the maximum-transmittable-length. Note that maximum-transmittable-length will in general not be a constant unless minimum values are used for each constraint.

See Clause 19.4 for an approach to determine the maximum APDU conveyable by the internetwork.

5.2.1.3 Maximum Segments Accepted

The maximum number of segments transmitted in a Confirmed-Request or ComplexACK message shall be the smallest of:

(a) the maximum number of segments transmittable by a device, which may be restricted by local limitations and is a local matter;
(b) the maximum number of segments accepted by the remote peer device.

If the sending device is the requesting BACnet-user, i.e. the message to be sent is a Confirmed-Request, then the maximum number of segments accepted by the remote peer device is specified in the Max_Segments_Accepted property of the remote peer's Device object.

If the sending device is not the requesting BACnet-user, i.e. the message to be sent is a ComplexACK, then the maximum number of segments accepted by the remote peer device is specified in the 'Max Segments Accepted' parameter of the BACnet-Confirmed-Request-PDU for which this is a response.

5.2.2 Segmentation Protocol Control Information (PCI)

To provide for the possibility of segmented messages, the headers of the BACnet-Confirmed-Request-PDU and BACnet-ComplexACK-PDU contain two Boolean parameters called 'Segmented Message' and 'More Follows.'

If the length of a fully encoded message of the type conveyed by one of the above APDUs results in an APDU whose length is less than or equal to the maximum-transmittable-length as determined according to Clause 5.2.1, the 'Segmented Message' and 'More Follows' parameters shall both be set to FALSE.

If, however, the encoded length of a message would result in an APDU length greater than the maximum-transmittable-length as determined according to Clause 5.2.1, the 'Segmented Message' parameter shall be set to TRUE for all segments, and the 'More Follows' parameter shall be set to TRUE for all segments except the last.

Two additional parameters are also present, conditionally, in the header of each APDU carrying a segment of a Confirmed-Request message or a ComplexACK message. The first conditional parameter is the 'Sequence Number.' This one-octet unsigned integer is used by the segment transmitter to indicate the position of the current segment in the series of segments composing the complete message. The second conditional parameter is the 'Proposed Window Size.' This one-octet unsigned integer is used by the segment transmitter to indicate the maximum number of message segments that it is prepared to transmit before it must receive a SegmentACK. The use of these parameters in the transmission of segmented messages is described in Clauses 5.3 and 5.4.

The 'Sequence Number' of the initial segment shall be zero. The segment receiver may request the transmission of the next segment or group of segments by sending a SegmentACK-PDU containing the 'Sequence Number' parameter of the last successfully received segment. Such a request shall also serve as an acknowledgment of this segment. If the Window Size is greater than one, such a SegmentACK-PDU shall also serve to acknowledge any previously transmitted but unacknowledged segments.

If either party in a segmented transaction wishes to terminate the transaction, that party may issue an Abort-PDU.

5.3 Transmission of BACnet APDUs

The formal description of the transmission and reception protocol for BACnet APDUs is contained in the Transaction State Machine description given in Clause 5.4. This clause is intended only as an overview of the protocol.

5.3.1 Confirmed-Request Message Transmission

Upon transmitting a complete unsegmented Confirmed-Request message or upon receiving the SegmentACK acknowledging the final segment of a segmented Confirmed-Request message, a client device shall start a timer that indicates the length of time the message has been outstanding. The timer shall be canceled upon the receipt of an Error,

AVA Documents
Document Sharing Hub

© ISO 2022 – All rights reserved

Full document details: BS EN ISO 16484-5:2022

Reject, Abort, SimpleACK, or ComplexACK APDU for the outstanding Confirmed-Request message, and the client application shall be notified. If the timer exceeds the value of the APDU_Timeout property in the client's Device object, then the complete Confirmed-Request message shall be retransmitted and the timer shall be reset to zero. All retransmitted Confirmed-Request messages shall follow this same procedure until the message has been retransmitted the number of times indicated in the Number_Of_APDU_Retries property of the client's Device object. If, after the Confirmed-Request message is retransmitted the appropriate number of times, a response is still not received, the message shall be discarded and the client application shall be notified.

5.3.2 Segmented Confirmed-Request Message Transmission

Before sending the first segment of a segmented Confirmed-Request-PDU, a client device shall choose a Proposed Window Size to indicate the maximum number of message segments it is prepared to transmit before it must receive a SegmentACK. The means of choosing the Proposed Window Size are a local matter, except that the value shall be in the range 1 to 127, inclusive. The Proposed Window Size shall be carried by the parameter of that name in each segment of the Confirmed-Request-PDU. The value of Proposed Window Size shall be the same in each segment of the Confirmed-Request-PDU.

