BSR/ASHRAE Addendum bj to
ANSI/ASHRAE Standard 135-2016

Public Review Draft

Proposed Addendum bj to Standard 135-2016, BACnet® - A Data Communication Protocol for Building Automation and Control Networks

Third Public Review (June 2019)
(Draft shows Proposed Changes to Current Standard)

This draft has been recommended for public review by the responsible project committee. To submit a comment on this proposed standard, go to the ASHRAE website at www.ashrae.org/standards-research--technology/public-review-drafts and access the online comment database. The draft is subject to modification until it is approved for publication by the Board of Directors and ANSI. Until this time, the current edition of the standard (as modified by any published addenda on the ASHRAE website) remains in effect. The current edition of any standard may be purchased from the ASHRAE Online Store at www.ashrae.org/bookstore or by calling 404-636-8400 or 1-800-727-4723 (for orders in the U.S. or Canada).

This standard is under continuous maintenance. To propose a change to the current standard, use the change submittal form available on the ASHRAE website, www.ashrae.org.

The appearance of any technical data or editorial material in this public review document does not constitute endorsement, warranty, or guaranty by ASHRAE of any product, service, process, procedure, or design, and ASHRAE expressly disclaims such.

ASHRAE, 1791 Tullie Circle, NE, Atlanta GA 30329-2305

[This foreword, the table of contents, the introduction, and the "rationales" on the following pages are not part of this standard. They are merely informative and do not contain requirements necessary for conformance to the standard.]

FOREWORD

The purpose of this addendum is to present a proposed change for public review. These modifications are the result of change proposals made pursuant to the ASHRAE continuous maintenance procedures and of deliberations within Standing Standard Project Committee 135. The proposed changes are summarized below.

In the following document, language to be added to existing clauses of ANSI/ASHRAE 135-2016 and Addenda is indicated through the use of *italics*, while deletions are indicated by ~~strikethrough~~. Where entirely new subclauses are proposed to be added, plain type is used throughout. Only this new and deleted text is open to comment at this time. All other material in this document is provided for context only and is not open for public review comment except as it relates to the proposed changes.

The use of placeholders like XX, YY, ZZ, X1, X2, NN, x, n, ? etc., should not be interpreted as literal values of the final published version. These placeholders will be assigned actual numbers/letters only after final publication approval of the addendum.

[This Table of Contents and the listed clause headers in brackets are provided only for convenience in this addendum. They will not be part of the standard]

Table of Contents

[The following introduction is informative and provided as background information and rationale for this addendum. It will not be part of the standard.]

INTRODUCTION

The BACnet protocol stack as outlined in Clause 4 of the standard was defined before 1995, when the TCP/IP protocol suite was expensive and not available for smaller devices common in building automation. With today's availability of IP network infrastructures for building automation that may be shared with other applications, and may be professionally managed by an IT department, there is a need for a more IT-friendly BACnet solutions that allows communicating BACnet across such infrastructures. This introduction first outlines current issues with BACnet in such environments and provides an overview of the solution proposed in this addendum.

Problem Statement

The following sections summarize the issues with BACnet in today's IP-based network infrastructures.

Networking Today and Trends

- Computer networking has become a synonym for IP networks.
- IP is the ubiquitous network infrastructure standard.
- Customers are familiar with and have IP networks in place.
- IP networking is emerging into:
    - All kinds of communication domains.
    - All kinds of media, including wireless (e.g., WPAN, WLAN, Cellular).
    - Field devices (Internet of Things, wired & wireless, CoRE).
    - Mobile devices.
- In the building automation and controls domain, network standards are rapidly moving to IP-centric solutions.
- IPv6 is emerging.
- The building automation and controls market is demanding simple plug-and-play devices.
- The size of internetworked systems is increasing.

BACnet needs to be enabled for shared and managed IP network infrastructures!

BACnet Issues in Shared IP Infrastructures

Building Automation Systems using BACnet may be required to use an IP infrastructure that is shared with office and other applications. Such infrastructures are typically managed by IT departments, for whom BACnet is an unknown protocol. BACnet/IP and BACnet/IPv6 have features and behaviors that are not well accepted by IT departments. As a result, the current application of BACnet has multiple issues from an IT perspective:

- Does not follow standards and behaviors acceptable by IT departments.
- Data security is not suitable for IT networks because it is not based on widely used standards such as Transport Layer Security (TLS).
- Demand for fixed IP addresses, in particular for BBMDs.
- UDP broadcasts that may propagate through the entire network are not acceptable.
- The use of BACnet routers is perceived as adding some extra routing to IP networks that is not manageable by the IT department.
- Excessive use of IT administered IP addresses may result in high infrastructure lease costs.

Values of BACnet that Need to be Preserved

- Designed for control, operation, and monitoring of BA domains.
- Powerful data & services model that reaches into semantic definitions.
- Interoperability among versions and vendors.
- Large installed base.
- Scalability (including support of inexpensive single twisted pair wired networks).
- Comprehensiveness of the network security architecture.

Features that Need to be Enabled for BACnet

- Enable the use of IP networks in a way suitable for highly managed IP infrastructures.
- Enable the use of standard IP application protocols, such as HTTP and WebSockets.
- Enable seamless and simple traversal of typical IP network hurdles, such as NATs and firewalls.
- Enable the use of IP infrastructure that is built for and is shared with office and other applications.
- Enable the use of the same networking infrastructure that is used for complementary systems, such as smart grid and enterprise applications.
- Enable the use of standard IP mechanisms for auto-configuration, name resolution, information security, and device discovery.
- Integration into management of IP network infrastructures (IT-managed environments).
- BACnet communication being completely agnostic to underlying IPv4 or IPv6 flavor of transport layer (even in mixed-scenarios).

Proposed Solution

This addendum proposes a new datalink layer option that makes full use of TLS secured WebSocket connections. This new BACnet Secure Connect datalink layer option uses a virtual hub-and-spoke topology where the spokes are WebSocket connections from the nodes to the hub.

Introduction of the BACnet Secure Connect Datalink Layer Option

The new BACnet Secure Connect, or BACnet/SC, datalink layer option enables full compatibility with all other datalink options of BACnet. The regular BACnet Network Layer routing function connects BACnet devices implementing existing datalink layer options with those implementing BACnet Secure Connect. However, since BACnet Secure Connect is based on TLS secured WebSocket connections, and WebSocket URIs are used for network level addressing, the most urgent features for compliance with IT infrastructure requirements are provided through this datalink option.

Among those IT infrastructure requirements are:

- No specific IT configurations for building automation. The use of the WebSockets protocol which is based on HTTP conforms to typical firewall configurations. No extra configuration by IT is required.
- Secured communication. The use of TLS to secure the WebSocket connections supports sufficient privacy and security for building automation communication in shared IT infrastructures.
- DNS host name resolution and DHCP supported. The use of WebSocket URIs includes the option for DNS host names and their resolution to IP addresses. Through this, DHCP based IP configuration is enabled.
- Works over IPv4 and over IPv6. The BACnet/SC microprotocol does not depend on the IP version and respective address format.
- Enable use of standard NATs. TCP based protocols such as the WebSocket protocol enable standard NAT procedures for TCP connections both in IPv4 and IPv6. The initiating TCP port is not relevant.

IP/IT management functionality such as SNMP may be needed so as to enable minimal common IT network management functionality even in multi-vendor installations. The definition of IP/IT management functionality is out of scope for this addendum.

The minimum BACnet/SC implementation is expected to work on relatively small device platforms.

The BACnet/SC datalink option supports the existing Clause 24 security architecture. But more important, it supports the use of Transport Layer Security (TLS) for securing the BACnet communication. Therefore, the BACnet/SC virtual link control messages do not require a Clause 24 security wrapper option. However, BACnet/SC messages are designed to add authentication and authorization data in the future.

The support of PKI and X.509 certificates enable strong security for all nodes and messages within a given BACnet/SC network.

Through these features, BACnet/SC supports a variety of shared IP infrastructures, such as:

- Use of standard IT provided infrastructure.
- Use in managed IT environments. Support standard IT network management workflows and policies.
- Use in non-managed IT environments.
- Use in BAS installed IT environments, by BAS field engineers.
- Use in network environments that are shared with other than building automation applications (web servers, enterprise applications, office applications, factory automation, etc.).
- Use in environments with limited availability of IT managed IP addresses.

The security features of BACnet/SC provide sufficient security to BACnet communication for a range of security environments:

- Use in open and unprotected environments.
- Use in environments that provide different levels of protection.
- Use for building automation applications that require different levels of security.
- Use in environments that are required to comply with security policies and security equipment in place.
- Changing security requirements over system life-time (e.g., commissioning under factory credentials, secured operation with operational credentials, increased security, new algorithms, key lengths, etc.).

135-2016*bj*-1. Introduce BACnet Secure Connect Datalink Layer Option

Rationale

The current BACnet protocol architecture uses various BACnet specific and IP based datalinks. The IP based datalinks (BVLL and BVLLv6) have properties and behaviors that do not always fit with the requirements, policies, and constraints of IT departments that administer and manage IP network infrastructures.

The need for using standardized and often already present IP network infrastructures for BACnet communication is increasing. Based on existing and new technologies, these IP infrastructures are currently reaching out both to small and constrained devices such as sensors and actuators (Internet of Things), but also into wide-area connectivity and cloud based applications.

The BACnet set of datalink options is extended to facilitate the use of the WebSocket protocol in IT environments. A new BACnet Secure Connect (BACnet/SC) datalink layer option is added which is specifically designed to meet the requirements of minimally managed to professionally managed IP infrastructures. This also includes the defined application of standard IP technologies for network security, such as TLS.

The use of VMACs for datalink addressing enables seamless communication with devices implemented on other datalinks.

[Insert new entries to **Clause 3.2**, preserving the alphabetical order, p. 2]

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

[Change entries in **Clause 3.2**, p. 2]

directly connected network: a *BACnet* 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.

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.

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

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.

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

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

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

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

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

[Insert new entries to **Clause 3.3**, preserving the alphabetical order, p. 8]

EUI Extended Unique Identifier (see IEEE 802)

[Insert new entries to **Clause 25**, preserving the alphabetical order, p. 932]

IETF RFC 8446 (2018), The Transport Layer Security (TLS) Protocol Version 1.3, Internet Engineering Task Force

IETF RFC 5289 (2008), TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM), Internet Engineering Task Force

IETF RFC 6455 (2011), The WebSocket Protocol, Internet Engineering Task Force

IETF RFC 6762 (2013), Multicast DNS, Internet Engineering Task Force

IETF RFC 7251 (2014), AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS, Internet Engineering Task Force

IETF RFC 7468 (2015), Textual Encodings of PKIX, PKCS, and CMS Structures, Internet Engineering Task Force

[Change **Clause 4.1**, p. 12]

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 following 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 YY*

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.

. . .

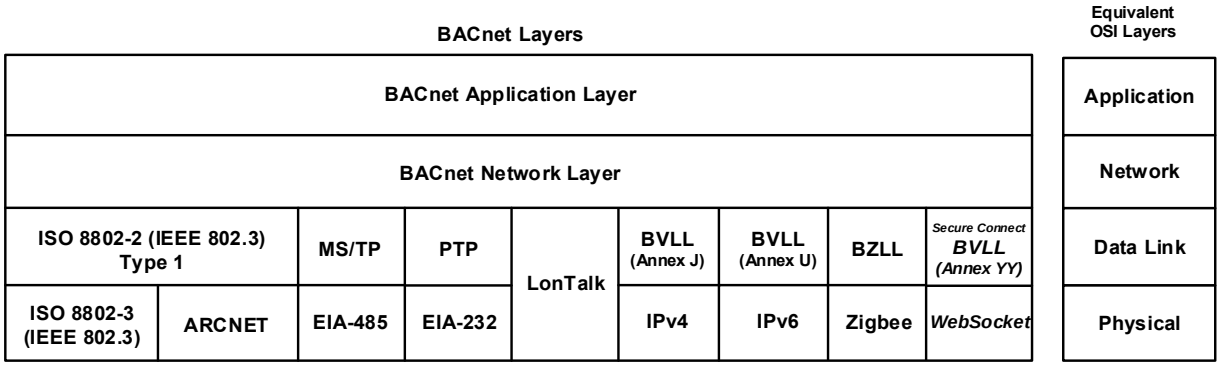
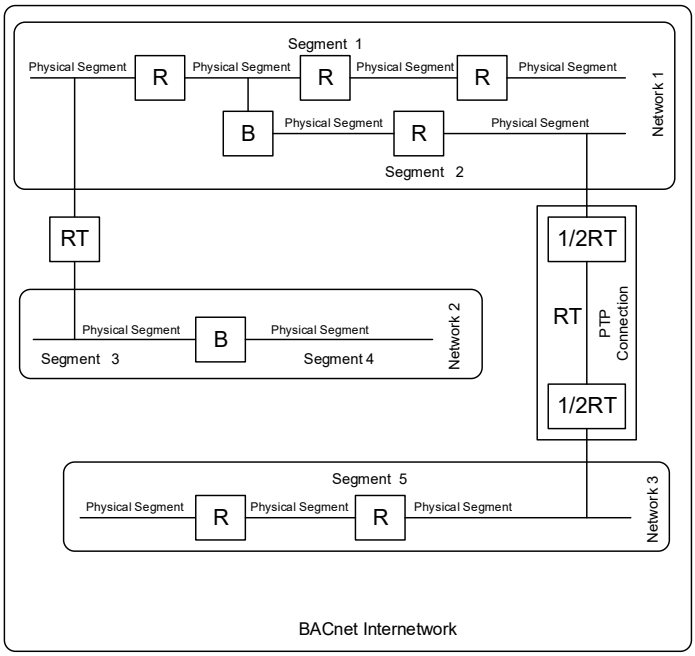


Figure 4-2. BACnet collapsed architecture

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.

. . .

[Replace **Figure 4-3**, p. 15]
[Current Figure 4-3, to be removed:]



B = Bridge
R = Repeater
RT = Router
1/2RT = Half Router

[New Figure 4-3:]

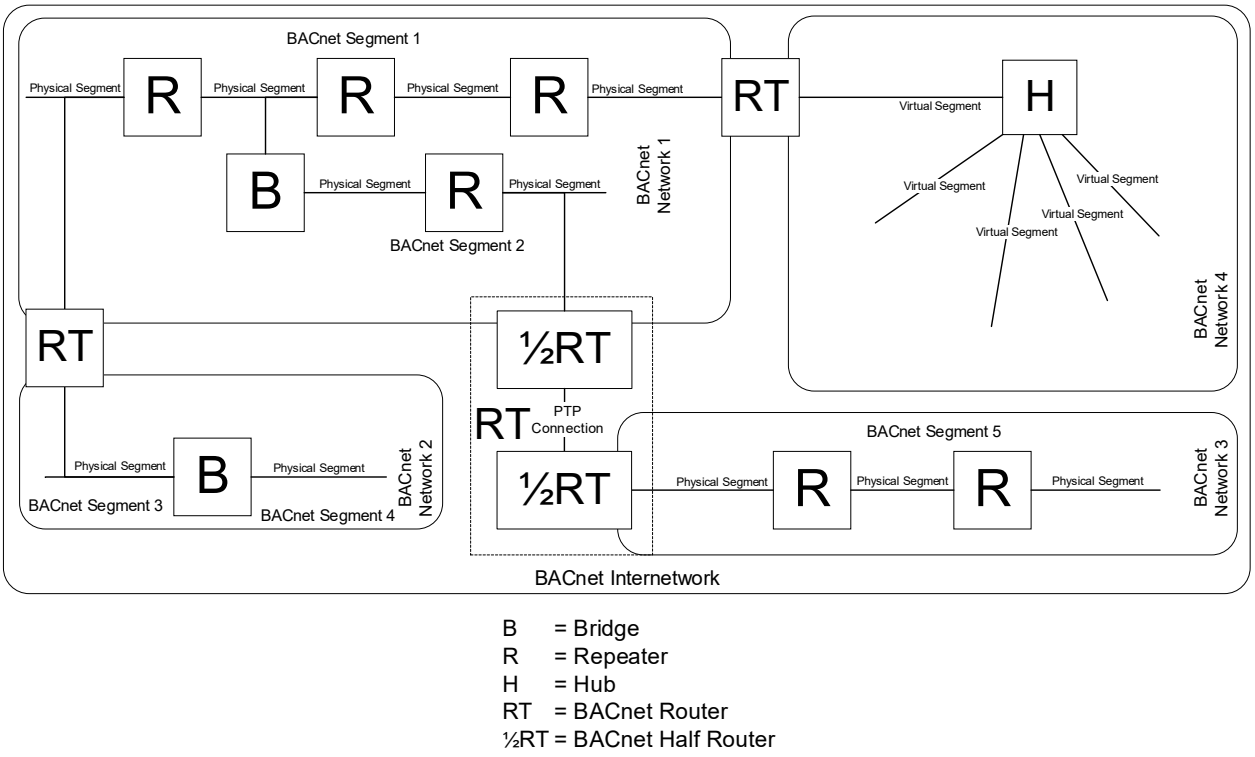


Figure 4-3. A BACnet internetwork, illustrating the concepts of Physical Segments, *Virtual Segments,* Repeaters, *Bridges, Hubs, BACnet* Segments, ~~Bridges,~~ *BACnet* Networks, *BACnet* Half Routers, and *BACnet* Routers.

[Change **Clause 4.3**, p. 12]

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 ~~an errant~~ *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.

BACnet also defines services that can be used to provide peer entity, data origin, and operator authentication. See Clause 24.

For the BACnet/SC datalink 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 YY.

135-2016*bj*-2. Introduce BACnet/SC in the Network and Application Layer Specifications

Rationale

The BACnet Secure Connect datalink layer option and the processing of BACnet/SC data options is introduced in the application and network layer.

[Change **Clause 5.1,** paragraph before Figure 5-1, p. 18]

5.1 The Application Layer Model

. . .

'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): ~~The optional security parameters~~ *An optional portion of the 'data_attributes' parameter that provides security related information* for the request to send, or from the received request. ~~It indicates the level of security (Key Id, Plain/Signed/Encrypted, User Authentication data, End-To-End, etc.), and its~~ *Its* format is a local matter.

[Change **Table 6-1**, p. 54]

6 THE NETWORK LAYER

. . .

. . . Another common network layer function is message segmentation and reassembly. To obviate the need for these capabilities at the network layer, BACnet imposes a limitation on the length of the NPDU in messages passed through a BACnet router. The maximum NPDU length shall not exceed the capability of any data link technology encountered along the path from source to destination. A list of the maximum NPDU lengths for BACnet data link technologies *and the minimum routed NPDU length between any two data link types supported by a BACnet router* is given in Table 6-1.

Table 6-1. ~~Maximum~~ NPDU Lengths ~~When Routing Through~~ *of* ~~Different~~ BACnet Data Link Layers

Data Link Technology	**Maximum NPDU Length *on Data Link***	***Minimum Routed NPDU Length***
ARCNET (ATA 878.1), as defined in Clause 8	501 octets	*501 octets*
BACnet/IP, as defined in Annex J	1497 octets	*1497 octets*
BACnet/IPv6, as defined in Annex U	1497 octets	*1497 octets*
BACnet/SC, as defined in Annex YY [1]	*61327 octets*[2]	*1497 octets*
Ethernet (ISO 8802-3), as defined in Clause 7	1497 octets	*1497 octets*
LonTalk, as defined in Clause 11	228 octets	*228 octets*
MS/TP, as defined in Clause 9	1497 octets	*1497 octets*
Point-To-Point, as defined in Clause 10	501 octets	*501 octets*
ZigBee, as defined in Annex O	501 octets	*501 octets*

[1] BACnet routing between network ports of this data link technology shall include forwarding additional data attributes along with the NPDU being forwarded. See Clause 6.6.

[2] This number is the result of subtracting the BACnet/SC BVLC message header size (16 octets) and the minimally supported data options size (4192 octets) from the maximum BVLC message size of 65535 octets.

. . .

[Change **Clause 6.1**, p. 54]

6.1 Network Layer Service Specification
…

N-UNITDATA.request (
destination_address,
data,
network_priority,
data_expecting_reply,
~~security_parameters~~ *data_attributes*
)

N-UNITDATA.indication (
source_address,
destination_address,
data,
network_priority,
data_expecting_reply,
~~security_parameters~~ *data_attributes*
)

N-RELEASE.request (
destination_address
)

N-REPORT.indication (
peer_address,
error_condition,
error_parameters,
~~security_parameters~~ *data_attributes*
)

The 'destination_address' and 'source_address' parameters provide the logical concatenation of 1) an optional network number, 2) the MAC address appropriate to the underlying LAN technology, and the 3) the link service access point. A network number of X'FFFF' indicates that the message is to be broadcast "globally" to all devices on all currently reachable networks. Currently reachable networks are those networks to which an active connection is already established within the BACnet internet. In particular, a global broadcast shall not trigger any attempts to establish PTP connections. The 'data' parameter is the network service data unit (NSDU) passed down from the application layer and is composed of a fully encoded BACnet APDU. The 'network_priority' is a numeric value used by the network layer in BACnet routers to determine any possible deviations from a first-in-first-out approach to managing the queue of messages awaiting relay. The data_expecting_reply parameter indicates whether (TRUE) or not (FALSE) a reply data unit is expected for the data unit being transferred. The optional parameter *'data_attributes' contains extra information about the 'data' parameter and can include the optional* 'security_parameters' *information that* contains the security information used to secure the request and context information required for security related N-REPORT.indication primitives to be related to application TSMs.

…

The N-REPORT.indication primitive is used by the local network layer to indicate failures to transmit N-UNITDATA.requests to peer devices. The errors may be locally detected error conditions, or error conditions reported by a peer device via a network layer message. This primitive is used extensively by the network security wrapper to indicate security errors up the stack. The 'peer_address' parameter is of the same form as the 'destination_address' or 'source_address' parameters of the N-UNITDATA primitives and indicates the peer with which the error condition arose. The optional parameter *'data_attributes' contains extra information about the error including the optional* 'security_parameters' *portion that* conveys information describing the security failure and context required to relate the error to a previous N-UNITDATA.request or N-UNITDATA.indication primitive.

[Change **Table 6-2**, p. 59]

Table 6-2. BACnet DADR and SADR Encoding Rules Based Upon Data Link Layer Technology

BACnet Data Link Layer	DLEN	SLEN	Encoding Rules
ARCNET, as defined in Clause 8	1	1	Encoded as in their MAC layer representations
BACnet/IP, as defined in Annex J	6	6	Encoded as specified in Clause J.1.2
BACnet/IPv6, as defined in Annex U	3	3	Encoded as specified in Clause H.7.2
BACnet/SC, as defined in Annex YY	*6*	*6*	*Encoded as specified in Clause H.7.X*
Ethernet, as defined in Clause 7	6	6	Encoded as in their MAC layer representations
LonTalk domain wide broadcast	2	2	The encoding for the SADR is shown in Figure 6-3 The encoding for the DADR is shown in Figure 6-4
LonTalk multicast	2	2	
LonTalk unicast	2	2	
LonTalk, unique Neuron_ID	7	2	
MS/TP, as defined in Clause 9	1	1	Encoded as in their MAC layer representations
ZigBee, as defined in Annex O	3	3	Encoded as specified in Clause H.7.2

[Change Clause 6.5, p. 66]

6.5 Network Layer Procedures

This clause describes the network layer procedures to be followed by BACnet router and non-router nodes for both local and remote data transfer. "Local" means that the source and destination devices are on the same BACnet network. "Remote" means that the source and destination devices are on different BACnet networks. The source and destination networks are interconnected by zero or more intervening networks joined by BACnet routers to form a BACnet internetwork. See Figure 4-3.

If a BACnet device supports a datalink that supports the 'data_attributes' parameter, the network layer shall support the forwarding of data attributes with a size of 4192 octets at a minimum. The insertion of 'data_attributes' for network layer messages originating at the local network layer is a local matter.

6.5.1 Network Layer Procedures for the Transmission of Local Traffic

Upon receipt of an N-UNITDATA.request primitive, the network entity (NE) shall inspect the DNET portion of the 'destination_address' parameter. The absence of DNET indicates that the destination device resides on the same BACnet network as the device issuing this transmission request. The value of the 'network_priority' parameter shall be included in the NPCI control octet although its use by receiving non-router entities is unspecified. The NE shall prepare a control NPCI octet indicating the absence of DNET, DADR, HOP COUNT, SNET, and SADR, concatenate it with the 'data' parameter conveyed in the N-UNITDATA.request primitive, and issue a DL-UNITDATA data link request primitive. The concatenation of the NPCI and the NSDU (the 'data' parameter from the N-UNITDATA.request), the NPDU, is passed as the 'data' parameter of the data link primitive. *If the N-*

UNITDATA.request provided 'data_attributes' parameter, it shall be passed to the datalink if the datalink supports this parameter.

6.5.2 Network Layer Procedures for the Receipt of Local Traffic

Upon receipt of an NPDU from the data link layer (conveyed by the 'data' parameter of the DL-UNITDATA data link indication primitive) whose first octet indicates BACnet version one, the destination NE shall interpret the second octet of the NPDU as control NPCI. If bit 7 of the control NPCI indicates that the message contains an APDU, then the procedure in Clause 6.5.2.1 is followed. Otherwise, a network layer message is being conveyed and the procedure in Clause 6.5.2.2 applies.

6.5.2.1 Receipt of Local APDUs

If the control NPCI octet indicates the absence of a DNET field or a DNET field is present and contains the global broadcast address X'FFFF', the NE shall attempt to locate a BACnet application entity. If a BACnet application entity is found, the NE shall issue an N-UNITDATA.indication primitive with the portion of the data link data following the NPCI as the 'data' parameter. *If the 'data_attributes' parameter was received with the data link indication primitive, it shall be passed to the N-UNITDATA.indication primitive*. If the application entity is not found and the NE resides in a non-routing node, the data link data shall be discarded. If the DNET is present and not equal to the global broadcast address X'FFFF' and the NE resides in a non-routing node, the data link data shall likewise be discarded and no further action taken. If the DNET is present and the NE resides in a BACnet router, the NE shall take the actions specified in Clause 6.5.4.

6.5.2.2 Receipt of Local Network Layer Messages

If the control NPCI octet indicates the absence of a DNET field or a DNET field is present and contains the global broadcast address X'FFFF', the NE shall attempt to interpret the network layer message. If the DNET field is absent and the message cannot be interpreted and the message was directed specifically at the router, a Reject-Message-To-Network shall be returned to the device that sent the message.

If the DNET is present and not equal to the global broadcast address X'FFFF' and the NE resides in a non-routing node, the data link data shall be discarded and no further action taken. If the DNET is present and the NE resides in a BACnet router, the NE shall take the actions specified in Clause 6.5.4.

6.5.3 Network Layer Procedures for the Transmission of Remote Traffic

Upon receipt of an N-UNITDATA.request primitive, the NE shall inspect the DNET portion of the 'destination_address' parameter. The presence of a DNET signifies that the destination device resides on a different BACnet network than the device issuing this transmission. A DNET value of X'FFFF' signifies a global broadcast and indicates that the message is to be directed to all local routers via a broadcast message on the local network. The NE shall prepare an NPCI control octet indicating the presence of DNET, DADR, and Hop Count but the absence of SNET and SADR. The NE shall also fill in the network priority field using the supplied parameter. The resulting control, priority, and address information shall then be concatenated with the 'data' parameter conveyed in the N-UNITDATA.request primitive and issued as a DL-UNITDATA data link request primitive. The concatenation of the NPCI and the 'data' parameter from the N-UNITDATA.request (the NSDU), the NPDU, is passed as the 'data' parameter of the data link primitive. *If the N-UNITDATA.request provided the 'data_attributes' parameter, it shall be passed to the datalink if the datalink supports that parameter.* The DA portion of the 'destination_address' parameter passed to the data link layer shall be the MAC address of the BACnet router corresponding to the DNET parameter or the appropriate broadcast DA if the address of the router is initially unknown. The broadcast DA is also to be used if the DNET global broadcast network number is present.

Note that five methods exist for establishing the address of a BACnet router for a particular DNET: 1) the address may be established manually at the time a device is configured, 2) the address may be learned by issuing a Who-Is request and noting the SA associated with the subsequent I-Am message (assuming the device specified in the Who-Is is located on a remote DNET and the I-Am message was handled by a router on the local network), 3) by using the network layer message Who-Is-Router-To-Network, 4) by using the local broadcast MAC address in the initial transmission to a device on a remote DNET and noting the SA associated with any subsequent responses from the remote device, and 5) by noting the SA associated with any requests received from the remote DNET. Which method is used shall be a local matter; however, devices shall not rely solely on method 1.

The local broadcast MAC address may be used in response messages, although it is discouraged. It is preferable that a device note the SA associated with the original request and reuse that SA in the response. For MS/TP networks, in order for MS/TP master devices to use the local broadcast MAC address in a response, a Reply Postponed MAC frame shall be sent in response to the BACnet Data Expecting Reply frame and the response may then be sent when the MS/TP master device receives the token. MS/TP slave devices are unable to use the local broadcast MAC address for responses because they never receive the token.

6.5.4 Network Layer Procedures for the Receipt of Remote Traffic

Upon receipt of an NPDU from the data link layer (conveyed by the 'data' parameter of the DL-UNITDATA indication primitive) whose first octet indicates BACnet version one, the NE shall interpret the second octet of the NPDU as control NPCI. If the NPCI control octet indicates the presence of a DNET field whose value is not X'FFFF' and the NE resides in a BACnet device that is not a router, the message shall be discarded. If the NPCI control octet indicates the presence of a DNET field and the NE resides in a BACnet router, it shall place the NPDU *and the data attributes received (conveyed by the 'data_attributes' parameter of the DL-UNITDATA.indication primitive), if any,* in its message queue (or queues, if separate queues are maintained for each DNET), arranged in order by priority. Within each priority, the messages shall be arranged in first-in-first-out order. If the NPCI control octet indicates that the NPDU contains a network layer message, the NE shall, in addition, inspect the Message Type field. If this field indicates the presence of a Reject-Message-To-Network message, the NE shall carry out the processing specified in Clause 6.6.3.5. If the SNET and SADR fields are present, the message has arrived from a peer router. If the SNET and SADR fields are absent, the message originated on a network directly connected to the router. In the latter case, the router shall add the SNET and SADR to the NPCI based on the router's knowledge of the network number of the network from which the message arrived, alleviating the requirement that the originating station know its own network number. The SADR field shall be set equal to the SA of the incoming NPDU.

If the NPCI control octet indicates the presence of a DNET field, the NE resides in a BACnet router, the NPDU is to be routed to a different device and the NPDU requires a reply (conveyed by the ‘data_expecting_reply’ parameter of the DL-UNITDATA indication primitive), then a DL-RELEASE request shall be issued to the data link layer entity specified by the source_address value of the DL-UNITDATA indication.

Three possibilities exist: either the router is directly connected to the network referred to by DNET, the message must be relayed to another router for further transmission, or a global broadcast is required. In the first case, DNET, DADR, and Hop Count shall be removed from the NPCI and the message shall be sent directly to the destination device with DA set equal to DADR. The control octet shall be adjusted accordingly to indicate only the presence of SNET and SADR. In the second case, if the Hop Count is greater than zero, the message shall be sent to the next router on the path to the destination network. If the next router is unknown, an attempt shall be made to identify it using a Who-Is-Router-To-Network message. If the Hop Count is zero, then the message shall be discarded. If the DNET global broadcast network number is present and the Hop Count is greater than zero, the router shall broadcast the message on each network to which the router is directly connected, except the network of origin, using the broadcast address appropriate to each data link. If the DNET global broadcast network number is present and the Hop Count is zero, then the message shall be discarded.

If data attributes were received with the NPDU, these data attributes shall be forwarded with the NPDU to the datalinks through the 'data_attributes' parameter of the DL-UNITDATA.request, if supported. If this parameter is not supported by the DL-UNITDATA.request primitive, the data attributes shall be ignored.

[Change Clause 6.6, p. 67]

6.6 BACnet Routers

BACnet routers are devices that interconnect two or more BACnet networks to form a BACnet internetwork. BACnet routers shall, at a minimum, implement the device requirements as specified in Clause 22.1.5. Table 6-1 specifies the maximum NPDU length *and the minimum NPDU length being routed* of the different data link layer types. Routers shall

be capable of routing ~~the maximum sized~~ NPDUs between any two of those data link layers supported by the router based on the ~~destination data link maximum NPDU size~~. *minimum routed NPDU lengths defined in Table 6-1.*

BACnet/SC to BACnet/SC BACnet routers, referred to as secure connect BACnet routers, shall support at least two BACnet/SC network ports and may support routing and forwarding of NPDUs exceeding 1497 octets, up to 61327 octets maximum between these ports. Secure connect BACnet routers shall support forwarding of the 'data_attributes' parameter content with the NPDU. Secure connect BACnet routers shall support forwarding of a minimum of 4192 octets of 'data_attributes' content between all BACnet/SC ports. See Clause YY.1.4.

BACnet routers make use of BACnet network layer protocol messages to maintain their routing tables. Routers perform the routing tasks described in Clause 6.5. See Figure 6-12 for a flow chart of router operation.

135-2016*bj*-3. Add new Annex YY for the BACnet Secure Connect Datalink Layer Option

Rationale

The BACnet Secure Connect (BACnet/SC) datalink is a new datalink layer option addressing modern IP infrastructure and IT security requirements. The use of the WebSocket protocol (RFC 6455) with Transport Layer Security (TLS) enables the secure exchange of NPDU packets across a wide range of non-managed to tightly managed IT environments. State-of-the-art TLS enables strong information security for BACnet communication.

The BACnet/SC datalink is based on a logical hub-and-spoke model where the spokes are bi-directional hub connections between nodes and a hub function. The hub function forwards directed and broadcast messages between connected nodes based on 6-octet virtual MAC addresses. The BACnet/SC datalink is designed to be able to work with different types of hub functions. Connecting to the BACnet/SC hub function defined in this Annex is required to be supported by all BACnet/SC nodes. The BACnet/SC hub function is an optional functionality of a network port that also has a BACnet/SC node. The use of hub functions other than the BACnet/SC hub function, such as an MQTT broker or other pub/sub mechanism, is expected to be standardized over time.

For the sake of efficiency, BACnet/SC nodes may bypass the hub function for sending directed messages by establishing direct WebSocket connections between each other.

BACnet/SC supports redundancy for the hub function via the concept of a failover hub function that can be deployed alongside the primary hub function. Nodes will connect to the failover hub function when they cannot connect to the primary hub function.

[Add new **Annex YY, BACnet Secure Connect,** p. 1348]

ANNEX YY - BACnet Secure Connect (NORMATIVE)

(This annex is part of this standard and is required for its use.)

This annex defines a data link protocol by which BACnet devices can transfer messages utilizing the WebSocket protocol as specified in RFC 6455. The Request For Comments (RFC) documents that define the WebSocket protocol are maintained by the Internet Engineering Task Force (IETF).

YY.1 BACnet Secure Connect Datalink

The BACnet Secure Connect, or BACnet/SC, datalink layer specifies a microprotocol enabling the use of WebSocket based connections, specifically the TLS-secured variant, for the exchange of BACnet messages between nodes.

The logical topology of a BACnet/SC network generally follows a hub-and-spoke model consisting of multiple BACnet/SC nodes and a hub function. See Figure YY-1. A BACnet/SC node wishing to participate in a BACnet/SC network establishes a hub connection to a hub function. This annex specifies the BACnet/SC hub function based on BACnet/SC connections which all BACnet/SC nodes shall be able to connect to.

Optionally, for transmitting directed BACnet messages, BACnet/SC nodes may support direct connections with other BACnet/SC nodes on the same BACnet network, as a BACnet/SC connection initiating peer, or as BACnet/SC connection accepting peer.

BACnet/SC connections use the secure variant of the WebSocket connections for bi-directional BACnet Virtual Link Layer Control (BVLC) messages.

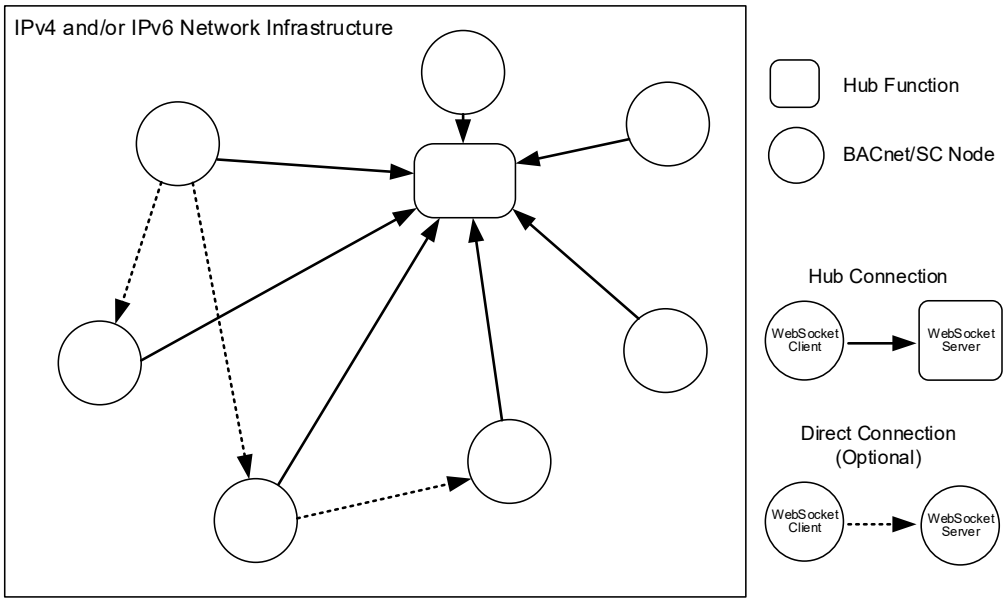


Figure YY-1. BACnet/SC Logical Network Topology

For enhanced availability of the central hub function for a BACnet/SC network, the hub connector specifies a failover hub concept in which the failover hub can be used in the case that the primary hub function is not available or not reachable.