Upon transmitting the first segment of a Confirmed-Request message, a client device shall start a timer that indicates the length of time this message segment has been outstanding. The timer shall be canceled upon the receipt of a Reject, Abort, or SegmentACK APDU for the outstanding Confirmed-Request message segment. If the timer exceeds the value of the APDU_Segment_Timeout property in the client's Device object, then the segment shall be retransmitted and the timer shall be reset to zero. All retransmitted segments shall follow this same procedure until the message segment has been retransmitted the number of times indicated in the Number_Of_APDU_Retries property of the client's Device object. If, after the message segments are retransmitted the appropriate number of times, a response is still not received, the message shall be discarded and the client application shall be notified.

Upon receipt of the first segment of a segmented Confirmed-Request-PDU, the server device shall choose an Actual Window Size to indicate the number of sequential message segments it expects to receive before it transmits a SegmentACK. The means of choosing the Actual Window Size are a local matter, except that the value shall be less than or equal to the 'proposed-window-size' parameter contained in the Confirmed-Request-PDU and shall be in the range 1 to 127, inclusive. The value of Actual Window Size shall be the same in each SegmentACK sent in response to a given Confirmed-Request. Regardless of the value of Actual Window Size, a SegmentACK shall be sent in response to the first segment of a Confirmed-Request.

Upon receipt of a SegmentACK APDU, the client device shall set its Actual Window Size equal to the value associated with the 'actual-window-size' parameter in the SegmentACK APDU. After this point, the client has authorization to send as many segments as the 'actual-window-size' parameter indicates before waiting for a SegmentACK APDU. No more than T_{seg} may be allowed to elapse between the receipt of a SegmentACK APDU and the transmission of a segment. No more than T_{seg} may be allowed to elapse between the transmission of successive segments of a group. After transmitting a set of segments that fills the window or completes the message, a client device shall start a timer that indicates the length of time these message segments have been outstanding. The timer shall be canceled upon receipt of a Reject, Abort, or SegmentACK APDU for some or all of the outstanding Confirmed-Request message segments. If the timer exceeds the value of the APDU_Segment_Timeout property in the client's Device object, then the segments shall be retransmitted and the timer shall be reset to zero. All retransmitted segments shall follow this same procedure until the message segments have been retransmitted the number of times indicated in its Device object's Number_Of_APDU_Retries property. If, after the Confirmed-Request message segments are retransmitted the appropriate number of times, a response is still not received, the message shall be discarded and the client application shall be notified.

It is possible to receive a Reject, Abort, or SegmentACK APDU during the sending of a sequence of Confirmed-Request segments even though the number of outstanding segments is less than indicated by the Actual Window Size. In this case, receipt of a Reject or Abort APDU shall terminate the Confirmed-Request transaction. Receipt of a SegmentACK APDU shall be considered as an acknowledgment for the segments up to and including the number indicated in the 'sequence-number' parameter of the SegmentACK APDU. Any unacknowledged segments shall be retransmitted following the above procedure.

It is recognized that in some cases where a Reject, Abort, or SegmentACK APDU is received, the client device may have sent, or irretrievably queued for sending, one or more (but less than Actual Window Size) additional Confirmed-Request-PDU segments.

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

5.3.3 Segmented ComplexACK Message Transmission

Before sending the first segment of a segmented ComplexACK-PDU, a server device shall choose a Proposed Window Size to indicate the maximum number of message segments it is prepared to transmit before it must receive a SegmentACK. The means of choosing the Proposed Window Size are a local matter, except that the value shall be in the range 1 to 127, inclusive. The Proposed Window Size shall be carried by the parameter of that name in each segment of the Confirmed-Request-PDU. The value of Proposed Window Size shall be the same in each segment of the ComplexACK-PDU.

Upon transmitting the first segment of a ComplexACK message, a server device shall start a timer that indicates the length of time this message segment has been outstanding. The timer shall be canceled upon the receipt of an Abort or SegmentACK APDU for the outstanding ComplexACK message segment. If the timer exceeds the value of the APDU_Segment_Timeout property in the server's Device object, then the segment shall be retransmitted and the timer shall be reset to zero. All retransmitted segments shall follow this same procedure until the message segment has been retransmitted the number of times indicated in the Number_Of_APDU_Retries property of the server's Device object. If, after the message segments are retransmitted the appropriate number of times, a response is still not received, the message shall be discarded.