YY.1.1 BACnet/SC Nodes

A BACnet/SC node is a network port that implements a BACnet/SC Virtual Link Layer (BVLL) entity for link control and NPDU transport, and the hub connector for connecting to the hub function to participate in the BACnet/SC network.

Figure YY-2 illustrates a BACnet Device implementing a BACnet/SC node.

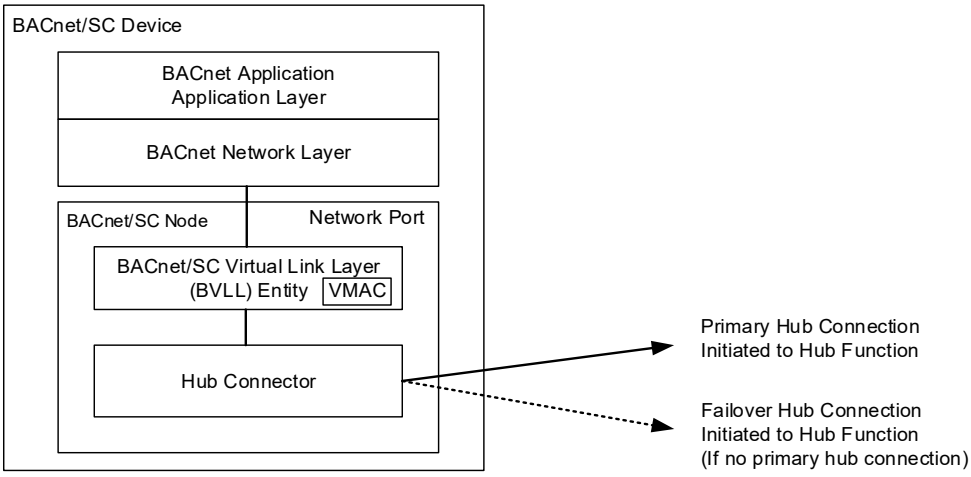


Figure YY-2. Example BACnet/SC Device

The BVLL for BACnet/SC defines the BACnet/SC Virtual Link Control (BVLC) messages that are used to control the virtual link and to convey BACnet NPDUs.

YY.1.1.1 BVLL Entity

The BVLL entity of a network port is the initiating and executing entity of BVLC messages and is identified in the BACnet/SC network by the VMAC of the node. See Clause YY.2.

YY.1.1.2 Hub Connector

The hub connector is required in a BACnet/SC node and maintains one initiated hub connection to a hub function at a time. For enhanced availability of the hub function of a BACnet/SC network, the hub connector shall support initiating a connection to the primary hub function, referred to as the primary hub connection, and shall support initiating a connection to the failover hub function, referred to as the failover hub connection and to be used when the primary hub function is not available. See Clause YY.3.

The Hub Connector shall support connecting to the BACnet/SC hub function by initiating BACnet/SC connections for the primary hub connection and for the failover hub connection.

YY.1.1.3 Optional Node Switch and Direct Connections

Optionally, a BACnet/SC node may support initiating or accepting WebSocket connections as BACnet/SC direct connections. See Clause YY.1.3. The support of direct connections requires a node switch function which is the endpoint of all WebSocket connections initiated or accepted by the BACnet/SC node as direct connections.

Figure YY-3 illustrates a BACnet Device implementing a BACnet/SC node with support of direct connections.

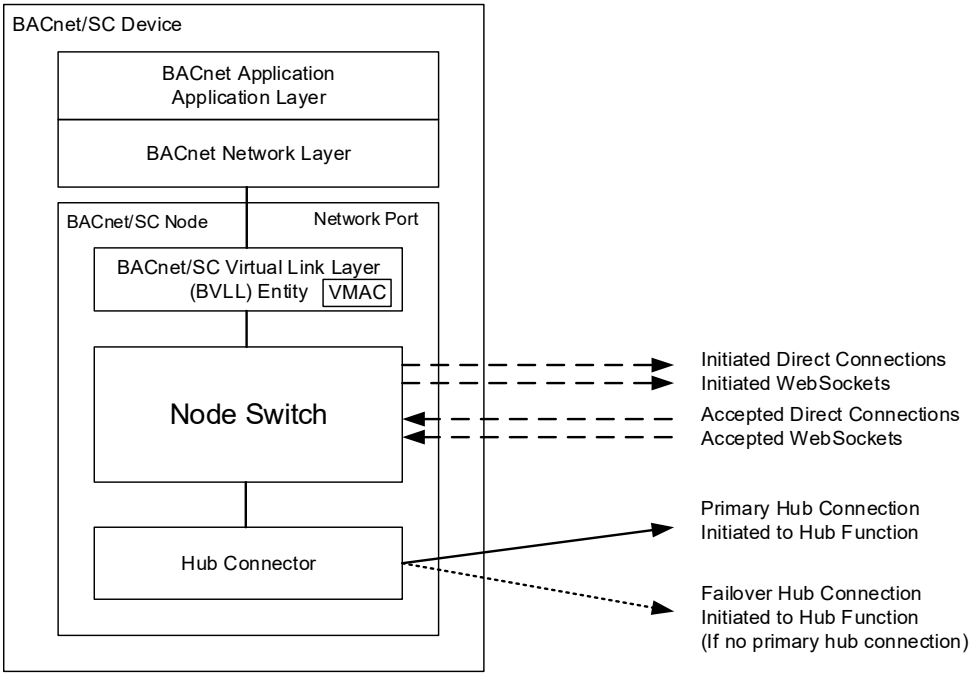


Figure YY-3. Example BACnet/SC Device Supporting Direct Connections

The BACnet/SC node switch function dispatches messages from the local BVLL entity to a direct connection or the hub connector, and from the hub connector or a direct connection to the local BVLL entity. For direct connections and the node switch function, see Clause YY.4.

YY.1.2 Hub Function

For every BACnet/SC network, one hub function is required. This hub function is referred to as the primary hub function for the BACnet/SC nodes.

Optionally, for enhanced availability, an additional hub function may be present and is used by the BACnet/SC nodes as the failover hub function. The distinction of which is the primary hub function, and which is the failover hub function, is a site specific determination, and configured into the BACnet/SC nodes accordingly.

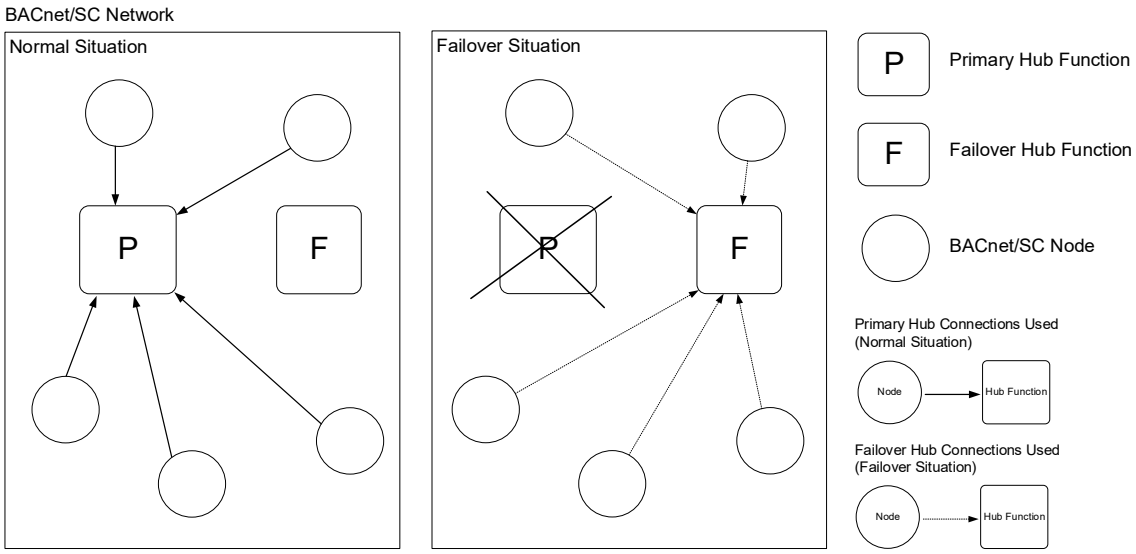


Figure YY-3. Example Failover Situation

The hub function forwards directed messages received from one hub connection to another hub connection and distributes broadcast messages to all hub connections.

This annex defines the BACnet/SC hub function that all BACnet/SC nodes are required to be able to connect to.

For the BACnet/SC hub function, see Clause YY.5.3. The BACnet/SC hub function accepts secure WebSocket connections as BACnet/SC connections. See Clause YY.1.3.

YY.1.3 BACnet/SC Connections

For each hub connection to the BACnet/SC hub function and for each direct connection, one WebSocket connection is used for one BACnet/SC connection for bi-directional BACnet Virtual Link Control (BVLC) message exchange.

After establishing a WebSocket connection, the BACnet/SC connection establishment phase detects error situations before the BACnet/SC connection is established and can be used for general and bi-directional BVLC message transmission. See Clause YY.6.1.

The WebSocket protocol as defined in RFC 6455 is used for establishment of WebSocket connections for BACnet/SC. See Clause YY.7.

YY.1.4 Service Specification

This clause describes the primitives and parameters associated with the services the BACnet/SC BVLL entity is providing to the BACnet network layer. The parameters are described in an abstract sense, which does not constrain the implementation method. Primitives and their parameters are described in a form that echoes their specification in ISO 8802-2. This is intended to provide a consistent interface to the BACnet network layer.

In addition to other datalink service primitives, these primitives support a 'data_attributes' parameter that specifies attributes to the 'data' parameter that can be forwarded by the BACnet network layer to reach the final destination of the Encapsulated-NPDU's payload. See Clause 6.6.

In BACnet/SC, the 'data_attributes' parameter is supported and conveys the data options to be sent or received.

YY.1.4.1 DL-UNITDATA.request

YY.1.4.1.1 Function

This primitive is the service request primitive for the unacknowledged connectionless-mode data transfer service.

YY.1.4.1.2 Semantics of the Service Primitive

The primitive shall provide parameters as follows:

DL-UNITDATA.request (
source_address,
destination_address,
data,
priority,
data_attributes
)

Each source and destination address consists of the logical concatenation of a medium access control (MAC) address and a link service access point (LSAP). For the case of BACnet/SC network ports, since the data link interface supports only the BACnet network layer, the LSAP is omitted and these parameters consist of only the VMAC address. See Clause YY.1.5.2.

The 'data' parameter specifies the link service data unit (LSDU) to be transferred by the BACnet/SC network port.

The 'priority' parameter specifies the priority desired for the data unit transfer. The priority parameter is ignored by BACnet/SC.

The 'data_attributes' parameter provides attributes for the content of the 'data' parameter. For a BACnet/SC network port, this parameter specifies the header options which shall be included in the 'Data Options' parameter in all BVLC messages resulting from this primitive.

YY.1.4.1.3 When Generated

This primitive is passed from the network layer to the BACnet/SC BVLL entity to request that a network protocol data unit (NPDU) be sent to one or more remote LSAPs using unacknowledged connectionless-mode procedures.

YY.1.4.1.4 Effect on Receipt

Receipt of this primitive causes the BACnet/SC BVLL entity to attempt to send the NPDU using unacknowledged connectionless-mode procedures.

YY.1.4.2 DL-UNITDATA.indication

YY.1.4.2.1 Function

This primitive is the service indication primitive for the unacknowledged connectionless-mode data transfer service.

YY.1.4.2.2 Semantics of the Service Primitive

The primitive shall provide parameters as follows:

DL-UNITDATA.indication (
source_address,
destination_address,
data,
priority,
data_attributes
)

Each source and destination address consists of the logical concatenation of a medium access control (MAC) address and a link service access point (LSAP). For the case of BACnet/SC devices, since the data link interface

supports only the BACnet network layer, the LSAP is omitted and these parameters consist of only the VMAC address. See Clause YY.1.5.2.

The 'data' parameter specifies the link service data unit (LSDU) received by the BACnet/SC network port.

The 'priority' parameter specifies the priority desired for the data unit transfer. The priority parameter is not provided by BACnet/SC.

The 'data_attributes' parameter provides attributes for the content of the 'data' parameter. For a BACnet/SC network port, this parameter includes the information that was received in the 'Data Options' parameter of the BVLC message received.

YY.1.4.2.3 When Generated

This primitive is passed from the BACnet/SC entity to the network layer to indicate the arrival of an NPDU from the specified remote entity.

YY.1.4.2.4 Effected on Receipt

The effect of receipt of this primitive by the network layer is specified in Clause 6.

YY.1.4.3 DL-RELEASE.request

YY.1.4.3.1 Function

This primitive is the service request primitive for the request to release the datalink state machine from waiting on a response message.

YY.1.4.3.2 Semantics of the Service Primitive

The primitive shall not provide any parameters as follows:

DL-RELEASE.request()

YY.1.4.3.3 When Generated

This primitive is passed from the network layer to the BACnet/SC BVLL entity to indicate that no reply is available from the higher layers.

YY.1.4.3.4 Effected on Receipt

In BACnet/SC, there is no effect of receipt of this primitive.

YY.1.5 Addressing within BACnet/SC Networks

YY.1.5.1 Network Location of Nodes

The network and resource location of the BACnet/SC hub function and the node switch of BACnet/SC nodes accepting direct connections are specified by WebSocket URIs as defined by the "wss" URI scheme in RFC 6455, Section 3.

The URIs to be used to connect to a hub function shall be configurable and shall be used only to connect to the hubs. See Clause YY.5.

For a direct connection to a node, the WebSocket URIs of that node where direct connections are accepted can be requested by initiating an Address-Resolution BVLC message sent to that node through the hub function. The Address-Resolution-ACK BVLC response can provide the possible WebSocket URIs where that node accepts WebSocket connections for BACnet/SC direct connections. See Clause YY.4.

Optionally, the WebSocket URIs for a direct connection to a node may be configured in the initiating node and shall take precedence over what is received in Address-Resolution-ACK messages from the responding node.

YY.1.5.2 VMAC Addressing of Nodes

For the BVLC message exchange, BACnet/SC nodes are identified by their 6-octet virtual MAC address as defined in Clause H.7.X.

For broadcast BVLC messages that need to reach all nodes of the BACnet/SC network, the destination VMAC address shall be the non-EUI-48 value X'FFFFFFFFFFFF', referred to as the Local Broadcast VMAC address.

The reserved EUI-48 value X'000000000000' is not used by this data link and therefore can be used internally to indicate that a VMAC is unknown or uninitialized.

YY.1.5.3 Device UUID

Every BACnet device that supports one or more BACnet/SC network ports shall have a Universally Unique ID (UUID) as defined in RFC 4122. This UUID identifies the device regardless of its current VMAC address or device instance number and is referred to as the device UUID.

This device UUID shall be generated before first deployment of the device in an installation, shall be persistently stored across device restarts, and shall not change over the entire lifetime of a device.

If a device is replaced in an installation, the new device is not required to re-use the UUID of the replaced device. For BACnet/SC, it is assumed that existing connections to the device being replaced are all terminated before the new device comes into operation.

YY.1.6 BACnet/SC Network Definition

A BACnet network based on the BACnet/SC datalink option is referred to as a BACnet/SC network. A BACnet/SC network is a set of two or more BACnet/SC nodes in which all nodes connect to the same primary hub function. In a BACnet/SC network, one hub function used as the primary hub shall be present. The presence of a hub function used as the failover hub is optional.

Direct connections between nodes shall only be established between nodes of the same BACnet/SC network. Nodes using direct connections shall remain connected to the hub. Broadcast BVLC messages shall always and only be sent to the hub function for distribution.

Only one direct connection shall exist at a time between any two BACnet/SC nodes, regardless of which node initiated or accepted the WebSocket connection.

YY.1.7 Remote MAC Addressing of Devices on BACnet/SC Networks

In BACnet network layer services, application layer services, and in data of type BACnetAddress, the MAC address of a BACnet/SC node shall be the node's virtual MAC address as defined in Clause YY.1.5.2.

YY.1.8 BACnet/SC Network Port Objects

Participation in a BACnet/SC network is represented by single network port regardless of the number of connections and initiated or accepted WebSocket connections in use by the network port.

For BACnet/SC network port implementations less than protocol revision 17, the configuration of BACnet/SC network ports is a local matter and cannot be represented by Network Port objects.

For BACnet/SC network port implementations with a protocol revision 17 and higher, BACnet/SC network ports shall be represented by a Network Port object at the BACNET_APPLICATION protocol level with a proprietary network type value. For the required standard properties to be present, see Clause 12.56.

YY.2 BACnet/SC Virtual Link Layer Messages

The BACnet/SC Virtual Link Layer (BVLL) provides the interface between the BACnet Network Layer (See Clause 6) and the underlying capabilities of the communication subsystem based on WebSockets (RFC 6455). This annex specifies the BACnet Virtual Link Control (BVLC) functions required to transport directed and broadcast messages, and to control the BVLL operation. The purpose and format of each BVLC message is described in the following subclauses. The BVLL behavior is defined in Clause YY.6.

The following table lists the BVLC messages defined for BACnet/SC.

Table YY-1 BACnet/SC BVLC Messages

BVLC Message	BVLC Function
X'00' BVLC-Result	Respond with ACK or NAK with error details
X'01' Encapsulated-NPDU	Convey an NPDU.
X'02' Address-Resolution	Request for the WebSocket URIs accepting direct connections.
X'03' Address-Resolution-ACK	Return WebSocket URIs accepting direct connections if any.
X'04' Advertisement	Inform about the sender node's current status
X'05' Advertisement-Solicitation	Request for the current status of the destination node.
X'06' Connect-Request	Request to accepting peer to accept a WebSocket connection for BACnet/SC
X'07' Connect-Accept	Response to initiating peer to accept a WebSocket connection for BACnet/SC
X'08' Disconnect-Request	Request and last message sent to request disconnection of the connection.
X'09' Disconnect-ACK	Response and last message sent to confirm disconnection to the connection peer
X'0A' Heartbeat-Request	Request a heartbeat from the connection peer.
X'0B' Heartbeat-ACK	Heartbeat response to connection peer.
X'0C' Proprietary-Message	Proprietary extension messages

Directed BVLC messages are addressed to a single destination node. Broadcast BVLC messages are addressed to all nodes of the BACnet/SC network and sent by a node to the hub function for distribution to all other nodes.

Response BVLC messages are directed messages and are returned as an immediate response to a BVLC message. The following are response messages: BVLC-Result, Address-Resolution-ACK, Connect-Accept in response to Connect-Request, Disconnect-ACK in response to Disconnect-Request, and Heartbeat-ACK in response to Heartbeat-Request. No response message shall be sent when a broadcast or response message is received.

YY.2.1 General BVLC Message Format

The following table shows the general BVLC message format for BACnet/SC.

Table YY-2 BACnet/SC BVLC Messages Structure

Field	Length	Description
BVLC Function	1-octet	BVLC function
Control Flags	1-octet	Determines presence of optional fields.
Message ID	2-octets	The message identifier
Originating Virtual Address	6-octets	Optional field, originating node VMAC address
Destination Virtual Address	6-octets	Optional field, destination VMAC address
Destination Options	Variable	Optional field, header options for the destination node
Data Options	Variable	Optional field, header options accompanying a payload containing data for upper layers
Payload	Variable	Optional field, the payload of the BVLC message

The 1-octet 'BVLC Function' field identifies the specific function to be carried out in support of the indicated communication subsystem or microprotocol type.

The 1-octet 'Control Flags' field indicates which optional parts are present. See Clause YY.2.2

The 2-octet 'Message ID' field is a numeric identifier of the message being sent. See Clause YY.3.1.3.

The optional 6-octet 'Originating Virtual Address' field indicates the VMAC address of the node that originally initiated the BVLC message. See Clause **Error! Reference source not found.**. If the sender of the message is also the originator of the message, then the 'Originating Virtual Address' field shall be omitted and the receiver shall assume the 'Originating Virtual Address' to be the VMAC of the sender.

The optional 6-octet 'Destination Virtual Address' field indicates the VMAC address of the destination node or the broadcast VMAC. See Clause **Error! Reference source not found.**. If the immediate receiver of a directed message is also the final destination of the message, then the 'Destination Virtual Address' field shall be omitted.

The optional and variable size 'Destination Options' field contains zero or more header options for the destination BACnet/SC node. See Clause YY.2.3.

The optional and variable size 'Data Options' field contains zero or more header options accompanying a data payload intended for upper layers. See Clause YY.2.3.

The remaining octets of the BVLC message, if any, are the variable size 'Payload' parameter conveying the payload of the BVLC message. See BVLC message definitions in Clause YY.2.4 and subsequent clauses.

All multi-octet numeric values are encoded with most significant octet first.

YY.2.2 Control Flags

The 'Control Flags' field indicates the presence or absence of optional fields in the BVLC message.

Bit 7..4	Reserved	Shall be zero.
Bit 3:	Originating Virtual Address Flag	1 = Originating Virtual Address is present 0 = Originating Virtual Address is absent
Bit 2:	Destination Virtual Address Flag	1 = Destination Virtual Address is present 0 = Destination Virtual Address is absent
Bit 1:	Destination Options Flag	1 = Destination Options field is present 0 = Destination Options field is absent
Bit 0:	Data Options Flag	1 = Data Options field is present 0 = Data Options field is absent

YY.2.3 Header Options

BVLC messages allow conveying header options in addition to defined payloads. Multiple header options with the same or different header option type may be present in each of the header options list parameters of a BVLC message.

The optional 'Destination Options' parameter is a list of header options. The header options in this list are addressed to the destination node or nodes addressed by the 'Destination VMAC Address' parameter.

The optional 'Data Options' parameter is a list of header options that accompany data payloads that are intended for upper layers. For standard BVLC messages, this parameter shall only be present in BVLC messages that convey an

NPDU, in which case, the header options in this list are associated with an NPDU that originates at the source BACnet device and accompany the NPDU to the ultimate destination device or devices. Because these are BACnet/SC options, they can only be conveyed to the ultimate destination device if that device is also a BACnet/SC device and the message has not passed through any non-BACnet/SC network segments while being routed.

Each header option includes a 'Header Marker' identifying the type of the option, a 'Header Length' field, and the 'Header Data' for the content of the header.

Header Marker	1-octet	Flags for the header option and numeric header option type.
Header Length	0 or 2-octets	Optional length of the 'Header Data' field, in octets. Present if and only if the 'Header Data Flag' flag is set (1).
Header Data	Variable	Optional octet string as defined for the header option type. Present if and only if the 'Header Data Flag' flag is set (1).

The 'Header Marker' octet includes the fields as follows:

Bit 7	More Options	1 = Another header option follows in the list. 0 = This is the last header option in the list.
Bit 6:	Must Understand	1 = This header option must be understood for consuming the message. 0 = This header option can be ignored if not understood.
Bit 5:	Header Data Flag	1 = The 'Header Length' and 'Header Data' fields are present 0 = The 'Header Length' and 'Header Data' fields are absent
Bits 4..0:	Header Option Type	1..31, The numeric header option type.

The 'More Options' flag indicates if the header option is the last option in the list (0), or at least one more header option follows in the header options list (1).

For the handling of the 'Must Understand' flag and the processing of header options when sending, forwarding, broadcasting, or receiving BVLC messages with header options, see Clause YY.3.1.4.

The following table lists the header option types defined by this standard and assigns the numeric header option type used in the 'Header Marker'.

Table YY-3 BVLC Header Options

Header Option Type	Numeric Header Option Type	Description
Secure Path	1	See Clause YY.2.3.1
Proprietary Header Option	31	See Clause YY.2.3.2

All other header options and numeric header option types are reserved for definition by ASHRAE.

The optional 2-octet 'Header Length' field indicates the length in octets of the 'Header Data' field. It shall be present if and only if the 'Header Data Flag' of the header marker is set (1).

The optional and variable size 'Header Data' field is an octet string whose content is defined by the respective header option type indicated by the 'Header Marker'. Shall be present if and only if the 'Header Data Flag' of the header marker is set (1). If zero data octets are present, the 'Header Data' field is considered empty.

YY.2.3.1 Secure Path Header Option

The ‘Secure Path’ header option specifies, by its presence, whether the service being requested represents a message which has only been transferred by BACnet/SC datalinks and secure connect BACnet routers.

The 'Secure Path' header option consists of the following fields.

Header Marker	1-octet	'Last Option' = 0 or 1, 'Must Understand' = 1, 'Header Data Flag' = 0, 'Header Option Type' = 1

This header option, if present, shall be a data option in the 'Data Options' parameter of BVLC messages conveying an NPDU. This header option shall be initially provided by the network or application entity initiating the payload of the NPDU being conveyed. It shall remain with the NPDU as long as the message does not pass through any non-BACnet/SC network segments while being routed. The processing of this information when received by the NPDU's payload final destination device's network or application entity is a local matter.

YY.2.3.2 Proprietary Header Options

Vendors may define and use proprietary header options. In order to distinguish vendor specific header options, the first two octets of the header data shall contain the vendor identifier code of the defining organization. See Clause 23.

Any proprietary header option shall consist of the following fields:

Header Marker	1-octet	'More Options' = 0 or 1, 'Must Understand' = 0 or 1, 'Header Data Flag' = 1, 'Header Option Type' = 31
Header Length	2-octets	Length of 'Header Data' field, in octets.
Header Data	3-N octets	Required to include:
Vendor Identifier	2-octets	Vendor Identifier, with most significant octet first, of the organization defining this option.
Proprietary Option Type	1-octet	An indication of the proprietary header option type.
Proprietary Header Data	Variable	A proprietary string of octets. Can be zero length.

For BVLC messages received, the processing of proprietary header options is a local matter.

For BVLC messages sent, the insertion of proprietary header options in the BVLC message is a local matter.

YY.2.4 BVLC-Result

This directed BVLC message provides a mechanism to acknowledge the result of those BVLL service requests that require an acknowledgment, whether successful (ACK) or unsuccessful (NAK). This message is the result of a BVLC function request and is not a response to the payload or data options. This response message is generated by a BACnet/SC node's BVLL entity and shall not convey data options.

YY.2.4.1 BVLC-Result Format

The BVLC-Result message consists of the following fields:

BVLC Function	1-octet	X'00'	BVLC-Result
Control Flags	1-octet		Control flags.
Message ID	2-octets		The message identifier of the message for which this message is the result.
Originating Virtual Address	0 or 6-octets		If absent, message is from connection peer node
Destination Virtual Address	0 or 6-octets		If absent, message is for connection peer node
Destination Options	Variable		Optional, 0 to N header options
Data Options	0-octets		Shall be absent.
Payload			
Result For BVLC Function	1-octet	Function	BVLC function for which this is a result
Result Code	1-octet	X'00'	ACK: Successful completion. The 'Error Header Marker' and all subsequent parameters shall be absent.
		X'01'	NAK: The BVLC function failed. The 'Error Header Marker', the 'Error Class', the 'Error Code', and the 'Error Details' shall be present.
Error Header Marker (Conditional)	1-octet		The header marker of the destination option that caused the BVLC function to fail. If the NAK is unrelated to a header option, this parameter shall be X'00'.
Error Class (Conditional)	2-octets		The 'Error Class' field of the 'Error' datatype defined in Clause 21.
Error Code (Conditional)	2-octets		The 'Error Code' field of the 'Error' datatype defined in Clause 21.
Error Details (Conditional)	Variable		UTF-8 encoded reason text. Can be empty.

YY.2.5 Encapsulated-NPDU

This directed or broadcast BVLC message is used to send directed NPDUs to another BACnet/SC node, or broadcast NPDUs to all nodes.

YY.2.5.1 Encapsulated-NPDU Format

The Encapsulated-NPDU message consists of the following fields:

BVLC Function	1-octet	X'01'	Encapsulated-NPDU
Control Flags	1-octet		Control flags
Message ID	2-octets		The message identifier
Originating Virtual Address:	0 or 6-octets		If absent, message is from connection peer node
Destination Virtual Address	0 or 6-octets		If absent, message is for connection peer node
Destination Options	Variable		Optional, 0 to N header options
Data Options	Variable		Optional, 0 to N header options
Payload			
BACnet NPDU	Variable		

YY.2.6 Address-Resolution

This directed BVLC message is sent by BACnet/SC nodes to request the list of possible WebSocket URIs at which the destination node accepts direct connections. See Clause YY.4.

YY.2.6.1 Address-Resolution Format

The Address-Resolution message consists of the following fields:

BVLC Function	1-octet	X'02'	Address-Resolution

Control Flags	1-octet		Control flags
Message ID	2-octets		The message identifier
Originating Virtual Address	0 or 6-octets		If absent, message is from connection peer node
Destination Virtual Address	0 or 6-octets		If absent, message is for connection peer node
Destination Options	Variable		Optional, 0 to N header options
Data Options	0-octets		Shall be absent.

YY.2.7 Address-Resolution-ACK

This directed BVLC message is the response to the Address-Resolution message. The Address-Resolution-ACK message is directed to the node that originally initiated the Address-Resolution message. See Clause YY.4.1.

YY.2.7.1 Address-Resolution-ACK Format

The Address-Resolution-ACK message consists of the following fields:

BVLC Function	1-octet	X'03'	Address-Resolution-ACK
Control Flags	1-octet		Control flags
Message ID	2-octets		The message identifier of the message for which this message is the response.
Originating Virtual Address	0 or 6-octets		If absent, message is from connection peer node
Destination Virtual Address	0 or 6-octets		If absent, message is for connection peer node
Destination Options	Variable		Optional, 0 to N header options
Data Options	0-octets		Shall be absent.
Payload			
WebSocket-URIs	Variable		UTF-8 string containing a list of WebSocket URIs as of RFC 3986, separated by a single space character (X'20'), where the source BACnet/SC node accepts direct connections.

YY.2.8 Advertisement

This directed BVLC message is advertising the configuration and status of the source node. See Cause YY.3.2.

YY.2.8.1 Advertisement Format

The Advertisement message consists of the following fields:

BVLC Function:	1-octet	X'04'	Advertisement
Control Flags	1-octet		Control flags
Message ID	2-octets		The message identifier
Originating Virtual Address:	0 or 6-octets		If absent, message is from connection peer node
Destination Virtual Address:	0 or 6-octets		If absent, message is for connection peer node
Destination Options	Variable		Optional, 0 to N header options
Data Options	0-octets		Shall be absent.
Payload			
Hub Connection Status	1-octet	X'00'	No hub connection.
		X'01'	Connected to primary hub.
		X'02'	Connected to failover hub.
Accept Direct Connections	1-octet	X'00'	The node does not support accepting direct connections.
		X'01'	The node supports accepting direct connections.
Maximum BVLC Length	2-octet		The maximum BVLC message size that can be received and processed by the node, in number of octets.
Maximum NPDU Length	2-octets		The maximum NPDU message size that can be handled by the node's network entity, in number of octets.

YY.2.9 Advertisement-Solicitation

This directed BVLC message is sent to a node to solicit that node to send an Advertisement message in a manner that the requesting node can receive.

YY.2.9.1 Advertisement-Solicitation Format

The Advertisement-Solicitation message consists of the following fields:

BVLC Function:	1-octet	X'05'	Advertisement-Solicitation
Control Flags	1-octet		Control flags
Message ID	2-octets		The message identifier
Originating Virtual Address:	0 or 6-octets		If absent, message is from connection peer node
Destination Virtual Address:	0 or 6-octets		If absent, message is for connection peer node
Destination Options	Variable		Optional, 0 to N header options
Data Options	0-octets		Shall be absent.

YY.2.10 Connect-Request

This directed BVLC message is sent to the connection accepting peer node to request acceptance of the connection established. See Clause YY.6.2.

YY.2.10.1 Connect-Request Format

The Connect-Request message consists of the following fields:

BVLC Function:	1-octet	X'06'	Connect-Request
Control Flags	1-octet		Control flags
Message ID	2-octets		The message identifier
Originating Virtual Address:	0-octets		Absent, is always from connection peer node
Destination Virtual Address:	0-octets		Absent, is always for connection peer node
Destination Options	Variable		Optional, 0 to N header options
Data Options	0-octets		Shall be absent.
Payload			
VMAC Address	6-octets		The VMAC address of the requesting node.
Device UUID	16-octet		The device UUID of the requesting node
Maximum BVLC Length	2-octet		The maximum BVLC message size that can be received and processed by the requesting node, in number of octets.
Maximum NPDU Length	2-octets		The maximum NPDU message size that can be handled by the requesting node's network entity, in number of octets.

YY.2.11 Connect-Accept

This directed BVLC message is sent to the connection requesting peer node to confirm acceptance of the connection established. See Clause YY.6.2.

YY.2.11.1 Connect-Accept Format

The Connect-Accept message consists of the following fields:

BVLC Function:	1-octet	X'07'	Connect-Accept
Control Flags	1-octet		Control flags
Message ID	2-octets		The message identifier
Originating Virtual Address:	0-octets		Absent, is always from connection peer node
Destination Virtual Address:	0-octets		Absent, is always for connection peer node
Destination Options	Variable		Optional, 0 to N header options
Data Options	0-octets		Shall be absent.
Payload			
VMAC Address	6-octets		For direct connections, the VMAC of the accepting node. For hub connections, the VMAC of the node in the network port that contains the hub function.
Device UUID	16-octets		For direct connections, the device UUID of the accepting node. For hub connections, the UUID of the device that contains the network port that contains the hub function.
Maximum BVLC Length	2-octets		The maximum BVLC message size that can be received and processed by the accepting node, in number of octets.
Maximum NPDU Length	2-octets		The maximum NPDU message size that can be handled by the accepting node's network entity, in number of octets.

YY.2.12 Disconnect-Request

This directed BVLC message is sent to the connection peer node to request disconnection of the connection. See Clause YY.6.2.

YY.2.12.1 Disconnect-Request Format

The Disconnect-Request message consists of the following fields:

BVLC Function:	1-octet	X'08'	Disconnect-Request
Control Flags	1-octet		Control flags
Message ID	2-octets		The message identifier
Originating Virtual Address:	0-octets		Absent, is always from connection peer node
Destination Virtual Address:	0-octets		Absent, is always for connection peer node
Destination Options	Variable		Optional, 0 to N header options
Data Options	0-octets		Shall be absent.

YY.2.13 Disconnect-ACK

This directed BVLC message is sent to the connection peer node to confirm the disconnection. See Clause YY.6.2.

YY.2.13.1 Disconnect-ACK Format

The Disconnect-ACK message consists of the following fields:

BVLC Function:	1-octet	X'09'	Disconnect-ACK
Control Flags	1-octet		Control flags
Message ID	2-octets		The message identifier
Originating Virtual Address:	0-octets		Absent, is always from connection peer node
Destination Virtual Address:	0-octets		Absent, is always for connection peer node
Destination Options	Variable		Optional, 0 to N header options
Data Options	0-octets		Shall be absent.

YY.2.14 Heartbeat-Request

This directed BVLC message is sent to the connection peer node to probe that the connection and connection peer node is still alive. See Clause YY.6.3.

YY.2.14.1 Heartbeat-Request Format

The Heartbeat-Request message consists of the following fields:

BVLC Function:	1-octet	X'0A'	Heartbeat-Request
Control Flags	1-octet		Control flags
Message ID	2-octets		The message identifier
Originating Virtual Address:	0-octets		Absent, is always from connection peer node
Destination Virtual Address:	0-octets		Absent, is always for connection peer node
Destination Options	Variable		Optional, 0 to N header options
Data Options	0-octets		Shall be absent.

YY.2.15 Heartbeat-ACK

This directed BVLC message is sent to the connection peer node to indicate that the sending node and the connection is alive. See Clause YY.6.3.

YY.2.15.1 Heartbeat-ACK Format

The Heartbeat-ACK message consists of the following fields:

BVLC Function:	1-octet	X'0B'	Heartbeat-ACK
Control Flags	1-octet		Control flags
Message ID	2-octets		The message identifier
Originating Virtual Address:	0-octets		Absent, is always from connection peer node
Destination Virtual Address:	0-octets		Absent, is always for connection peer node
Destination Options	Variable		Optional, 0 to N header options
Data Options	0-octets		Shall be absent

YY.2.16 Proprietary Message

This directed or broadcast BVLC message is for proprietary extensions at the data link level. The payload portion of the BVLL message shall always have a vendor identifier, a proprietary function identifier defined by that vendor, and an optional proprietary data field. The use and processing of proprietary messages is a local matter. Recipients of unexpected proprietary messages can either drop the message or respond with a negative BVLC-Result. The error class and code to use for negative responses is a local matter, however, note that the error code BVLC_PROPRIETARY_FUNCTION_UNKNOWN is available as a generic response.