Upon receipt of the first segment of a segmented ComplexACK APDU, the client device shall choose an Actual Window Size to indicate the number of sequential message segments it expects to receive before it transmits a SegmentACK. The means of choosing the Actual Window Size are a local matter, except that the value shall be less than or equal to the 'proposed-window-size' parameter contained in the ComplexACK-PDU and shall be in the range 1 to 127, inclusive. The value of Actual Window Size shall be the same in each SegmentACK sent in response to a given ComplexACK. Regardless of the value of Actual Window Size, a SegmentACK shall be sent in response to the first segment of a ComplexACK.

Upon receipt of a SegmentACK APDU, the server device shall set its Actual Window Size equal to the value associated with the 'actual-window-size' parameter in the SegmentACK APDU. After this point, the server has authorization to send as many segments as the 'actual-window-size' parameter indicates before waiting for a SegmentACK APDU. No more than T_{seg} may be allowed to elapse between the receipt of a SegmentACK APDU and the transmission of a segment. No more than T_{seg} may be allowed to elapse between the transmission of successive segments of a group. After transmitting a set of segments that fills the window or completes the message, a server device shall start a timer that indicates the length of time these message segments have been outstanding. The timer shall be canceled upon receipt of an Abort or SegmentACK APDU for some or all of the outstanding ComplexACK message segments. If the timer exceeds the value of the APDU_Segment_Timeout property in the server's Device object, then the segments shall be retransmitted and the timer shall be reset to zero. All retransmitted segments shall follow this same procedure until the message segments have been retransmitted the number of times indicated in the Number_Of_APDU_Retries property of the server's Device object. If, after the ComplexACK message segments are retransmitted the appropriate number of times, a response is still not received, the message shall be discarded.

It is possible to receive an Abort or SegmentACK APDU during the sending of a sequence of ComplexACK segments even though the number of outstanding segments is less than indicated by the Actual Window Size. In this case, receipt of an Abort APDU shall terminate the ComplexACK transaction. Receipt of a SegmentACK APDU shall be considered as an acknowledgment for the segments up to and including the number indicated in the 'sequence-number' parameter of the SegmentACK APDU. Any unacknowledged segments shall be retransmitted following the above procedure.

It is recognized that in some cases where an Abort or SegmentACK APDU is received, the server device may have sent, or irretrievably queued for sending, one or more (but less than Actual Window Size) additional ComplexACK segments.

5.3.4 SegmentACK APDU Transmission

A device shall transmit a SegmentACK upon any of the following conditions:

(a) The device receives the initial segment of a segmented message. In this case, the 'negative-ack' parameter of the SegmentACK shall have a value of FALSE, indicating that this is a positive acknowledgment, and the 'sequence-number' parameter of the SegmentACK shall have a value of zero, indicating that the first segment has been acknowledged and that the segment transmitter may continue sending, commencing with the next sequential segment.

(b) The device receives a quantity of unacknowledged, sequentially numbered segments for this transaction equal to the Actual Window Size. In this case, the 'negative-ack' parameter of the SegmentACK shall have a value of FALSE, indicating that this is a positive acknowledgment, and the 'sequence-number' parameter of the

AVA Documents
Document Sharing Hub

© ISO 2022 – All rights reserved

Full document details: BS EN ISO 16484-5:2022

SegmentACK shall have a value equal to the 'sequence-number' parameter of the last received segment, indicating that all segments up to and including 'sequence-number' have been acknowledged and that the segment transmitter may continue sending, commencing with the next sequential segment.

(c) The device receives a segment out of order (possibly indicating that a segment has been missed). In this case, the segment receiver shall discard the out-of-order segment. In this context, "out of order" means a segment whose 'sequence-number' is not equal to the next expected 'sequence-number.' The 'negative-ack' parameter of the SegmentACK shall have a value of TRUE, indicating that this is a negative acknowledgment. The 'sequence-number' parameter of the SegmentACK shall have a value equal to the 'sequence-number' parameter of the last received correctly ordered segment, indicating that all segments up to and including 'sequence-number' have been acknowledged and that the segment transmitter should resend, commencing with the next sequential segment after that indicated by the 'sequence-number' parameter contained in the SegmentACK.

(d) The device receives the final segment of a message. In this case, the 'negative-ack' parameter of the SegmentACK shall have a value of FALSE, indicating that this is a positive acknowledgment, and the 'sequence-number' parameter of the SegmentACK shall have a value equal to the 'sequence-number' parameter of the final message segment, indicating that all segments up to and including the final segment have been acknowledged.

5.3.5 Duplicate APDUs and Message Segments

5.3.5.1 Terminating Client TSMs

When using the BACnet error recovery procedures there is a possibility of the reception of duplicate messages or message segments during a transaction. At the client, a transaction begins, and a Transaction State Machine is created, when the first or only segment of a Confirmed-Request APDU is sent. The transaction ends when the client discards the Transaction State Machine due to one of the following circumstances:

(a) after reception from the server of a SimpleACK, unsegmented ComplexACK, Error, Reject, or Abort APDU containing the transaction's invoke-id;

(b) after transmission to the server of a SegmentACK APDU for the final segment of a segmented ComplexACK APDU received from the server;

(c) after exhausting the timeout and retry logic described in the previous clauses;

(d) after transmission to the server of an Abort APDU containing the transaction's invoke-id (i.e. the client aborts the transaction).

5.3.5.2 Terminating Server TSMs

At the server, a transaction begins, and a Transaction State Machine is created when the first or only segment of a Confirmed-Request APDU is received. The transaction ends when the server discards the Transaction State Machine due to one of the following circumstances:

(a) after transmission to the client of a SimpleACK, unsegmented ComplexACK, Error, Reject, or Abort APDU containing the transaction's invoke-id;

(b) after reception from the client of a SegmentACK APDU for the final segment of a segmented ComplexACK APDU transmitted by the server;

(c) after reception from the client of an Abort APDU containing the transaction's invoke-id;

(d) after exhausting the timeout and retry logic described in the previous clauses during the transmission of a segmented ComplexACK APDU.

5.3.5.3 Duplicate Message Procedures

The procedure for handling duplicate messages and message segments is as follows:

(a) The server receives a duplicate Confirmed-Request message. If the server has the capability of detecting a duplicate Confirmed-Request message, the message shall be discarded. If the server cannot distinguish between duplicate and non-duplicate messages, then the Confirmed-Request message shall be serviced. In this case, the client shall discard the server's response since the Invoke ID of the response will not bind to an active state transaction state machine.

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

(b) The server receives a duplicate Confirmed-Request message segment, that is, one that has already been acknowledged with a SegmentACK. In this case, the server shall discard the duplicate segment but shall return an appropriate SegmentACK APDU. A segment can be identified uniquely by the peer address, Invoke ID, and Sequence Number of the segment.

(c) The client receives a duplicate ComplexACK segment, that is, one that has already been acknowledged with a SegmentACK. In this case, the client shall discard the duplicate segment but shall return an appropriate SegmentACK APDU. A segment can be identified uniquely by the peer address, Invoke Id, and Sequence Number of the segment.

(d) A Device receives a duplicate SegmentACK APDU. In this case, the device shall discard the duplicate SegmentACK APDU. Other actions, including the possible re-sending of message segments, shall occur as specified in Clause 5.4.

5.3.6 Stale Resource Disposal

The error recovery procedure described here requires resources from both the server and the client. In the event that the error recovery process fails, the resources dedicated to this process need to be freed. In general, the resources that need to be freed are transaction specific and consist of a Transaction State Machine (TSM), timers, and APDU or APDU segment buffers. The exact time period before the resources should be freed is a local matter dependent upon the system design. As a design suggestion, it is recommended that resources should be considered stale and consequently freed:

(a) at the client, when a complete response to the Confirmed-Request APDU is received;

(b) at the client, when a Confirmed-Request APDU has been retransmitted the number of times specified in the Number_Of_APDU_Retries property without success;

(c) at the client, when a Confirmed-Request APDU segment has been retransmitted the number of times specified in the Number_Of_APDU_Retries property without success;

(d) at the server, when a complete response to a Confirmed-Request APDU has been transmitted and any associated SegmentACK received;

(e) at the server, when a ComplexACK APDU segment has been retransmitted the number of times specified in the Number_Of_APDU_Retries property without success;

(f) at any device, when a SegmentACK APDU has been transmitted and additional segments have not been received before the segment timeout expires.

5.4 Application Protocol State Machines

BACnet APDUs may be divided into two classes: those sent by requesting BACnet-users (clients) and those sent by responding BACnet-users (servers). All BACnet devices shall be able to act as responding BACnet-users and therefore shall be prepared to receive APDUs sent by requesting BACnet-users. Many devices will also be able to act as requesting BACnet-users, and such devices shall be prepared to receive APDUs sent by responding BACnet-users.

<u>APDUs sent by requesting BACnet-users (clients):</u>

BACnet-Unconfirmed-Request-PDU
BACnet-Confirmed-Request-PDU
BACnet-SegmentACK-PDU with 'server' = FALSE
BACnet-Abort-PDU with 'server' = FALSE

<u>APDUs sent by responding BACnet-users (servers):</u>

BACnet-SimpleACK-PDU
BACnet-ComplexACK-PDU
BACnet-Error-PDU
BACnet-Reject-PDU
BACnet-SegmentACK-PDU with 'server' = TRUE
BACnet-Abort-PDU with 'server' = TRUE

© ISO 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

Both the requesting and the responding BACnet-user shall create and maintain a Transaction State Machine (TSM) for each transaction. The TSM shall be created when the transaction begins and shall be disposed of when the transaction ends. In the state machine descriptions that follow, the creation of a TSM is represented by a transition out of the IDLE state, and the disposal of a TSM is represented by a transition into the IDLE state. A transaction is uniquely identified by the client BACnetAddress, the server BACnetAddress, and the Invoke ID (if any).