YY.2.16.1 Proprietary Message Format

The Proprietary-Request message consists of the following fields:

BVLC Function:	1-octet	X'0C'	Proprietary-Message
Control Flags	1-octet		Control flags
Message ID	2-octets		The message identifier
Originating Virtual Address:	0 or 6-octets		If absent, message is from connection peer node
Destination Virtual Address:	0 or 6-octets		If absent, message is for connection peer node
Destination Options	Variable		Optional, 0 to N header options
Data Options	Variable		Shall be absent.
Payload	3-N octets		The payload shall consist of at least the vendor identifier and the proprietary function octet.
Vendor Identifier	2-octets		Vendor Identifier, with most significant octet first, of the organization defining this message.
Proprietary Function	1-octet		The vendor-defined function code.
Proprietary Data	Variable		Optional vendor-defined payload data

YY.3 BACnet/SC Node Operation

YY.3.1 BVLC Message Exchange

YY.3.1.1 Response BVLC Messages

A BVLC-Result received shall not generate a BVLC-Result message in response. The handling of a BVLC-Result message received that cannot be matched to a BVLC request message sent is a local matter.

YY.3.1.2 Virtual Address Parameters in BVLC Messages

In every BVLC message, the 'Originating Virtual Address', if present, shall always be the VMAC of the node that originally initiated the BVLC message. If absent, the message originated at the connection peer node. If forwarded, the connection peer node's VMAC shall be inserted as the 'Originating Virtual Address' parameter.

If a BVLC message is sent to all nodes of the BACnet/SC network, the Local Broadcast VMAC address as defined in Clause YY.1.5.2 shall be used as the 'Destination Virtual Address' parameter. In this case, the 'Destination Virtual Address' parameter shall always remain present in the BVLC message so the recipient can determine if the message was directed or broadcast.

If a directed BVLC message is sent to a destination BACnet/SC node other than the connection peer, the VMAC of the destination node shall be used as the 'Destination Virtual Address' parameter. If a directed BVLC message is sent to the connection peer node, the 'Destination Virtual Address' parameter shall be absent.

If a response BVLC message is retuned on a BVLC message with no 'Originating Virtual Address' parameter, then the response BVLC message 'Destination Virtual Address' parameter shall be absent, and the response BVLC message shall be sent through the connection from which that BVLC message was received. In all other response messages, the 'Destination Virtual Address' shall be the VMAC indicated in the 'Originating Virtual Address' parameter of the message being responded to. Note that an Advertisement message is not considered a "response" to an Advertisement-Solicitation message; however, the solicited Advertisement shall be sent in a manner that the soliciting node will receive it using the rules above.

YY.3.1.3 Message ID Parameter

For BVLC messages originally initiated by the node, the determination of the 'Message ID' parameter value is a local matter.

To allow for matching the BVLC messages sent with response BVLC messages received, the message ID may be selected to be unique among all pending initiated BVLC messages within some maximum time the node waits for a response. The maximum time to wait for a response is a local matter.

For response BVLC messages, the message ID shall be the message ID of the causing message. Note that an Advertisement message is not considered a "response message" to an Advertisement-Solicitation message and does not copy the message ID of the solicitation.

YY.3.1.4 Header Options Processing and 'Must Understand'

The destination BACnet/SC node shall process the header options present in 'Destination Options'. Destination options whose 'Must Understand' flag is cleared (0) shall be ignored when not supported.

If a destination option is present whose 'Must Understand' flag is set (1) but the option is unknown or not supported by the BVLL entity of the destination node, then if the original message was a directed BVLC message, a BVLC-Result NAK for the BVLC message shall be returned indicating an 'Error Class' of COMMUNICATION and an 'Error Code' of HEADER_NOT_UNDERSTOOD. If the original message was a broadcast BVLC message, no BVLC-Result message shall be returned. The broadcast BVLC message shall be ignored.

For the handling of 'Data Options', see Clause YY.3.4. The hub function and the source and destination node's BVLL entity shall forward and not alter any of the data options.

The remaining parts of the BVLC message shall be processed as required.

YY.3.1.5 Common Error Situations

If a BVLC message is received that is truncated, for example, there are missing fields or incomplete fields, a BVLC-Result NAK shall be returned if it was a directed message, indicating an 'Error Class' of COMMUNICATON and 'Error Code' of MESSAGE_INCOMPLETE. The message shall be discarded and not be processed.

If a BVLC message is received that is an unknown BVLC function, a BVLC-Result NAK shall be returned if it was a directed message indicating an 'Error Class' of COMMUNICATON and 'Error Code' of BVLC_FUNCTION_UNKNOWN. The message shall be discarded and not be processed.

If a BVLC message is received for which a payload is required, but no payload is present, a BVLC-Result NAK shall be returned if it was a directed message indicating an 'Error Class' of COMMUNICATON and 'Error Code' of PAYLOAD_EXPECTED. The message shall be discarded and not be processed.

If a BVLC message is received in which a header has encoding errors, a BVLC-Result NAK shall be returned if it was a directed message indicating an 'Error Class' of COMMUNICATON and 'Error Code' of HEADER_ENCODING_ERROR. The message shall be discarded and not be processed.

If a BVLC message is received in which any control flag has an unexpected value, then a BVLC-Result NAK shall be returned if it was a directed message, indicating an 'Error Class' of COMMUNICATION and an 'Error Code' of PARAMETER_OUT_OF_RANGE. The message shall be discarded and not be processed.

If a BVLC message is received in which any parameter, field of a known header, or parameter in a BACnet/SC defined payload, is out of range, then a BVLC-Result NAK shall be returned if it was a directed message, indicating an 'Error Class' of COMMUNICATION and an 'Error Code' of PARAMETER_OUT_OF_RANGE. The message shall be discarded and not be processed.

If a BVLC message is received in which any data inconsistency exists in any parameter, field of a known header, or parameter in a BACnet/SC defined payload, then a BVLC-Result NAK shall be returned, indicating an 'Error Class' of COMMUNICATION and an 'Error Code' of INCONSISTENT_PARAMETERS. The message shall be discarded and not be processed.

YY.3.2 Advertisement Exchange

Nodes may initiate Advertisement or Advertisement-Solicitation messages to other nodes at any time, e.g., for synchronization or update of status information.

On receipt of an Advertisement message, the node shall update its status information of the sending node as provided by the Advertisement message.

On receipt of an Advertisement-Solicitation message, the node shall respond with an Advertisement message. Note that even though the Advertisement message is sent in response to the solicitation, it is not considered a "response message" and thus does not copy the message ID of the Address-Solicitation message.

YY.3.3 Address Resolution

An Address-Resolution message can be initiated to another node for requesting an Address-Resolution-ACK response returning the WebSocket URIs where the responding node accepts direct connections. The WebSocket URIs returned may or may not be valid for the network context of the requesting node. See Cause YY.1.5.1.

On receipt of an Address-Resolution message, an Address-Resolution-ACK message shall be returned if the node accepts direct connections. An empty string shall be returned if no such WebSocket URIs are currently known but the node supports accepting direct connections.

If accepting direct connections is not supported, a BVLC-Result NAK for the Address-Resolution message shall be returned, indicating an 'Error Class' of COMMUNICATION and an 'Error Code' of OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED.

YY.3.4 NPDU Exchange

The following message exchange procedures are performed by the BVLL entity of nodes for BVLC messages conveying an NPDU as payload.

On receipt of an Encapsulated-NPDU BVLC message from the node switch function or hub connector, the NPDU shall be extracted and forwarded to the local network entity in the 'data' parameter of the datalink indication primitive. Data options present in the message shall be forwarded to the local network entity in the 'data_attributes' parameter of the DL_UNITDATA.indication primitive.

On receipt of an NPDU from the local network entity in the 'data' parameter of the DL_UNITDATA.request primitive, and a destination VMAC address is provided; the BVLL entity shall create an Encapsulated-NPDU BVLC message with the NPDU as payload and forward it to the hub connector or node switch if present to send the message to the destination node. The 'Destination Virtual Address' shall be the destination VMAC address provided. Data options as provided by the network entity in the 'data_attributes' parameter shall be the 'Data Options' parameter in the Encapsulated-NPDU message.

On receipt of an Encapsulated-NPDU from the local network entity and an empty destination MAC address is provided; the BVLL shall create an Encapsulated-NPDU message with the NPDU as payload and the Local Broadcast VMAC as the 'Destination Virtual Address' and provide it to the hub connector for being sent. Data options as provided by the network entity in the 'data_attributes' parameter shall be the 'Data Options' parameter in the Encapsulated-NPDU message forwarded to the hub connector.

YY.4 Node Switch and Direct Connections

BACnet/SC nodes can optionally support BACnet/SC connections between BACnet/SC nodes of a BACnet/SC network for direct connections, in addition to the hub connections. BACnet/SC node implementations can optionally support initiating direct connections, or accepting direct connections, or both.

BACnet/SC nodes supporting direct connections are required to implement the BACnet/SC node switch function.

Only directed BVLC messages addressed to the connection peer node shall be sent through direct connections. All other BVLC messages are required to be sent through the hub connection to the hub function.

Nodes may optionally accept direct connections as an accepting peer and may optionally initiate direct connections to other nodes as an initiating peer.

If a node supports direct connections as an initiating peer, the method to determine when to initiate a direct connection is a local matter.

YY.4.1 URIs For Direct Connections

The WebSocket URIs for initiating a direct connection to an accepting peer shall use the "wss" scheme.

The WebSocket URIs to use for direct connections may be statically configured or dynamically discovered. If no WebSocket URIs are statically configured for a particular node, then the WebSocket URIs can be requested from that node by sending an Address-Resolution message to the node through the hub function.

If WebSocket URIs are provided with the Address-Resolution-ACK message in response, these WebSocket URIs can be used to attempt a direct connection to the node. The selection of the URI to use from those returned in the message is a local matter. The WebSocket URIs returned are not required to be valid for the network location and context of the Address-Resolution message initiating peer. If none of the returned URI's result in a connection, then a direct connection cannot be established; however, communication to the node through the hub function is still available.

YY.4.2 Node Switch Function

The BACnet/SC node's optional switch function is the peer for the direct connections of the BACnet/SC node. The node switch function forwards BVLC messages between the direct connections, the hub connector, and the local BVLL entity. See Figure YY-4. All direct connections are required to be established as BACnet/SC connections. See Clause YY.6.2.

The 'Destination Virtual Address' of BVLC messages received from the local BVLL entity is used to select the direct connection, the hub connector, or the local BVLL entity to forward a message to.

The node switch function shall maintain knowledge of the connection peer node's VMAC address and the connection peer node's Device UUID while the connection is established. This information is learned from the connect message exchange for the BACnet/SC connection after establishment of the WebSocket connection. See Clause YY.6.2.

For direct connection messages, both the source and destination VMAC address shall be omitted.

The node switch function dispatches messages between the local BVLL entity, direct connections, and the hub connector. See Figure YY-4 showing the node switch function of an example node that is connected to a hub function and has initiated one direct connection and has accepted two direct connections.

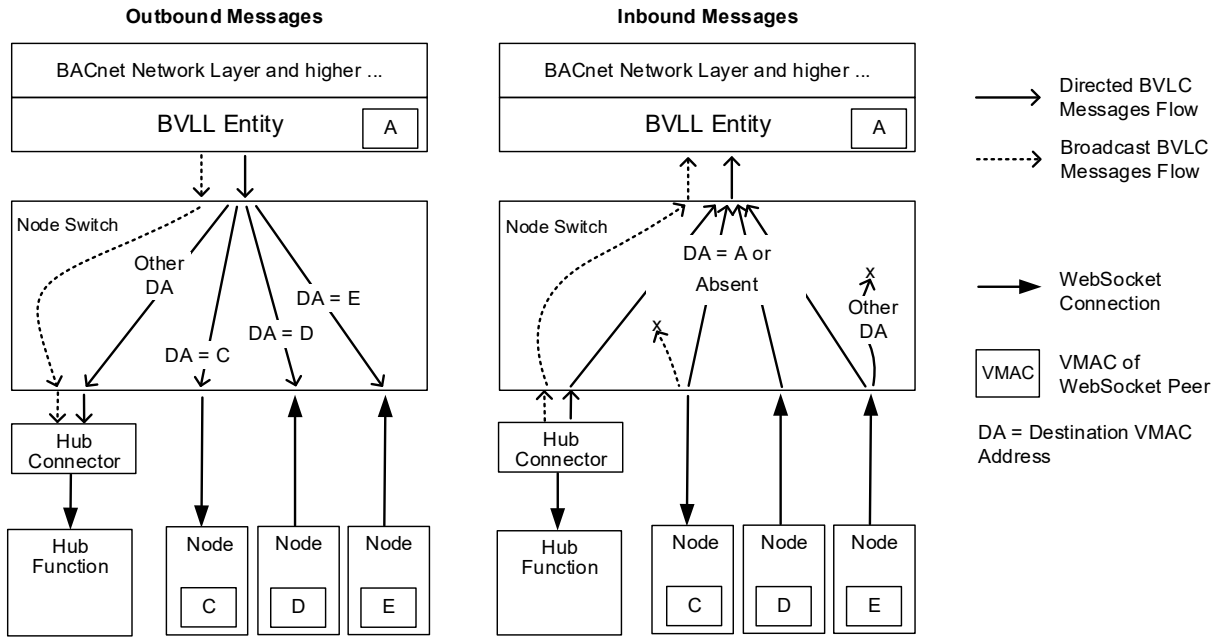


Figure YY-4. Node Switch Function

YY.4.2.1 Outbound Messages

On receipt of a directed BVLC message from the local BVLL entity and a destination VMAC address is provided, the directed BVLC message shall be sent through the direct connection with the connection peer node matching the destination VMAC, if one exists. If no such connection exists, the directed BVLC message shall be sent to the hub connector.

On receipt of a directed BVLC message from the local BVLL entity where the destination VMAC address is omitted and the direct connection to use is indicated by the local BVLL entity, the message shall be sent through that direct connection. The method of indication of the direct connection to be used is a local matter.

On receipt of a broadcast BVLC message from the local BVLL entity, the BVLC message shall be sent to the hub connector.

Directed BVLC messages being sent through a direct connection shall omit both the 'Destination VMAC Address', and the 'Originating VMAC Address' parameters.

All BVLC messages forwarded to the hub connector shall include both the 'Destination VMAC Address' parameter and, if required by the hub connector, the 'Originating VMAC Address' parameter, where the latter shall be the VMAC address of the BVLL entity. Note that the hub connector defined in Clause YY.5.3.4 does not need the 'Originating VMAC Address' parameter since it is not included in messages sent to the hub function because the BACnet/SC hub function already knows the VMAC from the Connect-Request message.

YY.4.2.2 Inbound Messages

On receipt of a directed BVLC message from any current direct connection or the hub connector whose destination VMAC is the VMAC of the local BVLL entity, or the destination VMAC address is absent, the message shall be forwarded to the local BVLL entity. All other directed BVLC messages shall be discarded.

On receipt of a broadcast BVLC message from the hub connector, the message shall be forwarded to the local BVLL entity.

On receipt of a broadcast BVLC message from a direct connection, the message shall be discarded.

For directed BVLC messages received from a direct connection whose destination VMAC address is absent, the hub switch shall indicate the VMAC address of the local BVLL entity as the destination VMAC address to the local BVLL entity.

For directed BVLC messages received from a direct connection whose originating VMAC address is absent, the hub switch shall forward the connection peer node's VMAC address as the originating VMAC address to the local BVLL entity.

YY.5 Hub Function and Hub Connector

For BACnet/SC networks, forwarding and distribution of BVLC messages among and between the BACnet/SC nodes is required. This functionality is performed by the hub function and the hub connectors of the BACnet/SC nodes connecting to the hub function.

The hub function can be used by nodes as the primary or failover hub function. This distinction is made by the BACnet/SC nodes only, and the hub function is equal for both. Different types of hub functions can be used for the primary and failover hub functions if the hub connectors of all BACnet/SC nodes used in the BACnet/SC network support both types.

The hub connector of the BACnet/SC node is related to the type of hub function being connected to, and the method of connection. For interoperability, this annex defines the BACnet/SC hub function based on BACnet/SC connections. All hub connectors shall support connection to the BACnet/SC hub function.

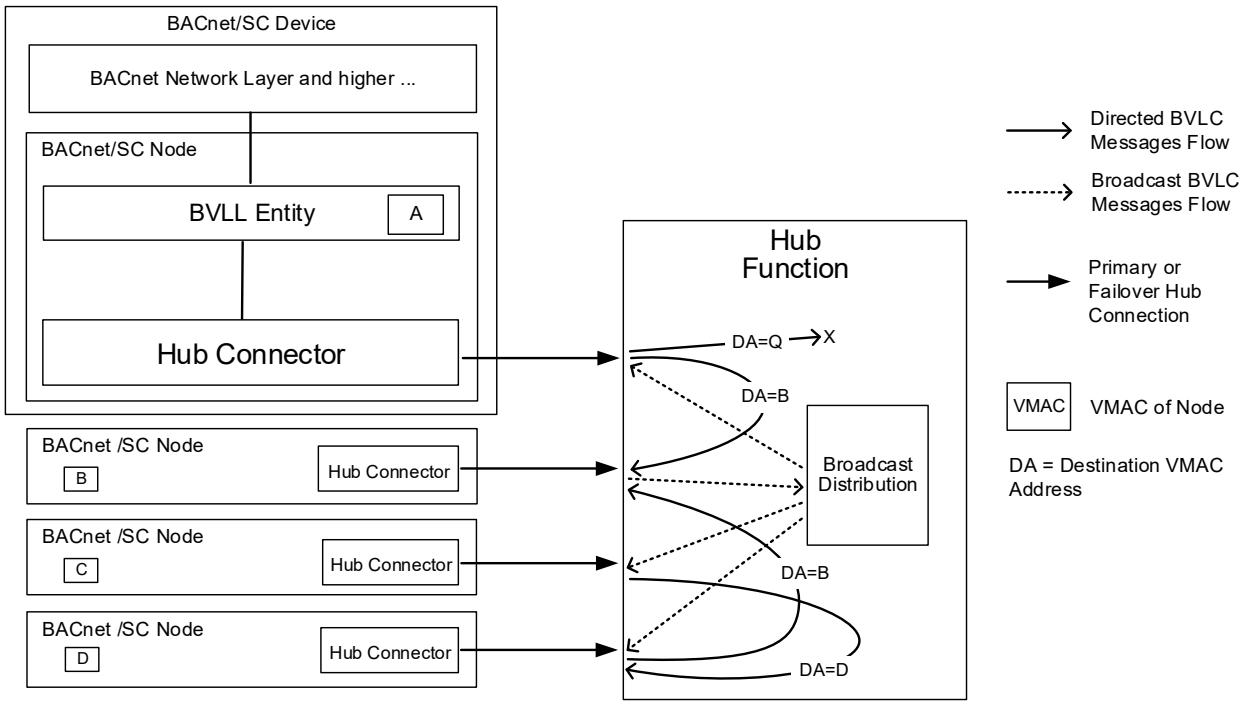


Figure YY-5. Hub Function Overview

YY.5.1 Hub Function Requirements

The hub function is required to accept hub connections initiated by the hub connector of BACnet/SC nodes.