When a PDU is received from the network layer, the PDU type, the source and destination BACnetAddresses, and the Invoke ID (if any) of the PDU shall be examined to determine the type (requesting BACnet-user or responding BACnet-user) and the identity of the TSM to which the PDU shall be passed. If no such TSM exists, one shall be created.

When a request is received from the application program, the request type, the source and destination BACnetAddresses, and the Invoke ID (if any) of the request shall be examined to determine the type (requesting BACnet-user or responding BACnet-user) and the identity of the TSM to which the request shall be passed. If no such TSM exists, one shall be created.

In order to simplify the state machine description, only the case of segmentation by the Application Entity is shown. Segmentation by the Application Program is possible as well. In this case, wherever the current TSM receives a segment or group of segments and sends SegmentACK, the modified TSM would instead pass the segments to the Application Program, and SegmentACK would be sent only upon direction from the Application Program via the SEGMENT_ACK.request primitive. Reception by the modified state machine of a SegmentACK-PDU would cause it to pass a SEGMENT_ACK.indication primitive to the Application Program.

5.4.1 Variables And Parameters

The following variables are defined for each instance of Transaction State Machine:

RetryCount used to count APDU retries

SegmentRetryCount used to count segment retries

DuplicateCount used to count duplicate segments

SentAllSegments used to control APDU retries and the acceptance of server replies

LastSequenceNumber stores the sequence number of the last segment received in order

InitialSequenceNumber stores the sequence number of the first segment of a sequence of segments that fill a window

ActualWindowSize stores the current window size

ProposedWindowSize stores the window size proposed by the segment sender

SegmentTimer used to perform timeout on PDU segments

RequestTimer used to perform timeout on Confirmed Requests

The following parameters are used in the description:

$\mathbf{T_{seg}}$ This parameter is the length of time a node shall wait for a SegmentACK-PDU after sending the final segment of a sequence. Its value is the value of the APDU_Segment_Timeout property of the node's Device object.

$\mathbf{T_{wait_for_seg}}$ This parameter is the length of time a node shall wait after sending a SegmentACK-PDU for an additional segment of the message. Its value is equal to four times the value of the APDU_Segment_Timeout property of the node's Device object.

$\mathbf{T_{out}}$ This parameter represents the value of the APDU_Timeout property of the node's Device object.

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

N_{retry} This parameter represents the value of the Number_Of_APDU_Retries property of the node's Device object.

N_{dup} This parameter represents the number of duplicates that will be silently dropped per window before a negative segment ack is returned. This parameter shall be equal to ActualWindowSize.

5.4.2 Window Query Functions

5.4.2.1 Function InWindow

The function "InWindow" performs a modulo 256 compare of two unsigned eight-bit sequence numbers. All computations and comparisons are modulo 256 operations on unsigned eight-bit quantities.

function InWindow(seqA, seqB)

(1) if seqA minus seqB, modulo 256, is less than ActualWindowSize, then return TRUE
(2) else return FALSE.

Example (not normative): if ActualWindowSize is equal to 4, then
InWindow(0, 0) returns TRUE
InWindow(1, 0) returns TRUE
InWindow(3, 0) returns TRUE
InWindow(4, 0) returns FALSE
InWindow(4, 5) returns FALSE (since the modulo 256 difference 4 - 5 = 255)
InWindow(0, 255) returns TRUE (since the modulo 256 difference 0 - 255 = 1)

5.4.2.2 Function DuplicateInWindow

The function "DuplicateInWindow" determines whether a value, seqA, is within the range firstSeqNumber through lastSequenceNumber, modulo 256, or if called at the start of a new Window and no new message segments have been received yet, it determines if the value seqA is within the range of the previous Window. All computations and comparisons are modulo 256 operations on unsigned eight-bit quantities.

function DuplicateInWindow(seqA, firstSeqNumber, lastSequenceNumber)

(1) Set local variable receivedCount to lastSequenceNumber minus firstSeqNumber, modulo 256.
(2) If receivedCount is greater than ActualWindowSize, then return FALSE.
(3) If seqA minus firstSeqNumber, modulo 256, is less than or equal to receivedCount, then return TRUE.
(4) If receivedCount is zero and firstSeqNumber minus seqA, modulo 256, is less than or equal to ActualWindowSize, then return TRUE.
(5) Else return FALSE.

Example (not normative): if ActualWindowSize is equal to 4, then
DuplicateInWindow(0, 0, 1) returns TRUE
DuplicateInWindow(1, 0, 1) returns TRUE
DuplicateInWindow(2, 0, 1) returns FALSE
DuplicateInWindow(3, 0, 1) returns FALSE

5.4.3 Function FillWindow

The function "FillWindow" sends PDU segments either until the window is full or until the last segment of a message has been sent. No more than T_{seg} may be allowed to elapse between the receipt of a SegmentACK APDU and the transmission of a segment. No more than T_{seg} may be allowed to elapse between the transmission of successive segments of a sequence.

function FillWindow(sequenceNumber)

(1) Set local variable ix to zero.
(2) If the next segment to transmit (the segment numbered sequenceNumber plus ix) is the final segment, goto step (g).

AVA Documents
Document Sharing Hub

© ISO 2022 – All rights reserved

Full document details: BS EN ISO 16484-5:2022

(3) Issue an N-UNITDATA.request with 'data_expecting_reply' = TRUE to transmit the next BACnet APDU segment, with 'segmented-message' = TRUE, 'more-follows' = TRUE, 'proposed-window-size' equal to ProposedWindowSize, and 'sequence-number' = sequenceNumber plus ix, modulo 256.
(4) Set ix equal to ix plus one.
(5) If ix is less than ActualWindowSize, goto step (2).
(6) Goto step (9).
(7) Issue an N-UNITDATA.request with 'data_expecting_reply' = TRUE to transmit the final BACnet APDU segment with 'segmented-message' = TRUE, 'more-follows' = FALSE, 'proposed-window-size' = ProposedWindowSize, and 'sequence-number' = sequenceNumber plus ix, modulo 256.
(8) Set SentAllSegments to TRUE, indicating that all segments have been transmitted at least once.
(9) Return to the caller.

5.4.4 State Machine for Requesting BACnet User (client)

5.4.4.1 IDLE

In the IDLE state, the device waits for the local application program to request a service.

SendUnconfirmed

If UNCONF_SERV.request is received from the local application program,

then issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-Unconfirmed-Request-PDU, and enter the IDLE state.

SendConfirmedUnsegmented

If CONF_SERV.request is received from the local application program and the length of the APDU is less than or equal to maximum-transmittable-length as determined according to Clause 5.2.1,

then assign an 'invoke-id' to this transaction; set SentAllSegments to TRUE; set RetryCount to zero; start RequestTimer; issue an N-UNITDATA.request with 'data_expecting_reply' = TRUE to transmit a BACnet-Confirmed-Request-PDU with 'segmented-message' = FALSE; and enter the AWAIT_CONFIRMATION state to await a reply.

CannotSend

If CONF_SERV.request is received from the local application program and the length of the APDU is greater than maximum-transmittable-length as determined according to Clause 5.2.1 and the Max_Segments_Accepted property of the destination's Device object is known and the total APDU cannot be transmitted without exceeding the maximum number of segments accepted,

then send an ABORT.indication with 'server' = FALSE and 'abort-reason' =APDU_TOO_LONG to the local application program and enter the IDLE state.

SendConfirmedSegmented

If CONF_SERV.request is received from the local application program and the length of the APDU is greater than maximum-transmittable-length as determined according to Clause 5.2.1, and the Max_Segments_Accepted property of the destination's Device object is not known, or Max_Segments_Accepted is known and the total APDU can be transmitted without exceeding the maximum number of segments accepted,

then assign an 'invoke-id' to this transaction; set SentAllSegments to FALSE; set RetryCount to zero; set SegmentRetryCount to zero; set InitialSequenceNumber to zero; set ProposedWindowSize to whatever value is desired; set ActualWindowSize to 1; start SegmentTimer; issue an N-UNITDATA.request with 'data_expecting_reply' = TRUE to transmit a BACnet-Confirmed-Request-PDU containing the first segment of the message, with 'segmented-message' = TRUE, 'more-follows' = TRUE, 'sequence-number' = zero, and 'proposed-window-size' = ProposedWindowSize; and enter the SEGMENTED_REQUEST state to await an acknowledgment. (The method used to determine ProposedWindowSize is a local matter, except that the value shall be in the range 1 to 127, inclusive.)

UnexpectedSegmentInfoReceived

If an unexpected PDU indicating the existence of an active server TSM (BACnet-ComplexACK-PDU with 'segmented-message' = TRUE or BACnet-SegmentACK-PDU with 'server' = TRUE) is received from the network layer,

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

then issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-Abort-PDU with 'server' = FALSE and 'abort-reason' = INVALID_APDU_IN_THIS_STATE and enter the IDLE state.