Directed BVLC messages received from a hub connection shall be forwarded by the hub function to the hub connection where the destination VMAC address matches the VMAC address of the connection peer node. If there is no match, the directed BVLC message shall be discarded.

Broadcast BVLC messages received from a hub connection shall be duplicated and a copy shall be sent to all hub connections except the one from which it was received.

The hub function shall support the forwarding and distribution of BVLC messages that convey, at the least, NPDU sizes of 1497 octets and 4192 octets of data options and destination options.

The hub function shall not send a BVLC message, or any copy of it, to the hub connection from which it was received.

YY.5.2 Hub Connector Requirements

The hub function and method of connection to use to connect to the hub function is expected to be indicated by the URI scheme used for the primary and failover hub URIs configured for the hub connector.

The hub connector of the BACnet/SC node shall support configuration of the WebSocket URI for the primary hub function and the WebSocket URI for the failover hub function.

The hub connector of every BACnet/SC node shall support connecting to the BACnet/SC hub function as the primary hub function or the failover hub function. The hub connector shall initiate BACnet/SC connections to the BACnet/SC hub function as hub connections. See Clause YY.7.

The URI for connecting to a BACnet/SC hub function is identified by the "wss" scheme. A local BACnet/SC hub function is referenced by a "wss" scheme URI with "localhost" used as the hostname.

The hub connector shall establish and maintain a hub connection to the hub function indicated by the URI configured for the primary hub. If a hub connection to the primary hub function cannot be established, the hub connector shall attempt to establish a hub connection to the hub function indicated by the URI for the failover hub, if configured. While the hub connection to the failover hub function is established, attempts to re-establish the hub connection to the primary hub function shall be continued respecting the reconnect timeout. If an established primary hub connection is lost, the hub connector shall first attempt to re-establish the primary hub connection.

One established hub connection shall be maintained and used at a time. If the connection to the primary hub function can be restored, the failover hub connection shall be terminated.

The hub connector shall forward BVLC messages from the local BVLL entity, via the node switch if present, to the hub connection currently established.

The hub connector shall forward BVLC messages received from the established hub connection to the local BVLL entity, or to the node switch, if present, which will then forward the message to the BVLL entity.

YY.5.3 BACnet/SC Hub Function

The BACnet/SC Hub Function accepts BACnet/SC connections as hub connections and performs the hub function. The BACnet/SC hub function is conceptually present in a network port that includes a BACnet/SC node. The BACnet/SC hub function acts as an independent endpoint of BACnet/SC connections that are used exclusively as hub connections. The local BACnet/SC node hub connector connects to the local hub function equally as to a remote hub function.

YY.5.3.1 Hub Connections

The BACnet/SC hub function accepts BACnet/SC connections from BACnet/SC hub connectors. The hub function accepts at most one hub connection from each BACnet/SC node.

The URI of where the BACnet/SC hub function accepts BACnet/SC connections as hub connections is a WebSocket URI identified by the "wss" scheme.

Figure YY-5 illustrates a BACnet/SC Device that also includes a BACnet/SC hub function. This hub function is used as the primary hub function by the BACnet/SC node of the network port.

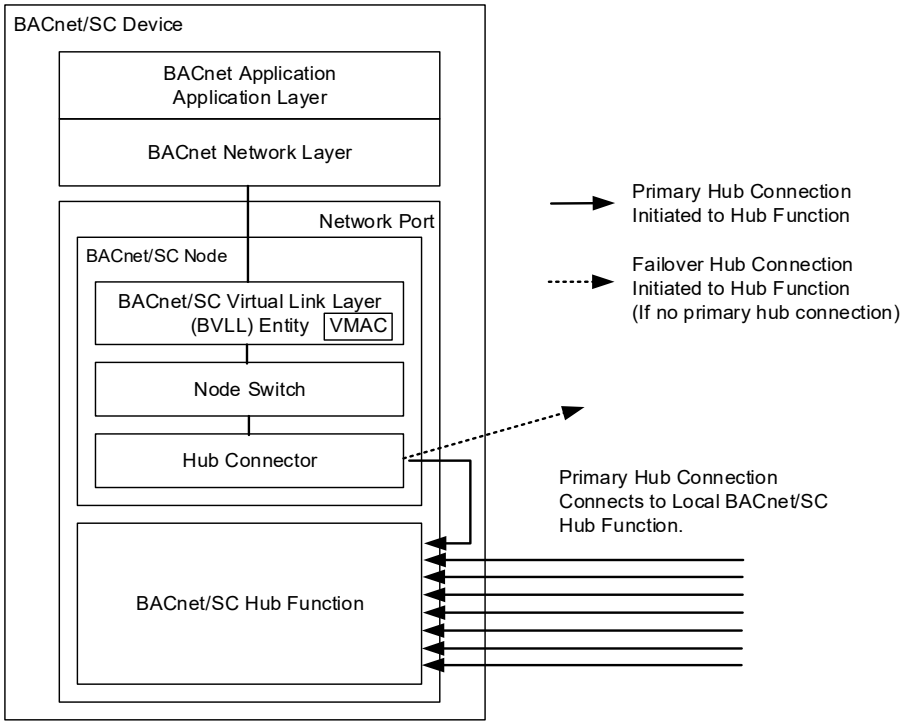


Figure YY-5. Example BACnet/SC Device with BACnet/SC Hub Function

YY.5.3.2 Directed BVLC Messages Forwarding

On receipt of a directed BVLC message from any current hub connection, the BVLC message shall be sent out through the hub connection with the connection peer node matching the destination VMAC. If no such connection exists, the hub function shall discard the directed BVLC message.

When a directed BVLC message is forwarded, the 'Originating Virtual Address' parameter shall be added, indicating the VMAC address of the connection peer node of the hub connection from which the message was received, and the 'Destination Virtual Address' parameter shall be removed.

YY.5.3.3 Broadcast BVLC Messages Forwarding

On receipt of a broadcast BVLC message from a hub connection, the hub function shall send a copy of the received message through each current hub connection, except the hub connection through which it was received.

When a broadcast BVLC message is forwarded, the 'Originating Virtual Address' parameter shall be added, indicating the VMAC address of the connection peer node of the hub connection from which the broadcast BVLC message was received, and the 'Destination Virtual Address' shall remain in the message so that the receiving node can determine that the message was a broadcast.

YY.5.3.4 Hub Connector for the BACnet/SC Hub Function

The hub connector of every BACnet/SC node shall support connecting to the BACnet/SC hub function as the primary hub function or the failover hub function. See Clause YY.5.2.

The URI for connecting to the primary hub function shall be configurable and is required to be a valid URI for an operational network.

The URI for connecting to the failover hub function shall be configurable. For installations where there is no failover hub function in use, this URI shall be empty, or otherwise marked as unconfigured, and the hub connector shall not attempt to connect to a failover hub.

YY.6 BACnet/SC Connections

BACnet/SC connections are based on secured WebSocket connections as defined by RFC 6455, and are used for bi-directional BVLC message exchange. For the application of the WebSocket protocol for BACnet/SC connections, see Clause YY.7.

The connection peer initiating the WebSocket connection is referred to as the initiating peer. The connection peer accepting the WebSocket connection is referred to as the accepting peer.

For direct connections, the node switch is the initiating peer or the accepting peer of BACnet/SC connections. For hub connections to the BACnet/SC hub function, the hub connector of a BACnet/SC node is the initiating peer, and the BACnet/SC hub function is the accepting peer.

While not needed for protocol operations, it may be useful for a node to know the identity of the BACnet device that is hosting the BACnet/SC hub function that it is connecting to. For example, this could provide additional checks that the URI is correct or provide access to other information about the hosting device. The use of this information is a local matter. To provide this information, in the Connect-Accept message from the hub function, the 'VMAC Address' parameter shall be the VMAC of the network port containing the hub function, and the 'Device UUID' parameter shall be the Device UUID of the BACnet device.

YY.6.1 BACnet/SC Reconnect Timeout

The minimum time for the initiating peer to wait between initiation attempts to reconnect a WebSocket connection is specified by the reconnect timeout. If the minimum reconnect timeout is configurable, the initiating peer shall support a range of 2..300 seconds for the minimum reconnect timeout. A fixed minimum reconnect timeout shall have a value in the range 10..30 seconds. Increasing reconnect timeouts should be applied between unsuccessful attempts to connect. The algorithm for increasing is a local matter; however, the reconnect timeout shall not be increased beyond 600 seconds.

YY.6.2 BACnet/SC Connection Establishment and Termination

Once a WebSocket connection is established as specified in Clause YY.7.5, the connection is required to be established as a BACnet/SC connection for general BVLC message exchange. To establish and close a BACnet/SC connection, both the initiating peer and the accepting peer execute a state machine and exchange BVLC messages to verify and accept the WebSocket connection to be a BACnet/SC connection, and to terminate such BACnet/SC connection.

BVLC messages other than Connect-Request, BVLC-Result, and Connect-Accept shall only be sent or processed on receipt when the connection is in CONNECTED state. In DISCONNECTING state, no BVLC message shall be initiated over the connection, but BVLC messages received shall be processed.

While waiting for a BVLC message from the connection peer, a connection wait timeout shall be applied. On expiration of the connection wait timeout, the peer shall close the WebSocket connection if any and enter the IDLE state.

The connection wait timeout shall be configurable. The BACnet/SC node shall support a minimum range of 5..300 seconds. The recommended default value is 10 seconds.

Closing an existing WebSocket connection, when one exists before entering the IDLE state, shall be performed as specified in Clause YY.7.5.5.

On unexpected failure, unexpected close by connection peer, or loss of the WebSocket connection, the local initiating or accepting peer shall enter the IDLE state.

YY.6.2.1 Duplicate Connections and VMAC Address Collisions

The BACnet/SC connections shall ensure that only one connection exists in a given context.

For the node switch, this shall ensure that only one direct connection to another node is used at a time. The local BVLL entity with its VMAC address and Device UUID is considered a connection peer of the node switch as well.

For the BACnet/SC hub function, this shall ensure that only one hub connection to a node is used at a time. The BACnet/SC node in the network port shall not be considered a connection peer unless it is currently connected to the local hub function by its hub connection.

The duplicate connection detection also provides detection of VMAC address collisions of initiating peers.

YY.6.2.2 BACnet/SC Connection Initiating Peer State Machine

The initiating peer state machine enters the IDLE state before a WebSocket connection is initiated. The time of initiation of a WebSocket connection is determined by the peer.

If an initiating peer receives a BVLC-Result NAK on the Connect-Request message with an 'Error Code' of NODE_DUPLICATE_VMAC, then the peer BACnet/SC node shall choose a new Random-48 VMAC before a reconnection is attempted.

For common error situations, also see Clause YY.3.1.5.

Figure YY-6 depicts the connection state machine for the BACnet/SC connection initiating peer.

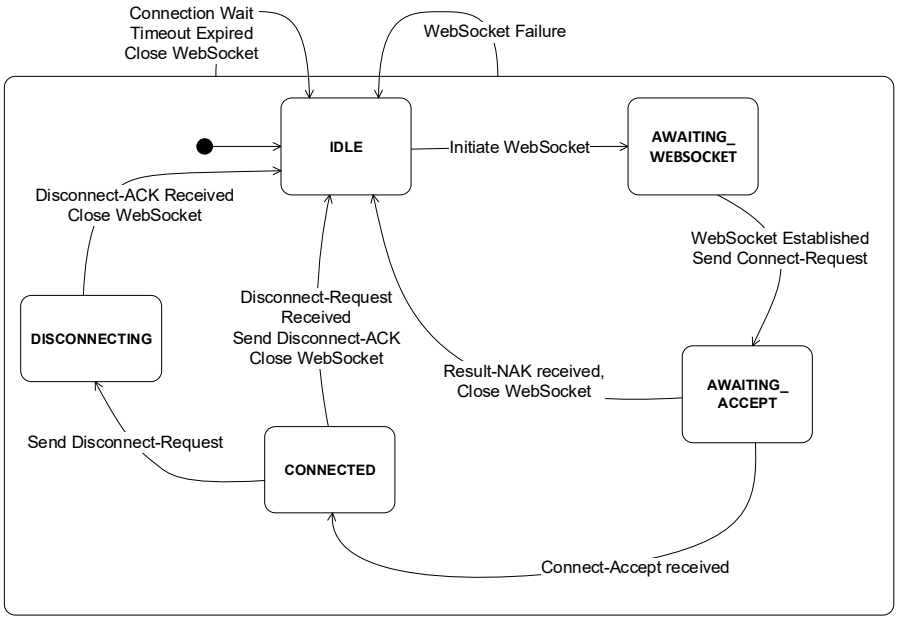


Figure YY-6. BACnet/SC Connection Initiating Peer Connection State Machine

In state **IDLE**

Initiating a WebSocket
On initiation of the WebSocket connection, the initiating peer shall enter the AWAITING_WEBSOCKET state.

In state **AWAITING_WEBSOCKET**

WebSocket established
On establishment of the WebSocket connection, the initiating peer shall send a Connect-Request and enter the AWAITING_ACCEPT state.

In state **AWAITING_ACCEPT**

BVLC-Result NAK, VMAC collision
On receipt of a BVLC-Result NAK message with an 'Error Code' of NODE_DUPLICATE_VMAC, then the initiating peer node shall choose a new Random-48 VMAC and enter the IDLE state.

BVLC-Result NAK received
On receipt of a BVLC-Result NAK message on the Connect-Request message initiated, the initiating peer shall close the WebSocket connection and enter the IDLE state.

Connect-Accept received
On receipt of a Connect-Accept message, enter the CONNECTED state.

Disconnect-Request received
On receipt of a Disconnect-Request message from the connection peer, respond with a Disconnect-ACK message, close the WebSocket connection, and enter the IDLE state.

Disconnect-ACK received
On receipt of a Disconnect-ACK message from the connection peer, close the WebSocket connection, and enter the IDLE state.

BVLC message received other than a Disconnect-Request, a Disconnect-ACK, or a response to the Connect-Request initiated
On receipt of a BVLC message other than a response to the initiated Connect-Request, discard the message and remain in the AWAITING_ACCEPT state.

In state **CONNECTED**

Local disconnection
On locally determined disconnection of the connection, the initiating peer shall send a Disconnect-Request message to the connection peer node and enter the DISCONNECTING state.

Disconnect-Request received
On receipt of a Disconnect-Request message from the accepting peer, respond with a Disconnect-ACK message to the accepting peer, close the WebSocket connection, and enter the IDLE state.

Disconnect-ACK received
On receipt of a Disconnect-ACK message from the accepting peer, close the WebSocket connection, and enter the IDLE state.

In state **DISCONNECTING**

Disconnect-ACK received
On receipt of a Disconnect-ACK message from the accepting peer, close the WebSocket connection, and enter the IDLE state.

YY.6.2.2.1 BACnet/SC Connection Accepting Peer State Machine

The accepting peer state machine enters the IDLE state before a WebSocket connection is accepted. In any state, in case an existing WebSocket connection is lost, the accepting peer shall enter the IDLE state. Figure YY-7 depicts the connection state machine for the BACnet/SC connection accepting peer.

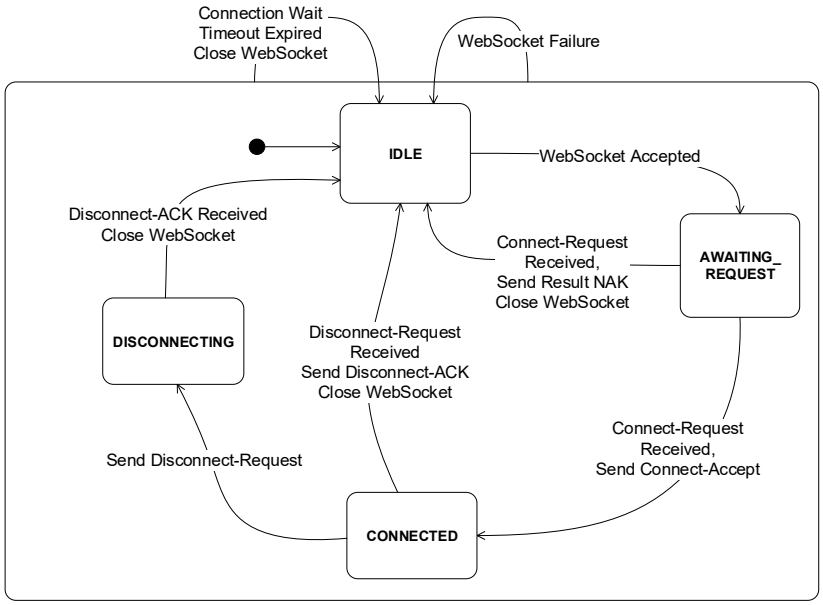


Figure YY-7. BACnet/SC Connection Accepting Peer Connection State Machine

In state **IDLE**

Accepting a WebSocket
On accepting a WebSocket connection, the accepting peer shall enter the AWAITING_REQUEST state.

In state **AWAITING_REQUEST**

BVLC message received other than Connect-Request
On receipt of a BVLC message other than a Connect-Request message, discard the message and remain in the AWAITING_REQUEST state.

Connect-Request received, new Device UUID, no VMAC collision
On receipt of a Connect-Request message from the initiating peer whose 'VMAC Address' is not equal to the accepting peer's VMAC address, and not equal to any of the initiating peers' VMAC addresses of existing connections, and the 'Device UUID' is not equal to any connection peer's Device UUID, then return a Connect-Accept message and enter the CONNECTED state.

Connect-Request received, known Device UUID
On receipt of a Connect-Request message from the initiating peer whose 'Device UUID' is equal to the initiating peer device UUID of an existing connection, then return a Connect-Accept message, disconnect and close the existing connection to the connection peer node with matching Device UUID, and enter the CONNECTED state.