UnexpectedPDU_Received
If an unexpected PDU not indicating the existence of an active server TSM (BACnet-SimpleACK-PDU, BACnet-ComplexACK-PDU with 'segmented-message' = FALSE, BACnet-Error-PDU, BACnet-Reject-PDU, or BACnet-Abort-PDU with 'server' = TRUE) is received from the network layer,

then enter the IDLE state. (There is no reason to issue REJECT.indication, ABORT.indication, etc., as the client has no knowledge of the transaction in question.)

SecurityError_Received
If a security error is received via an N-REPORT.indication from the network layer for which there is no active TSM associated with it,

then enter the IDLE state.

5.4.4.2 SEGMENTED_REQUEST

In the SEGMENTED_REQUEST state, the device waits for a BACnet-SegmentACK-PDU for one or more segments of a BACnet-Confirmed-Request-PDU.

SecurityError_Received
If a security error is received via an N-REPORT.indication from the network layer,

then stop SegmentTimer; send SEC_ERR.indication with the appropriate security error information to the local application program; and enter the IDLE state.

InsufficientSecurity_Received
If a PDU is received from the network layer and the security parameters do not match the initial request,

then stop SegmentTimer; issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-Abort-PDU with 'server' = FALSE and an 'abort-reason' = INSUFFICIENT_SECURITY describing the security error; send SEC_ERR.indication indicating insufficient security to the local application program; and enter the IDLE state.

DuplicateACK_Received
If a BACnet-SegmentACK-PDU that has sufficient security parameters and whose 'server' parameter is TRUE is received from the network layer and InWindow ('sequence-number' parameter of the BACnet-SegmentACK-PDU, InitialSequenceNumber) returns a value of FALSE,

then restart SegmentTimer and enter the SEGMENTED_REQUEST state to await an acknowledgment.

NewACK_Received
If a BACnet-SegmentACK-PDU that has sufficient security parameters and whose 'server' parameter is TRUE is received from the network layer and InWindow ('sequence-number' parameter of the BACnet-SegmentACK-PDU, InitialSequenceNumber) returns a value of TRUE and there is at least one segment remaining to send,

then set InitialSequenceNumber equal to the 'sequence-number' parameter of the BACnet-SegmentACK-PDU plus one, modulo 256; set ActualWindowSize equal to the 'actual-window-size' parameter of the BACnet-SegmentACK-PDU; set SegmentRetryCount to zero; call FillWindow (InitialSequenceNumber) to transmit one or more BACnet-Confirmed-Request-PDUs containing the next ActualWindowSize segments of the message; restart SegmentTimer; and enter the SEGMENTED_REQUEST state to await an acknowledgment.

FinalACK_Received
If a BACnet-SegmentACK-PDU that has sufficient security parameters and whose 'server' parameter is TRUE is received from the network layer and InWindow ('sequence-number' parameter of the BACnet-SegmentACK-PDU, InitialSequenceNumber) returns a value of TRUE and there are no more segments to send,

then stop SegmentTimer; start RequestTimer; and enter the AWAIT_CONFIRMATION state to await a reply.

AVA Documents
Document Sharing Hub

© ISO 2022 – All rights reserved

Full document details: BS EN ISO 16484-5:2022

Timeout

If SegmentTimer becomes greater than T_{seg} and SegmentRetryCount is less than N_{retry},

then increment SegmentRetryCount; call FillWindow(InitialSequenceNumber) to retransmit one or more BACnet-Confirmed-Request-PDUs containing the next ActualWindowSize segments of the message; restart SegmentTimer; and enter the SEGMENTED_REQUEST state to await an acknowledgment.

FinalTimeout

If SegmentTimer becomes greater than T_{seg} and SegmentRetryCount is greater than or equal to N_{retry},

then stop SegmentTimer; send ABORT.indication with 'server' = FALSE and 'abort-reason' = TSM_TIMEOUT to the local application program; and enter the IDLE state.

AbortPDU_Received

If a BACnet-Abort-PDU that has sufficient security parameters and whose 'server' parameter is TRUE is received from the network layer,

then stop SegmentTimer; send ABORT.indication to the local application program; and enter the IDLE state.

SimpleACK_Received

If a BACnet-SimpleACK-PDU that has sufficient security parameters is received from the network layer and SentAllSegments is TRUE,

then stop SegmentTimer; send CONF_SERV.confirm(+) to the local application program; and enter the IDLE state.

UnsegmentedComplexACK_Received

If a BACnet-ComplexACK-PDU that has sufficient security parameters is received from the network layer whose 'segmented-message' parameter is FALSE and SentAllSegments is TRUE,

then stop SegmentTimer; send CONF_SERV.confirm(+) to the local application program; and enter the IDLE state.