Connect-Request received, new Device UUID, VMAC collision
On receipt of a Connect-Request message from the initiating peer whose 'VMAC Address' is equal to the accepting peer's VMAC address, or equal to any initiating peer VMAC addresses of an existing connection, and the ‘Device UUID’ is not equal to any initiating peer Device UUID of an existing connection, then return

a BVLC-Result NAK message with an 'Error Class' of COMMUNICATION and an 'Error Code' of NODE_DUPLICATE_VMAC', close the WebSocket connection and enter the IDLE state

In state **CONNECTED**

Local disconnection
On locally determined disconnection of the connection, the accepting peer shall send a Disconnect-Request message to the initiating peer and enter the DISCONNECTING state.

Disconnect-Request received
On receipt of a Disconnect-Request message from the initiating peer node, respond with a Disconnect-ACK message, close the WebSocket connection, and enter the IDLE state.

In state **DISCONNECTING**

Disconnect-ACK received
On receipt of a Disconnect-ACK message from the initiating peer, close the WebSocket connection, and enter the IDLE state.

YY.6.3 Connection Keep-Alive

Initiating peers shall keep established BACnet/SC connections alive through periodic initiation of Heartbeat-Request messages to the accepting peer.

A Heartbeat-Request message shall be initiated to the accepting peer if no BVLC message was received over the connection within the heartbeat timeout.

On receipt of Heartbeat-Request, the accepting peer shall respond with a Heartbeat-ACK message to the initiating peer.

If no BVLC message is received from the initiating peer within 2 times the heartbeat timeout, the connection shall be disconnected by the accepting peer.

If the heartbeat timeout is configurable, the connection peer shall support a minimum range of 3..300 seconds. A fixed heartbeat timeout shall have a value in the range 30..300 seconds.

The connections may be kept alive for as long as the WebSocket connection maximum lifetime allows. See Clause YY.7.5.4.

YY.7 Application of WebSockets in BACnet/SC

All secure WebSocket connections established for BACnet/SC connections shall apply the WebSocket protocol as specified in RFC 6455 and in the following subclauses. Only TLS-secured WebSocket connections are used. A minimum version of TLS V1.3 is required.

Secure WebSocket connections are used for BACnet/SC connections for bi-directional exchange of binary encoded BACnet/SC BVLC messages. A BACnet/SC network port may initiate and/or accept one or more WebSocket connections for BACnet/SC. Each WebSocket connection shall be used exclusively for one BACnet/SC connection.

In WebSocket connections for BACnet/SC connections, the connection peer that initiated the WebSocket connection is referred to as the initiating peer (i.e., "client" in RFC6455), and the connection peer that accepted the WebSocket connection is referred to as the accepting peer (i.e., "server" in RFC6455).

YY.7.1 The WebSocket Protocol

The use and support of the WebSocket protocol for BACnet/SC connections, and the purpose of the connection, shall be indicated by the WebSocket sub-protocol.

For BACnet/SC hub connections, the sub-protocol name "hub.bsc.bacnet.org" shall be used.

For BACnet/SC direct connections, the sub-protocol name "dc.bsc.bacnet.org" shall be used.

These sub-protocol identifiers are registered with IANA as defined in RFC 6455.

The use of the WebSocket Ping and Pong message exchange is optional and implementations shall not rely on its use by the connection peer.

YY.7.2 WebSocket URIs

The WebSocket URIs used for identifying peers accepting BACnet/SC connections shall be of "wss" URI scheme as defined in RFC 6455, Section 3.

YY.7.3 WebSocket Binary Data Payload Format

For BACnet/SC connections, fully encoded BVLC messages shall be sent as binary data payload frames over the WebSocket connection using data frame opcode 0x2. See RFC 6455 Section 5.6.

YY.7.4 Connection Security

The use of secure WebSocket connections as of RFC 6455 and TLS V1.3 as of RFC 8446 for BACnet/SC connections provides for confidentiality, integrity, and authenticity of BVLC messages transmitted across the connection.

The establishment of a secure WebSocket connection shall be performed as defined in RFC 6455. For establishing a secure WebSocket connection, mutual TLS authentication shall be performed. "Mutual authentication" in this context means that both the initiating peer and the accepting peer shall:

(a) Validate that the peer's operational certificate is well formed.
(b) Validate that the peer's certificate is active as of the current date and not expired.
(c) Validate that the peer's certificate is not revoked, if such information is available.
(d) Validate that the peer's certificate is directly signed by one of the locally configured CA certificates.

To ensure interoperability, no additional checks beyond the above shall be performed by default. Any additional checks, e.g., DN or SAN matches, shall only be performed if specifically enabled as directed by the installation.

In BACnet/SC, it is assumed that both the initiating and accepting peer of an established WebSocket connection are trusted, including all code they execute. The validation of such code and its origins is outside the scope of this standard.

BACnet/SC implementations shall support a minimum TLS version of 1.3 as specified in RFC 8446. Support of newer versions of TLS or cipher suites beyond those required by TLS 1.3 is a local matter. Additional supported TLS versions and cipher suites shall be listed in the PICS. See Annex A.

YY.7.4.1 Certificate Management

Secure WebSocket connections require the use of TLS. The creation of private keys, public certificates, and the management of the certificate signing authority, or authorities, are site-specific deployment options beyond the scope of this standard. However, to ensure interoperability, a BACnet/SC implementation shall support the storage and use of certificates as defined in the following clauses.

YY.7.4.1.1 Operational Credentials

Before deployment to an active network, the connection peers shall be configured with a CA certificate store containing one or more CA certificates of those CAs that are accepted to have signed the peer's certificate, and a unique operational certificate with matching private key. The operational certificate shall be issued and directly signed by a CA that is verifiable with one of the CA certificates configured in the CA certificate store as accepted CAs. This allows peer-to-peer mutual authentication so that the accepting peer and the initiating peer can each verify that the certificate presented to it was signed by one of the CAs in its CA certificate store.

YY.7.4.1.2 Signing CA

The choice of one or multiple CAs to sign the operational public certificates used in a site shall be dictated by site policy. The signing CAs shall support processing of certificate signing requests in Privacy Enhanced Mail (PEM) format (RFC 7468) conveying a certificate signing request, and return the signed certificates in PEM formatted PKCS7 structure.

YY.7.4.1.3 Configuring Operational Certificates

The configuration of operational credentials is performed by the configuration tool of the device. The configuration tool shall support the exchange of certificate signing requests and signed certificates in PEM format as of RFC 7468 with the signing CA of the installation. The protocol used by the tool to communicate to the signing CA for this exchange is outside the scope of this standard.

For devices that cannot generate their own public/private key pairs, the key pair needs to be generated by a configuration tool. In this case, the tool shall generate the key pair and create a certificate signing request based on certificate parameters defined by the installation. The tool shall submit the certificate signing request to the signing CA for the installation. The signed certificate returned from the CA, the private key, and the CA certificates required for the installation are configured into the device by the tool. The private key shall only be transferred in a secured environment, or over communication secured by TLS.

A device that supports an internal security function that allows it to generate its private keys and public certificates by itself, e.g., a hardware security module, is not allowed to expose the private keys, nor is it allowed to accept a private key from a configuration tool. To create a signed operational certificate, the configuration tool provides certificate parameters of the installation to the device and initiates a public certificate and private key generation by the device. The public certificate generated shall be used by the tool to build a certificate signing request to be sent to a signing CA of the installation. The signed certificate returned from the CA, and the CA certificates for the accepted CAs as required for the installation are configured into the device by the tool.

YY.7.4.2 Factory Defaults Condition

In the factory defaults condition, a connection peer shall not have operational credentials configured, and the device shall not contain any sensitive data.

YY.7.4.2.1 Reset to Factory Defaults

Devices shall provide a suitably secure out-of-band mechanism to place itself into "factory defaults" condition. It is recommended that this requires physical access to the device.

Performing a reset to "factory defaults" condition shall erase all operational certificates and respective private keys and all CA certificates from all BACnet/SC network ports. Any sensitive data the device contains shall also be erased. It is not allowed to simply block access to existing sensitive data while in the factory defaults condition because an attacker with physical access can use this condition to insert new operational credentials and then use that false trust relationship to access sensitive data that was not erased.

YY.7.5 WebSocket Connection Operation

WebSocket connections shall be initiated, accepted, and terminated by the peers as defined in RFC6455.

YY.7.5.1 Initiating WebSocket Connections

For BACnet/SC, only secured WebSocket connections over TLS V1.3 or higher shall be initiated.

If the WebSocket URI provided indicates a URI scheme other than "wss", no WebSocket connection to that URI shall be initiated. If applicable, an 'Error Code' of WEBSOCKET_SCHEME_NOT_SUPPORTED shall be indicated.

If the DNS resolution of the host name in the WebSocket URI fails, the following error codes can be used to indicate DNS error conditions, if known. If the specific DNS error is unknown, or no specific code is available, DNS_ERROR shall be indicated.

Situation	Error Code
DNS is unknown or not reachable	DNS_UNAVAILABLE
The host name cannot be resolved to its IP or IPv6 address.	DNS_NAME_RESOLUTION_FAILED
There is an error in the local DNS resolver that prevents it from resolving the host name.	DNS_RESOLVER_FAILURE
Any other DNS error situation	DNS_ERROR

If the host with the IP address resulting from DNS host name resolution is not reachable, and the respective IP error is available, then the following error codes can be used to indicate IP error conditions. If the specific IP error is unknown, or no specific code is available, IP_ERROR shall be indicated.

Situation	Error Code
IP address not reachable	IP_ADDRESS_NOT_REACHABLE
Any other IP error situation	IP_ERROR

If the TCP connection to the IP address and port cannot be established, and the respective TCP error is available, then the following error codes can be used to indicate TCP error conditions. If the specific TCP error is unknown, or no specific code is available, TCP_ERROR shall be indicated.

Situation	Error Code
The connection could not be established due to no response within timeout.	TCP_CONNECT_TIMEOUT
The connection is not accepted by the peer.	TCP_CONNECTION_REFUSED
Any other TCP error situation	TCP_ERROR

If the TLS session on the TCP connection cannot be established, and the respective fatal TLS error is available, then the following error codes can be used to indicate TLS error conditions. If the specific TLS error is unknown, or no specific code is available, TLS_ERROR shall be indicated.

Situation	Error Code
The security parameters of the client and server do not match.	TLS_SECURITY_PARAMETER_MISMATCH
The client certificate contains an error that prevents it from being authenticated.	TLS_CLIENT_CERTIFICATE_ERROR
The server certificate contains an error that prevents it from being authenticated.	TLS_SERVER_CERTIFICATE_ERROR
Authentication of the client failed.	TLS_CLIENT_AUTHENTICATION_FAILED
Authentication of the server failed.	TLS_SERVER_AUTHENTICATION_FAILED
Client certificate validity window does not include current time.	TLS_CLIENT_CERTIFICATE_EXPIRED
Server certificate validity window does not include current time.	TLS_SERVER_CERTIFICATE_EXPIRED
Client certificate revoked	TLS_CLIENT_CERTIFICATE_REVOKED
Server certificate is revoked	TLS_SERVER_CERTIFICATE_REVOKED
Any other TLS error situation	TLS_ERROR

If the HTTP exchange for upgrade to the WebSocket protocol fails, and the respective HTTP error is available, then the following error codes can be used to indicate HTTP error conditions. If the specific HTTP error is unknown, or no specific code is available, HTTP_ERROR shall be indicated.

Situation	Error Code
Server reports unexpected response code.	HTTP_UNEXPECTED_RESPONSE_CODE
Server does not accept upgrade to the WebSocket protocol.	HTTP_NO_UPGRADE
Redirect to another location of the peer WebSocket port received.	HTTP_RESOURCE_NOT_LOCAL
Proxy Authentication failed	HTTP_PROXY_AUTHENTICATION_FAILED
No response from server within timeout.	HTTP_RESPONSE_TIMEOUT
Syntax error in HTTP response received.	HTTP_RESPONSE_SYNTAX_ERROR
Errors in values of the HTTP response received.	HTTP_RESPONSE_VALUE_ERROR
Missing header fields in response.	HTTP_RESPONSE_MISSING_HEADER
Response contains any other error in HTTP header fields.	HTTP_WEBSOCKET_HEADER_ERROR
No upgrade request was received by the server.	HTTP_UPGRADE_REQUIRED
Upgrading to WebSocket protocol failed.	HTTP_UPGRADE_ERROR
No more HTTP connections are available currently.	HTTP_TEMPORARY_UNAVAILABLE
No inbound requests supported. The host is not an HTTP server.	HTTP_NOT_A_SERVER
Any other error situation	HTTP_ERROR

YY.7.5.2 Accepting WebSocket Connections

A network port that accepts WebSocket connections implements an HTTP server and supports HTTP upgrades to the WebSocket protocol. If serving as a BACnet/SC network port, it shall accept WebSocket connections for the appropriate BACnet/SC WebSocket sub-protocol, and exchange binary payloads as defined by this Annex.

For BACnet/SC, only WebSocket connections secured with TLS V1.3 or higher shall be accepted.

For accepted WebSocket connections that fail or terminate unintentionally, the error codes defined in Clause YY.7.5.1 can be used to indicate error situations to local higher layers or to a management entity.

YY.7.5.3 BACnet/SC BVLC Message Exchange

All BVLC messages are binary encoded as defined in Clause YY.2 and shall be transmitted as binary data frames. BVLC messages can be sent through a WebSocket connection in both directions.

If a non-binary data frame is received, then the WebSocket connection shall be closed with a status code of 1003 - WEBSOCKET_DATA_NOT_ACCEPTED.

If the length of a BVLC message received through a WebSocket connection exceeds the maximum BVLC length supported by the receiving node, the BVLC message shall be discarded and not be processed.

YY.7.5.4 Refreshing WebSocket Connections

WebSocket connections may be required to be refreshed such as when new key material must be generated periodically for TLS. TLS mechanisms shall be used to force session key refreshes. The method to determine the time of refreshing is a local matter. Security requirements, network load produced, and processing power requirements shall be considered in this determination.

YY.7.5.5 Closing WebSocket Connections

WebSocket connections may be closed by either end at any time. See RFC 6455.

The WebSocket close handshake shall be performed when intentionally closing a WebSocket connection. When a WebSocket connection is closed, the resulting close status shall be indicated for the associated WebSocket connection. The close status code received from the WebSocket connection shall map to error codes as follows. For the meaning of WebSocket response codes, see RFC 6455 Section 7.4.1.

WebSocket Close Status Code	Error Code
1000	WEBSOCKET_CLOSED_BY_PEER
1001	WEBSOCKET_ENDPOINT_LEAVES
1002	WEBSOCKET_PROTOCOL_ERROR
1003	WEBSOCKET_DATA_NOT_ACCEPTED
1006	WEBSOCKET_CLOSED_ABNORMALLY
1007	WEBSOCKET_DATA_INCONSISTENT
1008	WEBSOCKET_DATA_AGAINST_POLICY
1009	WEBSOCKET_FRAME_TOO_LONG
1010	WEBSOCKET_EXTENSION_MISSING
1011	WEBSOCKET_REQUEST_UNAVAILABLE
all other codes	WEBSOCKET_ERROR

135-2016*bj*-4. Add a Device_UUID Property to the Device Object

Rationale

For detection of duplicate connections and VMAC collisions, a BACnet device requires an immutable unique identifier that is independent of the device instance number.

[Change **Table 12-13**, p. 211]
[Note to reviewer: BACnet/SC is defined to be protocol revision agnostic but requires a device UUID. See Clause YY.1.5.3. However, the new Device_UUID property can only appear in the Device object of devices with the Protocol Revision of this addendum or higher. Before this protocol revision, the device UUID for BACnet/SC is not represented in the Device object and is configured by some other means.]

Table 12-13. Properties of the Device Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
. . .		
Device_UUID	*OCTET STRING (Size(16))*	*O*[x]
. . .		
Profile_Name	CharacterString	O

[1] These properties are required if, and shall be present only if, segmentation of any kind is supported.
. . .
[X] *This property shall be present if the device supports BACnet/SC network ports.*
...

[Add new **Clause 12.11.X**, p. 211

12.11.X Device_UUID

This read-only property, of type OCTET STRING (Size(16)), represents the Universally Unique ID of the device in binary representation.

This property shall have a valid 16-octet UUID before the device is first deployed in an installation. For the generation of UUIDs and the binary representation in the OCTET STRING, see RFC 4122.

[Insert into production **BACnetPropertyIdentifier** in **Clause 21**, preserving the alphabetical and numerical order, p. 845]

BACnetPropertyIdentifier ::= ENUMERATED { -- see below for numerical order
...
device-uuid *(?)*
...
-- numerical order reference
...
-- *see device-uuid* *(?)*
...
}
-- The special property identifiers ...

135-2016*bj*-5. Extend APDU Encoding for Large APDU Sizes

Rationale

This addendum section extends the APDU header parameter encoding for BACnet/SC allowing larger APDU sizes than 1476 octets.

[Clause 20 APDU Header Parameter Extension]

[Change Clause **20.1.2.5 max-apdu-length-accepted**, p. 759]

20.1.2.5 max-apdu-length-accepted

This parameter specifies the maximum size of a single APDU that the issuing device will accept. This parameter is included in the confirmed request so that the responding device may determine how to convey its response. *The exact maximum APDU length accepted by the issuing device may be larger than indicated in this parameter and is indicated in the Max_APDU_Length_Accepted property of the Device object of the issuing device. See Clause 12.11.18 and Clause 5.2.1.2.* The parameter shall be encoded as follows:

B'0000' Up to MinimumMessageSize (50 octets)
B'0001' Up to 128 octets
B'0010' Up to 206 octets (fits in a LonTalk frame)
B'0011' Up to 480 octets (fits in an ARCNET frame)
B'0100' Up to 1024 octets
B'0101' Up to 1476 octets (fits in *a 1497 octet NPDU in* one Ethernet frame)*, or larger than 1476 octets.*
B'0110' reserved by ASHRAE
B'0111' reserved by ASHRAE
B'1000' reserved by ASHRAE
B'1001' reserved by ASHRAE
B'1010' reserved by ASHRAE
B'1011' reserved by ASHRAE
B'1100' reserved by ASHRAE
B'1101' reserved by ASHRAE
B'1110' reserved by ASHRAE
B'1111' reserved by ASHRAE

135-2016*bj*-6. New Error Codes for BACnet/SC

Rationale

This addendum section introduces new error codes for BACnet/SC.

For the allowance of using these new error codes in any protocol revision, a new clause in Clause 22 "CONFORMANCE AND INTEROPERABILITY" is proposed that specifies what a device can use from protocol revisions higher than the device implements.

[Add new error codes to **Clause 18.7**, p. 740]

18.7 Error Class - COMMUNICATION

...