SegmentedComplexACK_Received

If a BACnet-ComplexACK-PDU that has sufficient security parameters is received from the network layer whose 'segmented-message' parameter is TRUE and whose 'sequence-number' parameter is zero and this device supports segmentation and SentAllSegments is TRUE,

then save the BACnet-ComplexACK-PDU segment; stop SegmentTimer; compute ActualWindowSize based on the 'proposed-window-size' parameter of the received BACnet-ComplexACK-PDU and on local conditions; issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-SegmentACK-PDU with 'negative-ack' = FALSE, 'server' = FALSE, and 'actual-window-size' = ActualWindowSize; start SegmentTimer; set LastSequenceNumber to zero; set InitialSequenceNumber to zero; set DuplicateCount to zero; and enter the SEGMENTED_CONF state to receive the remaining segments. (The method used to determine ActualWindowSize is a local matter, except that the value shall be less than or equal to the 'proposed-window-size' parameter of the received BACnet-ComplexACK-PDU and shall be in the range 1 to 127, inclusive.)

ErrorPDU_Received

If a BACnet-Error-PDU that has sufficient security parameters is received from the network layer and SentAllSegments is TRUE,

then stop SegmentTimer; send CONF_SERV.confirm(-) to the local application program; and enter the IDLE state.

RejectPDU_Received

If a BACnet-Reject-PDU that has sufficient security parameters is received from the network layer,

then stop SegmentTimer; send REJECT.indication to the local application program; and enter the IDLE state.

© ISO 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS EN ISO 16484-5:2022

UnexpectedPDU_Received

If a BACnet-SimpleACK-PDU, BACnet-ComplexACK-PDU, or BACnet-Error-PDU that has sufficient security parameters is received from the network layer and SentAllSegments is FALSE,

or if a BACnet-ComplexACK-PDU that has sufficient security parameters is received from the network layer whose 'segmented-message' parameter is TRUE and this device does not support segmentation,

or if a BACnet-ComplexACK-PDU that has sufficient security parameters is received from the network layer whose 'segmented-message' parameter is TRUE and whose 'sequence-number' parameter is not zero,

then stop SegmentTimer; issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-Abort-PDU with 'server' = FALSE; send ABORT.indication with 'server' = FALSE and 'abort-reason' = INVALID_APDU_IN_THIS_STATE to the local application program; and enter the IDLE state.

SendAbort

If ABORT.request is received from the local application program,

then stop SegmentTimer; issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-Abort-PDU with 'server' = FALSE; and enter the IDLE state.

5.4.4.3 AWAIT_CONFIRMATION

In the AWAIT_CONFIRMATION state, the device waits for a response to a BACnet-Confirmed-Request-PDU.

SecurityError_Received

If a security error is received via an N-REPORT.indication from the network layer,

then stop RequestTimer; send SEC_ERR.indication with the appropriate security error information to the local application program; and enter the IDLE state.

InsufficientSecurity_Received

If a PDU is received from the network layer and the security parameters do not match the initial request,

then stop RequestTimer; issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-Abort-PDU with 'server' = FALSE and an 'abort-reason' = INSUFFICIENT_SECURITY; send SEC_ERR.indication indicating insufficient security to the local application program; and enter the IDLE state.

SimpleACK_Received

If a BACnet-SimpleACK-PDU that has sufficient security parameters is received from the network layer,

then stop RequestTimer; send CONF_SERV.confirm(+) to the local application program; and enter the IDLE state.

UnsegmentedComplexACK_Received

If a BACnet-ComplexACK-PDU that has sufficient security parameters is received from the network layer whose 'segmented-message' parameter is FALSE,

then stop RequestTimer; send CONF_SERV.confirm(+) to the local application program; and enter the IDLE state.

SegmentedComplexACK_Received

If a BACnet-ComplexACK-PDU that has sufficient security parameters is received from the network layer whose 'segmented-message' parameter is TRUE and whose 'sequence-number' parameter is zero and this device supports segmentation,

then stop RequestTimer; compute ActualWindowSize based on the 'proposed-window-size' parameter of the received BACnet-ComplexACK-PDU and on local conditions; issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-SegmentACK-PDU with 'negative-ack' = FALSE, 'server' = FALSE, and 'actual-window-size' = ActualWindowSize; start SegmentTimer; set LastSequenceNumber to zero; set InitialSequenceNumber to zero; set DuplicateCount to zero; and enter the SEGMENTED_CONF state to

AVA Documents
Document Sharing Hub

Full document details: BS EN ISO 16484-5:2022