BVLC_FUNCTION_UNKNOWN *- The indicated BVLC function is unknown.*

BVLC_PROPRIETARY_FUNCTION_UNKNOWN *- The indicated BVLC proprietary function is unknown.*

HEADER_ENCODING_ERROR *- A header of the message has encoding errors.*

HEADER_NOT_UNDERSTOOD *- A message header that must be understood is not supported.*

MESSAGE_INCOMPLETE*- A message was presented that is incomplete.*

NOT_A_BACNET_SC_HUB *- The node received a message that would require it to be a hub, but the node is not a hub*

PAYLOAD_EXPECTED *- A payload was expected to be present in the message.*

UNEXPECTED_DATA *- Data was found that was not expected.*

HTTP_UNEXPECTED_RESPONSE_CODE *- Server reports unexpected response code.*

HTTP_NO_UPGRADE *- Server does not accept upgrade to the WebSocket protocol.*

HTTP_RESOURCE_NOT_LOCAL *- Redirect to another location of the peer WebSocket port received.*

HTTP_PROXY_AUTHENTICATION_FAILED *- Proxy Authentication failed*

HTTP_RESPONSE_TIMEOUT *- No response from server within timeout.*

HTTP_RESPONSE_SYNTAX_ERROR *- Syntax error in HTTP response received.*

HTTP_RESPONSE_VALUE_ERROR *- Errors in values of the HTTP response received.*

HTTP_RESPONSE_MISSING_HEADER *- Missing header fields in response.*

HTTP_WEBSOCKET_HEADER_ERROR *- Response contains any other error in HTTP header fields.*

HTTP_UPGRADE_REQUIRED *- No upgrade request was received by the server.*

HTTP_UPGRADE_ERROR *- Upgrading to the WebSocket protocol failed.*

HTTP_TEMPORARY_UNAVAILABLE *- No more HTTP connections are available currently.*

HTTP_NOT_A_SERVER *- No inbound requests are supported. The host is not an HTTP server.*

HTTP_ERROR *- An error occurred in HTTP.*

WEBSOCKET_SCHEME_NOT_SUPPORTED *- The WebSocket URI presented to the WebSocket port indicates a scheme whose respective protocol variant is not supported by the WebSocket port.*

WEBSOCKET_UNKNOWN_CONTROL_MESSAGE *- A WebSocket control message was received that was not understood.*

WEBSOCKET_CLOSE_ERROR *- An error occurred in closing the WebSocket connection.*

WEBSOCKET_CLOSED_BY_PEER *- The WebSocket connection was closed by the peer.*

WEBSOCKET_ENDPOINT_LEAVES *- An endpoint is "going away", such as a server going down or a client having left the WebSocket connection.*

WEBSOCKET_PROTOCOL_ERROR *- An endpoint has closed the WebSocket connection due to a protocol error.*

WEBSOCKET_DATA_NOT_ACCEPTED *- An endpoint has closed the WebSocket connection due to data of a type not accepted.*

WEBSOCKET_CLOSED_ABNORMALLY *- The WebSocket connection was closed abnormally.*

WEBSOCKET_DATA_INCONSISTENT *- An endpoint has closed the WebSocket connection due to data received that is inconsistent with the type of the message.*

WEBSOCKET_DATA_AGAINST_POLICY *- An endpoint has closed the WebSocket connection due to data received that is violating its policy.*

WEBSOCKET_FRAME_TOO_LONG *- An endpoint has closed the WebSocket connection due to data received that is too long to be processed.*

WEBSOCKET_EXTENSION_MISSING *- The initiating peer failed to establish the WebSocket connection due to extensions not confirmed by the answering peer.*

WEBSOCKET_REQUEST_UNAVAILABLE *- The answering peer has closed the WebSocket connection due to a condition that prevented it from executing the request received.*

WEBSOCKET_ERROR *- A WebSocket error occurred, and the WebSocket connection is closed.*

TLS_CLIENT_CERTIFICATE_ERROR *- The client certificate contains an error that prevents it from being authenticated.*

TLS_SERVER_CERTIFICATE_ERROR *- The server certificate contains an error that prevents it from being authenticated.*

TLS_CLIENT_AUTHENTICATION_FAILED *- Authentication of the client failed.*

TLS_SERVER_AUTHENTICATION_FAILED *- Authentication of the server failed.*

TLS_CLIENT_CERTIFICATE_EXPIRED *- Client certificate validity window does not include current time.*

TLS_SERVER_CERTIFICATE_EXPIRED *- Server certificate validity window does not include current time.*

TLS_CLIENT_CERTIFICATE_REVOKED *- Client certificate is revoked.*

TLS_SERVER_CERTIFICATE_REVOKED *- Server certificate is revoked.*

TLS_ERROR *- An error occurred in TLS.*

DNS_UNAVAILABLE *- The DNS name resolution service is unavailable*

DNS_NAME_RESOLUTION_FAILED *- The DNS host name resolution failed*

DNS_RESOLVER_FAILURE *- The DNS resolver failed.*

DNS_ERROR *- A DNS error occurred.*

TCP_CONNECT_TIMEOUT *- The TCP connection could not be established due to no response within timeout.*

TCP_CONNECTION_REFUSED *- The TCP connection is not accepted by the peer.*

TCP_CLOSED_BY_LOCAL *- The TCP connection was closed by the local endpoint.*

TCP_CLOSED_OTHER *- The TCP connection was closed due to some unspecified reason.*

TCP_ERROR *- An error occurred in TCP.*

IP_ADDRESS_NOT_REACHABLE *- The IP address of the peer is not reachable.*

IP_ERROR *- An error occurred on the IP protocol level.*

[Add new error codes to the **Error** production in **Clause 21**, p. 798]

Error ::= SEQUENCE {
-- NOTE: The valid combinations of error-class and error-code are defined in Clause 18.
error-class ENUMERATED {
...
},
-- Enumerated values 0-63 are reserved for definition by ASHRAE. Enumerated values
-- 64-65535 may be used by others subject to the procedures and constraints described
-- in Clause 23.

error-code ENUMERATED { -- see below for numerical order
...
[Insert here the new error codes as added in the numerical order list below into the alphabetical order list, maintaining the alphabetical order]
...
-- numerical order reference
...
-- see invalid-value-in-this-state (138),
-- *see bvlc-function-unknown* *(?),*
-- *see bvlc-proprietary-function-unknown* *(?),*
-- *see header-encoding-error* *(?),*
-- *see header-not-understood* *(?),*
-- *see message-incomplete* *(?),*
-- *see not-a-bacnet-sc-hub* *(?),*
-- *see payload-expected* *(?),*
-- *see unexpected-data* *(?),*
-- *see http-unexpected-response-code* *(?),*
-- *see http-no-upgrade* *(?),*
-- *see http-resource-not-local* *(?),*
-- *see http-proxy-authentication-failed* *(?),*
-- *see http-response-timeout* *(?),*
-- *see http-response-syntax-error* *(?),*
-- *see http-response-value-error* *(?),*
-- *see http-response-missing-header* *(?),*
-- *see http-websocket-header-error* *(?),*
-- *see http-upgrade-required* *(?),*
-- *see http-upgrade-error* *(?),*
-- *see http-temporary-unavailable* *(?),*
-- *see http-not-a-server* *(?),*
-- *see http-error* *(?),*
-- *see websocket-scheme-not-supported* *(?),*
-- *see websocket-unknown-control-message* *(?),*
-- *see websocket-close-error* *(?),*
-- *see websocket-closed-by-peer* *(?),*
-- *see websocket-endpoint-leaves* *(?),*
-- *see websocket-protocol-error* *(?),*
-- *see websocket-data-not-accepted* *(?),*
-- *see websocket-closed-abnormally* *(?),*
-- *see websocket-data-inconsistent* *(?),*
-- *see websocket-data-against-policy* *(?),*
-- *see websocket-frame-too-long* *(?),*
-- *see websocket-extension-missing* *(?),*
-- *see websocket-request-unavailable* *(?),*
-- *see websocket-error* *(?),*


```
                    -- see tls-client-certificate-error          (?),
                    -- see tls-server-certificate-error          (?),
                    -- see tls-client-authentication-failed      (?),
                    -- see tls-server-authentication-failed      (?),
                    -- see tls-client-certificate-expired        (?),
                    -- see tls-server-certificate-expired        (?),
                    -- see tls-client-certificate-revoked        (?),
                    -- see tls-server-certificate-revoked        (?),
                    -- see tls-error                             (?),
                    -- see dns-unavailable                       (?),
                    -- see dns-name-resolution-failed            (?),
                    -- see dns-resolver-failure                  (?),
                    -- see dns-error                             (?),
                    -- see tcp-connect-timeout                   (?),
                    -- see tcp-connection-refused                (?),
                    -- see tcp-closed-by-local                   (?),
                    -- see tcp-closed-other                      (?),
                    -- see tcp-error                             (?),
                    -- see ip-address-not-reachable              (?),
                    -- see ip-error                              (?),
                    ...
                    }
                    -- Enumerated values 0-255 are reserved for definition by ASHRAE. Enumerated values
                    -- 256-65535 may be used by others subject to the procedures and constraints described
                    -- in Clause 23.
}
```


135-2016*bj*-7. Interoperability Specification Extensions for BACnet/SC

Rationale

This addendum section defines the Annex A PICS modifications, Annex K BIBB extensions, and Annex L Device Profile additions for BACnet/SC.

[Change **Clause 23**, p. 875]

23 EXTENDING BACnet TO ACCOMMODATE VENDOR PROPRIETARY INFORMATION

The objective of BACnet is to provide the mechanisms by which building automation equipment may exchange information. To aid in interoperability, BACnet defines a standardized set of data structures, called objects, which contain information that is common to most building systems. BACnet may also be used to exchange non-standardized information between devices that understand this information. There are ~~four~~ *five* independent areas where BACnet may be extended to exchange non-standard information:

(a) A vendor may define proprietary extended values for enumerations used in BACnet.
(a) A vendor may invoke a proprietary service using the PrivateTransfer services.
(b) A vendor may add new proprietary properties to a standard object.
(c) A vendor may define new proprietary object types.
(d) *A vendor may include standard items from future revisions of this standard, limited to:*
    a. Error classes and error codes
    b. BACnetEngineeringUnits enumeration values
    c. BIBBs
    d. Device Profiles
    e. Data Links that do not require any object, property, or service changes

[Annex A PICS Changes for BACnet/SC]

[Change **Annex A**, p. 936]

BACnet Protocol Implementation Conformance Statement

...

BACnet Standardized Device Profiles Supported (Annex L):

☐ **BACnet Cross-Domain Advanced Operator Workstation (B-XAWS)**
…
☐ **BACnet Access Control Credential Reader (B-ACCR)**
☐ ***BACnet Secure Connect Hub (B-SCHUB)***

☐ **BACnet General (B-GENERAL)**

...

***BACnet* Data Link Layer Options:**

...
☐ Point-To-Point, modem, (Clause 10), baud rate(s): _____

☐ *BACnet Secure Connect Node (Annex YY)*
If direct connections are supported:
Maximum number of simultaneous direct connections initiated: _____
Maximum number of simultaneous direct connections accepted: _____

☐ *BACnet Secure Connect Hub Function (Annex YY)*
Maximum number of simultaneous hub connections accepted: _____

☐ *HTTPS Proxy Support*
List the types of HTTPS proxies supported _____

☐ *Transport Layer Security V1.3 supported (RFC 8446)*
Additional cipher suites supported beyond the minimum requirement of TLS V1.3, using the cipher suite names as of the TLS Cipher Suite Registry at IANA (See RFC 8446):
_____
_____

☐ *Transport Layer Security versions higher than V1.3 supported*
The TLS versions higher than V1.3 that are supported, including the supported cipher suites for the version beyond those required, using the cipher suite names as defined by the TLS version supported:
_____
_____

☐ *Generates private keys internally, and provides matching certificate signing requests.*

☐ *DNS host name resolution supported (RFC 1123)*
☐ *mDNS host name resolution supported (RFC 6762)*

...

[Annex K Network Management BIBB Additions for BACnet/SC]

[Add new BIBBs to **Clause K.5**, p. 1076]

K.5.X1 BIBB - Network Management-Secure Connect Hub-B (NM-SCH-B)

The B device supports at least one BACnet/SC network port and is capable of supporting the BACnet/SC hub function on at least one of these ports. The B device is capable of accepting hub connections and performing the BACnet/SC hub function.

Devices claiming conformance to this BIBB shall meet the minimum requirements for a BACnet device as described by this standard and specifically by Clause 22.

BACnet/SC BVLL Function	Initiate	Execute
Encapsulated-NPDU	x	x
Advertisement	x	x
Advertisement-Solicitation		x
Connect-Request		x
Connect-Accept	x	
Disconnect-Request	x	x
Disconnect-ACK	x	x

Heartbeat-Request		x
Heartbeat-ACK	x	

K.5.X2 BIBB - Network Management-Secure Connect Direct Connect -A (NM-SCDC-A)

The A device supports at least one BACnet/SC network port that implements the node switch and is able to initiate at least one direct connection.

Devices claiming conformance to this BIBB shall meet the minimum requirements for a BACnet device as described by this standard and specifically by Clause 22.

BACnet/SC BVLL Function	Initiate	Execute
Encapsulated-NPDU	x	x
Address-Resolution	x	
Address-Resolution-ACK		x
Advertisement	x	x
Advertisement-Solicitation	x	x
Connect-Request	x	
Connect-Accept		x
Disconnect-Request	x	x
Disconnect-ACK	x	x
Heartbeat-Request	x	x
Heartbeat-ACK	x	x

K.5.X3 BIBB - Network Management-Secure Connect Direct Connect - B (NM-SCDC-B)

The B device supports at least one BACnet/SC network port that implements the node switch and is able to accept at least one direct connection.

Devices claiming conformance to this BIBB shall support the following BACnet/SC BVLL functions.

BACnet/SC BVLL Function	Initiate	Execute
Encapsulated-NPDU	x	x
Address-Resolution		x
Address-Resolution-ACK	x	
Advertisement	x	x
Advertisement-Solicitation	x	x
Connect-Request		x
Connect-Accept	x	
Disconnect-Request	x	x
Disconnect-ACK	x	x
Heartbeat-Request		x
Heartbeat-ACK	x	

[Annex L Device Profile Changes for the BACnet/SC Hub Function]

[Change **Annex L**, p.1079]

...

BACnet device profiles are categorized into families:

- Operator Interfaces. This family is composed of B-XAWS, B-AWS, B-OWS, and B-OD.
- Life Safety Operator Interfaces. This family is composed of B-ALSWS, B-LSWS, and B-LSAP.
- Access Control Operator Interfaces. This family is composed of B-XAWS, B-AACWS, B-ACWS, and B-ACSD.
- Controllers. This family is composed of B-BC, B-AAC, B-ASC, B-SA, and B-SS.
- Life Safety Controllers. This family is composed of B-ALSC and B-LSC.
- Access Control Controllers. This family is composed of B-AACC and B-ACC.
- Miscellaneous. This family is composed of B-RTR, B-GW, B-BBMD, B-ACDC, ~~and~~ B-ACCR*, and B-SCHUB*.

[Change **Clause L.7**, p. 1092]

L.7 Miscellaneous Profiles

The following table indicates which BIBBs shall be supported by the device types of this family, for each interoperability area.

Data Sharing

B-RTR	**...**	***B-SCHUB***
DS-RP-B	...	*DS-RP-B*
DS-WP-B		

Alarm & Event Management

B-RTR	**...**	***B-SCHUB***

Scheduling

B-RTR	**...**	***B-SCHUB***

Trending

B-RTR	**...**	***B-SCHUB***

Device & Network Management

B-RTR	**...**	***B-SCHUB***
DM-DDB-B	...	*DM-DDB-B*
DM-DOB-B		*DM-DOB-B*
NM-RC-B		
		NM-SCH-B

[1] ...

[Insert new **Clauses L.7.X1**, maintaining the order as in the table in Clause L.7, p. 1094]

L.7.X1 BACnet Secure Connect Hub (B-SCHUB)

A B-SCHUB is a device that is able to perform the BACnet Secure Connect hub function as defined in Annex YY . The BACnet/SC hub function accepts secured Web Socket connections as defined for BACnet/SC hub connections.

Data Sharing
• Ability to provide the values of any of its BACnet objects.

Alarm and Event Management
• No requirement

Scheduling
• No requirement

Trending
• No requirement

Device and Network Management
• Ability to respond to queries about its status
• Ability to respond to requests for information about any of its objects
• Ability to respond to communication control messages
• Ability to perform the BACnet/SC hub function.

135-2016bj-8. Define Extended 6-Octet VMAC

Rationale

This addendum section defines a new extended 6-octet EUI-48 VMAC for BACnet/SC. This allows the use of an Ethernet MAC address as the VMAC, as well as a larger 44 bit random space to avoid collisions.

[Add new Clause H.7.X, p. 1020]

H.7.X EUI-48 and Random-48 VMAC Address

When a particular data link layer specifies that a EUI-48 VMAC is to be used, then the device shall use a 6-octet VMAC in the form of an IEEE EUI-48 identifier. The means of obtaining or generating the EUI-48 identifier is a local matter. For example, if a device has a physical Ethernet adapter, and there is only one BACnet device hosted by that adapter on a particular BACnet network, then the Ethernet hardware address would be an appropriate choice for the initial value of the VMAC.

The Random-48 VMAC is a 6-octet VMAC address in which the least significant 4 bits (Bit 3 to Bit 0) in the first octet shall be B'0010' (X'2'), and all other 44 bits are randomly selected to be 0 or 1. The generation of a Random-48 VMAC shall yield any Random-48 VMAC in the entire range with equal probability.

To ensure that the VMAC is not used by another device, the device shall attempt to resolve its own VMAC in the network. If the device detects that another device is already using this VMAC, the device shall generate a new Random-48 VMAC address and try again.

The values X'000000000000' and X'FFFFFFFFFFFF' are not valid as VMAC addresses for a device and can have other uses defined by the data link.

[Add a new entry to **History of Revisions**, p. 1364]

(This History of Revisions is not part of this standard. It is merely informative and does not contain requirements necessary for conformance to the standard.)

HISTORY OF REVISIONS

…	…	**…**
1	X	**Addendum bj to ANSI/ASHRAE 135-2016** Approved by the ASHRAE Standards Committee MONTH X, 20XX; by the ASHRAE Board of Directors MONTH X, 20XX; and by the American National Standards Institute MONTH X, 20XX. 1. Introduce BACnet Secure Connect Datalink Layer Option 2. Introduce BACnet/SC in the Application and Network Layer Specifications 3. Add new Annex YY for the BACnet Secure Connect Datalink Layer Option 4. Add a Device_UUID Property to the Device Object 5. Extend APDU Encoding for Large APDU Sizes 6. New Error Codes for BACnet/SC 7. Interoperability Specification Extensions for BACnet/SC 8. Define Extended 6-Octet VMAC